Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

Ingeniería de Ejecución en Informática

MÓDULO MANTENEDOR DE CONTRATOS

Autor:

Alexis Osvaldo Veas González

Profesor Guía:

Guillermo Cabrera Guerrero

Enero, 2009

Agradecimientos

Agradezco a mis Padres por la Luz, a Paula por el Sol.

RESUMEN

La Caja de Ahorros de Empleados Públicos, se encuentra en pleno proceso de actualización de toda su plataforma de hardware y software, con el fin de mejorar la productividad de sus empleados y elevar la calidad de la atención al público, centrándose en el manejo de la información.

Para estos efectos cuenta con un departamento de desarrollo, en donde se gestan y construyen los sistemas que permitirán alcanzar tales objetivos. A la luz de esta necesidad, es que nace la idea de confeccionar un Mantenedor de Contratos que entregue a los encargados de la administración de contratos una herramienta capaz de proporcionar una sustento y ayuda a la mantención que los funcionarios en este ámbito realizan, además de permitir reestructurar los procesos de negocio, estandarizándolos y haciéndolos mas comprensibles para todos los involucrados en este proceso.

El sistema "Mantenedor de Contratos", no incluye la confección de los contratos. Este se centra en el proceso de administración que los contratos requieren periódicamente, como es el control de Pago a proveedores y la renovación o el término de un contrato. Además de proporcionar: herramientas de auditoria que faciliten la obtención de información sin depender de los encargados de los contratos, enlazar esta información con el sistema de contabilidad central que agilizará la confección de las ordenes de egreso (documento contable que registro las salidas de dinero) por conceptos de pago de proveedores con contrato.

El proceso de desarrollo de este proyecto esta basado en el paradigma de desarrollo iterativo y la metodología orientada a objeto en el análisis y diseño de la solución. La etapa de análisis se basa en casos de uso mediante los cuales se estipulan los requerimientos, posteriormente se definen las clases, luego en el diseño se confeccionan los diagramas de secuencia mediante los cuales se establecen las operaciones necesarias para llevar a cabo los requerimientos. Posteriormente se diseñan las capas que estarán involucradas en los procesos intermedios y la capa de datos. Y por ultimo se estipulan las bases para la confección del diseño de interfaces.

El Sistema será proporcionado a los usuarios mediante el portal de la Intranet corporativa de la institución, será desarrollado con la herramienta Microsoft Visual Studio .NET 2003, el lenguaje de programación será C#, el motor de base de datos es SQL 2000.

Capítulo Uno

1 Presentación del Proyecto

1.1 INTRODUCCIÓN

Los sistemas informáticos que apoyan la administración y la gestión, prestan herramientas fundamentales al negocio, el control de los destinos de los flujos de dinero, el control de avance de obras, información histórica para futuras referencias y auditorias de responsabilidades. La unión de herramientas estadísticas con formularios de ingreso de datos, entregan a los usuarios información para el apoyo en la toma de decisiones que pueden cambiar la historia de las empresas.

Actualmente en el mercado es posible encontrar variados y complejos sistemas que cumplen papeles fundamentales, desde simples mantenedores para empresas pequeñas hasta grandes y robustos sistemas como son las soluciones de PeopleSoft, utilizados por CCU S.A. o SAP Portal utilizados en las plataformas de Codelco.

La importancia de contar con sistemas informáticos que sirvan de apoyo a la administración y gestión radica en la necesidad de las empresas de proteger sus intereses económicos, proteger las inversiones que se desarrollan y evaluar si es necesario contar con nuevas instancias de negocio que permitan destinar menores cantidades de tiempo y recursos al enfrentar riesgos y dificultades en el desarrollo de estos.

A continuación se detalla el proceso mediante el cual el proyecto Mantenedor de Contratos, se llevó a cabo abarcando todas las etapas de desarrollo de software.

1.2 ÁMBITO DEL PROBLEMA

1.2.1 Descripción de la Empresa

La Caja de Ahorros de Empleados Públicos, fue fundada en el año 1858, mientras era Presidente de la Republica Don Manuel Montt Torres (1809-1880), convirtiéndose en la primera en el continente Americano. Organizada como corporación de derecho privado de acuerdo con las normas del código Civil sobre personas jurídicas.

Esta institución fue concebida y estructurada como un instrumento para que los servidores públicos que, a la sazón, excepcionalmente obtenían una jubilación reunieran un pequeño patrimonio destinado a la protección exclusiva de su viuda o herederos.

1.2.1.1 Organigrama

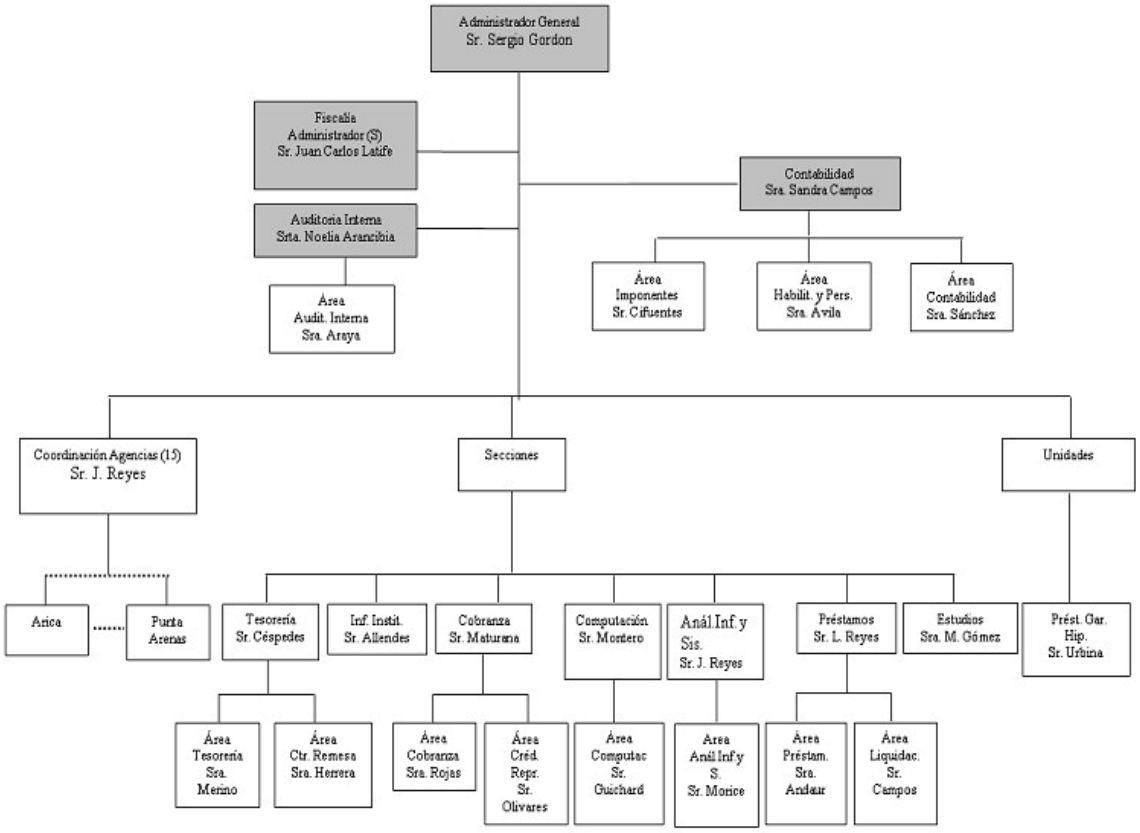


Ilustración 1.1 Organigrama

1.2.1.2 Área Análisis Informática y Sistemas.

Esta área esta compuesta por un Jefe de sección, sobre el cual pesa tanto la responsabilidad de la modernización de los actuales sistemas con que cuenta la Institución, como si también mantener actualizados y en funcionamiento las plataformas de hardware existentes. El equipo de trabajo divide sus funciones, entre la mantención de los equipos, tanto hardware como software, y la creación de herramientas capaces de facilitar el negocio.

Actualmente los desarrollos se realizan con Visual Basic 6.0 y SQL Server 2000, por lo que una implementación orientada a la Web vendrá a demostrar la utilidad y ventajas de esta plataforma. Cabe destacar que actualmente existe un desarrollo Web realizado en las instalaciones de la empresa, el cual espera la aprobación, para su utilización, una vez que este proyecto este en su segunda fase de construcción. Todo lo anterior debido a las dudas del área, en cuanto a orientar los futuros desarrollos utilizando la plataforma Web, en especial la herramienta de desarrollo escogida para la confección de este proyecto.

Plataforma de desarrollo: Visual Studio .NET de Microsoft.
Administrador de Base de Datos: SQL Server 2000 de Microsoft.
Control de versiones: Visual SourceSafe de Microsoft.

1.2.2 Exposición del Problema

Las empresas en la actualidad realizan negocios entre ellas, a distintos niveles y de distintas naturalezas, estableciendo mediante contratos, las obligaciones que se comprometen a efectuar.

Dentro de todos los departamentos de una empresa es posible encontrar estos contratos, por servicios que van desde Aseo, Mantención de Ascensores, Arriendos de Propiedades hasta Telecomunicaciones, Servicios de Procesamiento de Datos o Auditorias, como ejemplo. Al analizar, desde un punto de vista económico la importancia del control de la evolución de los contratos, podemos determinar que ésta radicaría en que de por medio existen las transacciones de dinero. Por lo tanto, una mala administración y control de éstos, se refleja

en pérdidas que van desde simples demoras en los plazos hasta grandes cantidades de dinero.

Un caso real sucedido en CAEP fue con una empresa de telecomunicaciones; se provoco un cambio en una de las jefaturas que tenía este contrato a cargo y como no se informó que este contrato vencía ese mes y que si explícitamente no se daba término, este se renovaba automáticamente, las facturas se siguieron pagando por alrededor de un año y medio provocando un gasto extra de unos 15 millones de pesos por un servicio que no fue utilizado.

Actualmente CAEP cuenta con anticuados software hechos a la medida para el control de la contabilidad y algunos mantenedores en desuso. Dentro de estos encontramos un mantenedor de contratos que solo permite el ingreso de los contratos, pero no su mantención, ni menos el control de pagos. Está escrito en Fortran y a pesar de haber sido desarrollado íntegramente en la institución no se posee el código fuente de este para su análisis.

Capítulo Dos

2 Objetivos

2.1 OBJETIVO GENERAL Y ESPECÍFICOS

Continuación se detallan el objetivo general y los objetivos específicos del proyecto, los cuales deberán ser alcanzados a través del desarrollo de este proyecto.

2.1.1 Objetivo General

Implementación de un sistema que permita mejorar el control y el acceso a la información de los contratos que posee la Institución, ya sean vigentes o históricos, información asociada a vigencia, plazos, periodicidad de cancelación, cálculos anómalos.

2.1.2 Objetivos Específicos

Proporcionar información relevante para los distintos responsables de proyectos.

Facilitar la autoría interna de los contratos de las áreas y secciones de CAEP.

Mejorar la comunicación entre los distintos involucrados en el proceso de los contratos.

Reducir el tiempo y el flujo de papel.

Capítulo Tres

3 Presentación del Problema

3.1 ESTADO DEL ARTE

A continuación se explican la situación actual del proceso que viven los contratos en la caja, las debilidades de este y se explica como actualmente los responsable control sus evoluciones.

3.1.1 Situación Actual

Una vez que el contrato ha sido firmado y por ende, autorizado por el Administrador, este es traspasado a la persona que lo tendrá a cargo, el responsable del contrato, el cual velará por el cumplimiento de esta y será la cara visible con el proveedor.

Dentro de las tareas que un responsable de contrato debe desarrollar se encuentra:

a) Visto bueno al Pago de factura: Periódicamente, según fecha de pago de cada factura, estas son recepcionadas por la sección Informaciones Institucionales, luego estas son enviadas por mano al responsable del contrato respectivo, quien las recibe y compara los valores que tiene con los estipulados en cada contrato, con esto da el visto bueno para que se efectúen los pagos, por lo tanto al momento de recibir la factura el debe contar con una copia del contrato para su evaluación y posterior pago.

Una vez que este visto bueno es entregado, la factura y la copia del contrato viajan a Contabilidad para registrar el pago y luego a Tesorería para que se genere le pago.

b) Chequeo de Vigencia y Plazos: Adicionalmente le responsable del contrato debe ir chequeando la vigencia de este, adelantando decisiones para continuar o detener los servicios asociados.

3.1.2 Debilidades del Sistema Actual

Las variables que se detallan a continuación son las debilidades del actual proceso de negocio, las cuales se esperan controlar de manera de alertar ante posibles errores en la mantención de los contratos.

Las diferentes modalidades de pago en los contratos en cuanto a su periodicidad, renovación o vigencia.

Cada responsable de contrato, guarda la información relacionada y es la única persona que conoce el procedimiento. En el caso que se provoque un cambio en el responsable del contrato, este debe informar al nuevo responsable, el actual proceder.

Cada uno de los responsables administra y gestiona el contrato a su modo, creándose múltiples formas de llevar este control, desde planilla de cálculo hasta bitácoras o cuadernos de registro.

Cuando se presenta un problema en el cual se vea afectado los compromisos u obligaciones que el contrato estipula, se deben de ocupar recursos extras para determinar responsabilidades ya que toda la información relacionada a los procesos sucedidos ha sido manipulada y mantenida sin el establecimiento de estándares.

Las auditorias son relativamente demorosas e involucran a personal y recursos para esto, dejando de realizar sus tareas habituales lo que se traduce en demoras de compromisos establecidos, horas extras en el personal a cargo de la auditoria, etc.

No existen reportes inmediatos que permitan ver exclusivamente los recursos utilizados por cada contrato, ni menos que permitan evaluar la conveniencia de estos y permitir tomar resoluciones ha cerca de su continuidad, los tiempos entre la solicitud del informe o reporte van desde un par de semanas a meses.

No existe comunicación entre los usuarios responsables de contratos, lo que permitiría una mejora en la administración, mantención o control de estos. El aislamiento que viven dificulta el control en la gestión de los responsables.

Contabilidad y Tesorería debe esperar que el responsable haga llegar los datos para registrar y generar el pago respectivamente.

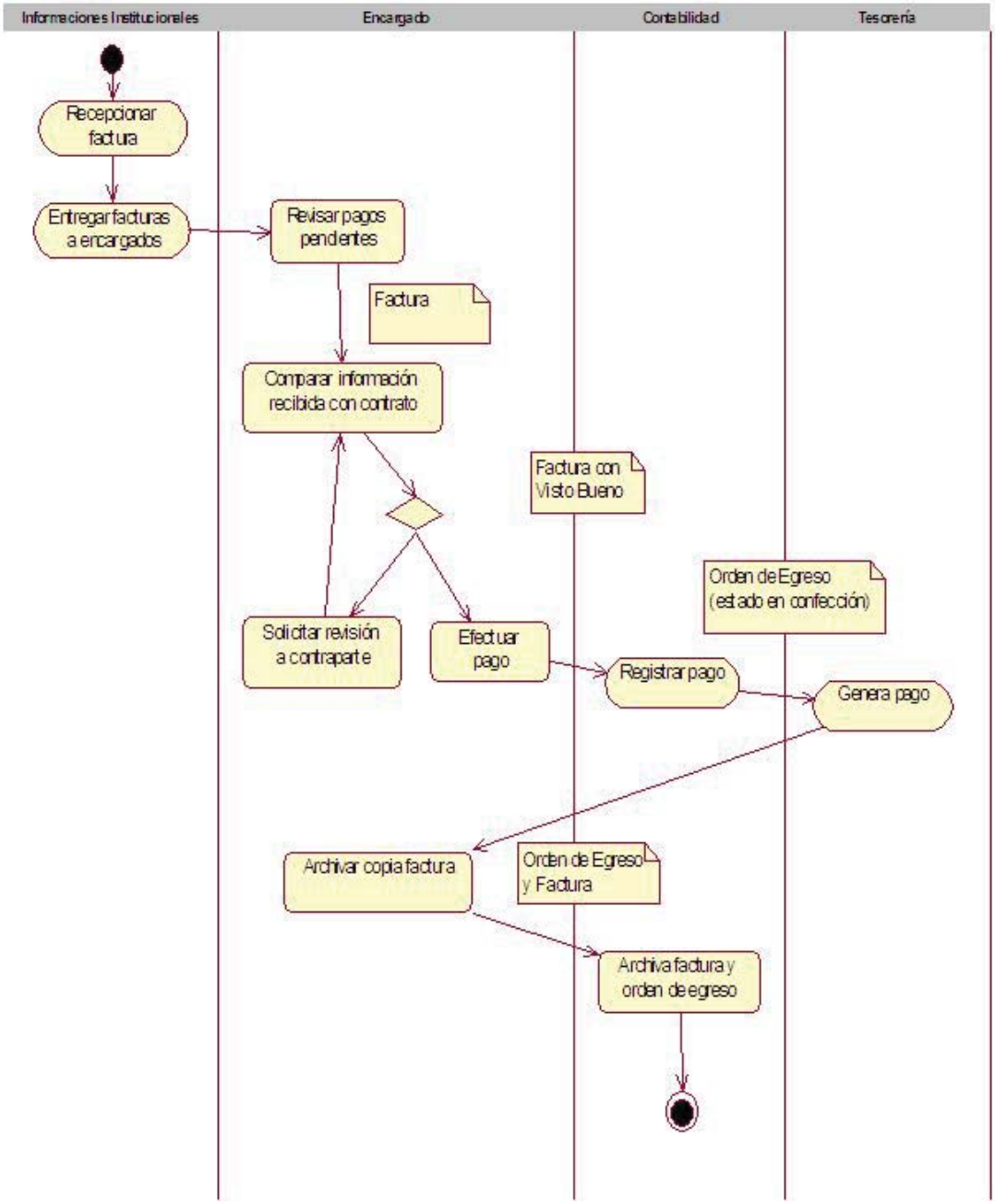


Ilustración 3.1 Diagrama de Actividad Situación Actual

Capítulo Cuatro

4 Plan de Trabajo

4.1 PLANIFICACIÓN

“La EDT es una descomposición jerárquica, orientada al producto entregable, del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos. La EDT organiza y define el alcance total del proyecto. La EDT subdivide el trabajo del proyecto en porciones de trabajo más pequeñas y fáciles de manejar, donde cada nivel descendente de la EDT representa una definición cada vez más detallada del trabajo del proyecto. El trabajo planificado comprendido dentro de los componentes de la EDT del nivel más bajo, denominados paquetes de trabajo, puede programarse, supervisarse, controlarse y estimarse sus costes.” [1]

En esta etapa del informe se reflejará el proceso de planificación y gestión del proyecto. Se inicia con la construcción de una WBS (Work Breakdown Structure) o EDT por sus siglas en español, Estructura de Desglose de Trabajo o Estructura de Descomposición de Trabajo; con esto determinamos las Tareas a realizar.

Dentro de las formas gráficas de representación de la EDT se eligió la de árbol, por presentar una visualidad más amplia y general de las actividades y tareas a realizar. Cabe destacar la herramienta de diagramación utilizada fue WBS Chart Pro, la cual se comunica perfectamente con MSProject, lo cual agiliza el proceso de pasar de una WBS a una carta Gantt.

Luego se realiza la EDO Estructura de desglose de Organizaciones, para establecer los responsables que deberán desarrollar los WorkPackage determinados en la WBS.

4.1.1 Especificación de Tareas

A continuación se realiza una especificación de tareas junto con sus responsables y recursos:

Numero	1.1.1
Nombre	WBS
Descripción	Creación de una EDT, Estructura de Desglose de Trabajo, el cual permita la especificación del alcance del proyecto y sirva de base para la programación
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe de proyecto
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	EDT
Predecesoras	

Numero	1.1.2
Nombre	OBS
Descripción	Crear la Estructura de Desglose de Organización la cual permita generar la matriz de responsabilidades
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe de proyecto

Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	OBS
Predecesoras	1.1.1 (D. Discrecional)

Numero	1.1.3
Nombre	Matriz de Responsabilidades
Descripción	Obtener el cruce entre la OBS y la WBS para generar el listado de responsables por tarea.
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe Proyecto
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	Tabla Matriz de Responsabilidades
Predecesoras	1.1.1 (D. Obligatoria); 1.1.2 (D. Obligatoria)

Numero	1.1.4
Nombre	Definición de Entregables
Descripción	Especificar los Hitos y sus entregables por tarea.

Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe Proyecto
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	Especificación de tareas completa
Predecesoras	1.1.3 (D. Obligatoria)

Numero	1.2.1
Nombre	Diagrama Gantt
Descripción	Se especifica el programa del proyecto mediante una herramienta gráfica como es el Diagrama Gantt.
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe Proyecto
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	Carta Gantt
Predecesoras	1.1.4 (D. Obligatoria)

Numero	1.2.2

Nombre	Diagrama Pert
Descripción	Construcción de 3 programas distintos.
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe de proyecto
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	Diagramas Pert
Predecesoras	1.2.1 (D. Obligatoria)

Numero	1.3
Nombre	Plan de Riesgos
Descripción	Identificación y Análisis de Riesgos, junto con la creación de plan de mitigación y contingencia
Esfuerzo Estimado	2 Día/ Hombre
Personas	Jefe de proyecto
Recursos	1 PC Desarrollo
Duración	2 Día
Entregables	Análisis de Riesgos
Predecesoras	1.1.4 (D. Obligatoria)

Numero	2.1.1
Nombre	Identificación
Descripción	Se realizará una reunión con el usuario para identificar los requerimientos funcionales del sistema.
Esfuerzo Estimado	2 Día/ Hombre
Personas	Analista/Programador, Usuario
Recursos	Sala de reuniones
Duración	2 Día
Entregables	Informe de identificación de requerimientos
Predecesoras	1.3 (D. Discrecional)

Numero	2.1.2
Nombre	Especificación
Descripción	Se debe construir una especificación formal de requerimientos mediante la utilización de casos de usos que deberán ser aprobados por el usuario.
Esfuerzo Estimado	4 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo
Duración	4 Día

Entregables	Especificación Formal Usando Casos de Uso
Predecesoras	2.1.1

Numero	2.2
Nombre	Arquitectura del Sistema
Descripción	Construcción y especificación de la disposición de los elementos estructurales que componen el sistema, en cuanto a su ubicación lógica.
Esfuerzo Estimado	1 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo
Duración	1 Día
Entregables	Informe de Arquitectura del Sistema
Predecesoras	2.1.2 (D. Obligatoria)

Numero	3.1
Nombre	Base de Datos
Descripción	Diseño de datos basados en los casos de usos.
Esfuerzo Estimado	2 Día/ Hombre
Personas	Analista/Programador

Recursos	1 PC Desarrollo
Duración	2 Día
Entregables	Modelo Datos Relacional
Predecesoras	2.2 (D. Obligatoria)

Numero	3.2
Nombre	Diseño de Módulos
Descripción	Diseño Modular de los componentes de software.
Esfuerzo Estimado	5 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo
Duración	5 Día
Entregables	Diagrama de Clases de diseño, diagrama de Componentes
Predecesoras	3.1 (D. Obligatoria)

Numero	3.3
Nombre	Diseño Procedimientos
Descripción	Se realizará un diagrama de Interacción, específicamente de secuencia para representar la colaboración de clases y sus respectivos métodos.

Esfuerzo Estimado	5 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo
Duración	5 Día
Entregables	Diagramas de Secuencia.
Predecesoras	3.2 (D. Obligatoria)

Numero	3.4
Nombre	Plan de Pruebas
Descripción	Se confeccionará un Plan de pruebas para cada uno de los casos de uso.
Esfuerzo Estimado	2 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo
Duración	2 Día
Entregables	Plan de Pruebas
Predecesoras	3.3 (D. Obligatoria)

Numero	4.1

Nombre	Base de Datos
Descripción	Se construye y codifica la base de datos del sistema
Esfuerzo Estimado	2 Día/ Hombre
Personas	Analista/Programador
Recursos	Servidor Base de Datos , 1 PC Desarrollo
Duración	2 Día
Entregables	Base de Datos lógica.
Predecesoras	3.1 (D. Obligatoria)

Numero	4.2
Nombre	Implementación C.U.
Descripción	Codificación de los casos de uso.
Esfuerzo Estimado	15 Día/ Hombre
Personas	Analista/Programador
Recursos	Servidor Base de Datos, Servidor Web, 1 PC Desarrollo
Duración	15 Día
Entregables	Listado de interfaces finales, código fuente y componente ejecutable.
Predecesoras	4.1 (D. Obligatoria)

Numero	5.1
Nombre	Pruebas Unitarias
Descripción	Se llevará acabo el plan de pruebas, previo ajuste.
Esfuerzo Estimado	5 Día/ Hombre
Personas	Analista/Programador
Recursos	Servidor Base de Datos, Servidor Web, 1 PC Desarrollo
Duración	5 Día
Entregables	Plan de pruebas ajustado, Informe de resultados de las pruebas
Predecesoras	4.2 (D. Obligatoria)

Numero	5.2
Nombre	Pruebas Integración
Descripción	Se realizaran las pruebas de integración del modulo con los demás módulos del sistema SIIN.
Esfuerzo Estimado	3 Día/ Hombre
Personas	Analista/Programador
Recursos	Servidor Base de Datos, Servidor Web, 1 PC Desarrollo, módulos Perfiles, Contabilidad Proyecto SIIN
Duración	3 Día
Entregables	Informe de pruebas del sistema, Informe de fallas detectadas

Predecesoras	5.1 (D. Obligatoria)

Numero	6.1
Nombre	Correcciones Módulos
Descripción	Se realizarán las correcciones correspondientes, las cuales fueron detectadas en la fase de pruebas del sistema
Esfuerzo Estimado	5 Día/ Hombre
Personas	Analista/Programador
Recursos	Servidor Base de Datos, Servidor Web, 1 PC Desarrollo
Duración	5 Día
Entregables	Informe de Módulos corregidos
Predecesoras	5.2 (D. Obligatoria)

Numero	6.2
Nombre	Actualización de Documentos
Descripción	Se actualizarán los documentos de Diseño según las nuevas modificaciones.
Esfuerzo Estimado	2 Día/ Hombre
Personas	Analista/Programador
Recursos	1 PC Desarrollo

Duración	2 Día
Entregables	Documentos Actualizados
Predecesoras	6.1 (D. Obligatoria)

Numero	7.1
Nombre	Presentación Gerencia
Descripción	Se realizará una presentación del sistema a la gerencia la que incluye un informe de funcionalidades.
Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe de proyecto, Jefes de Departamento y Administrador.
Recursos	Sala de Reuniones, Proyector, 1 PC
Duración	1 Día
Entregables	Informe de funcionalidades, Presentación PowerPoint
Predecesoras	6.2 (D. Obligatoria)

Numero	7.2
Nombre	Presentación Área Informática.
Descripción	Se realizará una presentación del sistema al área Informática de la institución, de esta manera los encargados de respaldo y soporte se informarán al respecto.

Esfuerzo Estimado	1 Día/ Hombre
Personas	Jefe de proyecto
Recursos	Sala de Reuniones, Proyector, 1 PC
Duración	1 Día
Entregables	Informe de Respaldo y soporte.
Predecesoras	7.1 (D. Obligatoria)

4.1.2 Definición de Hitos

Hitos Externos	**Fecha**
Entrega especificación de Casos de Uso.	24/03/2006
Entrega de Carta Gantt	10/03/2006
Presentación gerencia	22/05/2006
Presentación Área informática	23/05/2006

Hitos Internos	**Fecha**
Revisión Especificación Formal de Requerimientos	22/03/2006
Informe de Diseño	07/04/2006
Instalación Componentes ejecutables	02/05/2006
Pruebas del sistema	11/05/2006

Presentaciones	22/05/2006

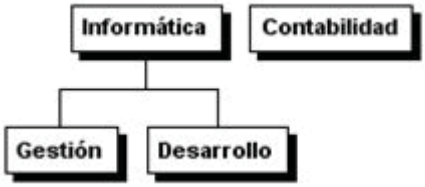


Ilustración 4.1 Organización Breakdown Structure (OBS)

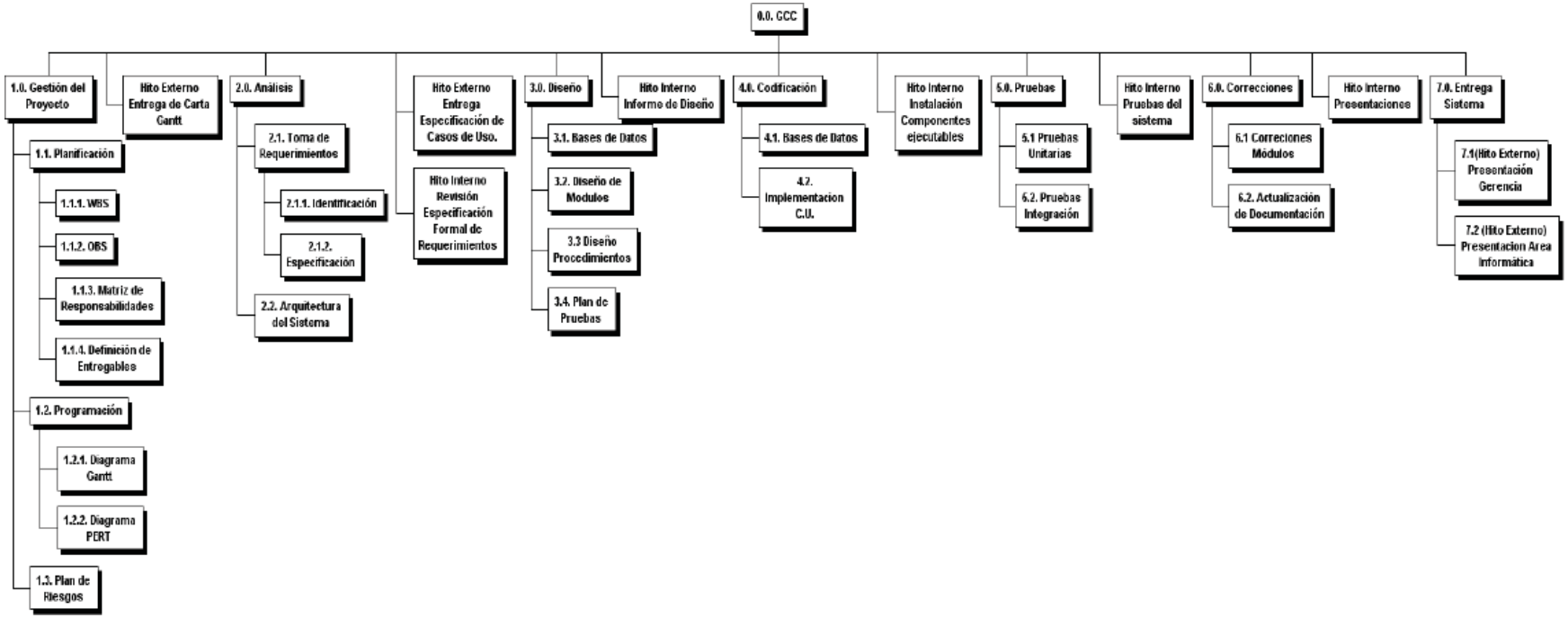

0.0. GCC
1.0. Gestión del Proyecto
1.1. Planificación
1.1.1. WBS
1.1.2. OBS
1.1.3. Matriz de Responsabilidades
1.1.4. Definición de Entregables
1.2. Programación
1.2.1. Diagrama Gantt
1.2.2. Diagrama PERT
1.3. Plan de Riesgos
Hito Externo Entrega de Carta Gantt
2.0. Análisis
2.1. Toma de Requerimientos
2.1.1. Identificación
2.1.2. Especificación
2.2. Arquitectura del Sistema
Hito Externo Entrega Especificación de Casos de Uso.
Hito Interno Revisión Especificación Formal de Requerimientos
3.0. Diseño
3.1. Bases de Datos
3.2. Diseño de Modulos
3.3 Diseño Procedimientos
3.4. Plan de Pruebas
Hito Interno Informe de Diseño
4.0. Codificación
4.1. Bases de Datos
4.2. Implementacion C.U.
Hito Interno Instalación Componentes ejecutables
5.0. Pruebas
5.1 Pruebas Unitarias
5.2. Pruebas Integración
Hito Interno Pruebas del sistema
6.0. Correcciones
6.1 Correciones Módulos
6.2. Actualización de Documentación
Hito Interno Presentaciones
7.0. Entrega Sistema
7.1(Hito Externo) Presentación Gerencia
7.2 (Hito Externo) Presentacion Area Informática

Ilustración 4.2 Work Breakdown Structure (WBS)

Id		Nombre de tarea	Duración	Comienzo	Fin
1		**0.0. GCC**	**62 días**	**mié 01/03/06**	**jue 25/05/06**
2		**1.0. Gestión del Proyecto**	**6 días**	**mié 01/03/06**	**mié 08/03/06**
3		**1.1. Planificación**	**2,67 días**	**mié 01/03/06**	**vie 03/03/06**
4		1.1.1. WBS	0,5 días	mié 01/03/06	mié 01/03/06
5		1.1.2. OBS	0,92 días	mié 01/03/06	mié 01/03/06
6		1.1.3. Matriz de Responsabilidades	0,83 días	mié 01/03/06	jue 02/03/06
7		1.1.4. Definición de Entregables	0,92 días	jue 02/03/06	vie 03/03/06
8		**1.2. Programación**	**3,33 días**	**vie 03/03/06**	**mié 08/03/06**
9		1.2.1. Diagrama Gantt	1,17 días	vie 03/03/06	lun 06/03/06
10		1.2.2. Diagrama PERT	1 día	mié 08/03/06	mié 08/03/06
11		1.3. Plan de Riesgos	1,67 días	vie 03/03/06	mar 07/03/06
12		Hito Externo Entrega de Carta Gantt	0,33 días	vie 10/03/06	vie 10/03/06
13		**2.0. Analisis**	**8 días**	**jue 09/03/06**	**lun 20/03/06**
14		**2.1. Toma de Requerimientos**	**7 días**	**jue 09/03/06**	**dom 19/03/06**
15		2.1.1. Identificación	1,83 días	jue 09/03/06	vie 10/03/06
16		2.1.2. Especificación	4,67 días	lun 13/03/06	dom 19/03/06
17		2.2. Arquitectura del Sistema	1 día	lun 20/03/06	lun 20/03/06
18		Hito Interno Revisión Especificación Formal de Requerimientos	1 día	lun 20/03/06	lun 20/03/06
19		Hito Externo Entrega Especificación de Casos de Uso.	0,33 días	vie 24/03/06	vie 24/03/06
20		**3.0. Diseño**	**13,83 días**	**mar 21/03/06**	**vie 07/04/06**
21		3.1. Bases de Datos	1,75 días	mar 21/03/06	mié 22/03/06
22		3.2. Diseño de Modulos	4,67 días	mié 22/03/06	mié 29/03/06
23		3.3 Diseño Procedimientos	4,67 días	mié 29/03/06	mié 05/04/06
24		3.4. Plan de Pruebas	1,83 días	jue 06/04/06	vie 07/04/06
25		Hito Interno Informe de Diseño	1 día	vie 07/04/06	vie 07/04/06
26		**4.0. Codificación**	**16,17 días**	**vie 07/04/06**	**lun 01/05/06**
27		4.1. Bases de Datos	1,83 días	vie 07/04/06	mar 11/04/06
28		4.2. Implementacion C.U.	14,5 días	mar 11/04/06	lun 01/05/06
29		Hito Interno Instalación Componentes ejecutables	1 día	mar 02/05/06	mar 02/05/06
30		**5.0. Pruebas**	**7,83 días**	**mar 02/05/06**	**jue 11/05/06**
31		5.1 Pruebas Unitarias	4,83 días	mar 02/05/06	lun 08/05/06
32		5.2. Pruebas Integración	3 días	lun 08/05/06	jue 11/05/06
33		Hito Interno Pruebas del sistema	1 día	jue 11/05/06	jue 11/05/06
34		**6.0. Correcciones**	**7 días**	**jue 11/05/06**	**lun 22/05/06**
35		6.1 Correciones Módulos	4,5 días	jue 11/05/06	jue 18/05/06
36		6.2. Actualización de Documentación	2,5 días	jue 18/05/06	lun 22/05/06
37		Hito Interno Presentaciones	1 día	lun 22/05/06	lun 22/05/06
38		**7.0. Entrega Sistema**	**2 días**	**mié 24/05/06**	**jue 25/05/06**
39		7.1(Hito Externo) Presentación Gerencia	1 día	mié 24/05/06	mié 24/05/06
40		7.2 (Hito Externo) Presentacion Area informática	1 día	jue 25/05/06	jue 25/05/06

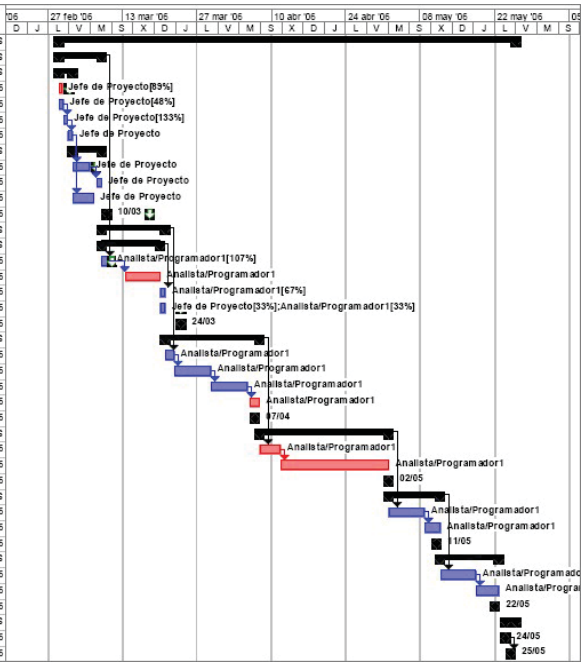


Ilustración 4.3 Carta Gantt

Capítulo Cinco

5 Desarrollo del Proyecto

5.1 ESTUDIO DE FACTIBILIDAD

Esta etapa del proceso de software determina si el sistema solicitado es factible. Existen tres aspectos relacionados:

5.1.1 Factibilidad Técnica

Dividiremos esta área en dos criterios: desarrollo y explotación; y cada uno se dividirá en dos, software y hardware. Se especifican los requerimientos mínimos para cada caso.

Desarrollo

- Software:

  Licencia para el desarrollo de software con la herramienta Microsoft Visual Studio .NET, versión 2003

  Procesador: Equipo con procesador a 600 MHz o CPU compatible con Intel Pentium de velocidad superior

  Memoria: 160 megabytes (MB) de RAM

  Disco duro: 3.3 gigabytes (GB) de espacio disponible en disco duro

  SQL Server 2000

  Procesador: Equipo con procesador a 450 MHz o CPU compatible con Intel Pentium de velocidad superior

  Memoria: 512 megabytes (MB) de RAM

  Disco duro: 6 gigabytes (GB) de espacio disponible en disco duro

  Sistemas operativos:

  WinXp:

  Procesador: Pentium a 233 MHz o mayor velocidad (se recomienda 300 MHz)

  Memoria: 64 megabytes (MB) de RAM (se recomienda 128 MB)

  Disco duro: 1.5 gigabytes (GB) de espacio disponible en disco duro

  Win2003 Server:

  Procesador: Pentium a 133 megahercios (MHz) o mayor velocidad (se recomienda 550 MHz)

  Memoria: 128 megabytes (MB) de RAM (se recomienda 256 MB)

  Disco duro: 4 gigabytes (GB) de espacio disponible en el disco duro

- **Hardware:**
  - Estaciones:
    - **Procesador:** Equipo con procesador a 600 MHz o CPU compatible con Intel Pentium de velocidad superior
    - **Memoria:** 256 megabytes (MB) de RAM
    - **Disco duro:** 20 gigabytes (GB) de espacio disponible en disco duro
  - Servidores:
    - **Procesador:** Equipo con al menos un procesador a 1 GHz o superior
    - **Memoria:** 512 megabytes (MB) de RAM
    - **Disco duro:** 30 gigabytes (GB) de espacio disponible en disco duro

Explotación

- **Software:**
  - SQL Server 2000
  - Sistemas operativos: WinXp, Win2003 Server.
- **Hardware:**
  - Estaciones:
    - **Procesador:** Equipo con procesador a 1 GHz o CPU compatible con Intel Pentium de velocidad superior
    - **Memoria:** 512 megabytes (MB) de RAM
    - **Disco duro:** 30 gigabytes (GB) de espacio disponible en disco duro
  - Servidores:
    - **Procesador:** Equipo con al menos un procesador a 2 GHz o superior
    - **Memoria:** 1 gigabytes (GB) de RAM
    - **Disco duro:** 40 gigabytes (GB) de espacio disponible en disco duro

La institución en la cual se desarrollará la implantación cuenta actualmente con la totalidad de estos requerimientos, debido al cambio tecnológico que se efectuó con miras a la puesta en marcha del sistema Integrado SIIN.

Específicamente:

- **Software:**
  Licencia para el desarrollo de software con la herramienta Microsoft Visual Studio .NET, versión 2003
  Motor Base de Datos SQL Server 2000
  Sistema Operativo WinXp.
  Sistema Operativo Win2003 Server
  Sistema Operativo Win2000 Server.
  Servidor Web IIS 6.0
- **Hardware:**
  Estaciones de Trabajo Usuarios:
  Procesador: Pentium IV, 3.0 GHz.
  Memoria: 1 gigabytes (GB) de RAM
  Disco duro: 80 gigabytes (GB)
  Estaciones Desarrollador:
  Procesador: DualCore 2.3 GHz
  Memoria: 2 gigabytes (GB) de RAM
  Disco duro: 250 gigabytes (GB)
  Servidores ProLiant de HP:
  ML 370.
  Procesador: 2 Procesadores Dual-Core Intel® Xeon® Processor L5240 (3.00 GHz, 40W, 1333 MHz FSB)
  Memoria: 4 gigabytes (GB) de RAM
  Disco duro: 2 discos de 72 gigabytes (GB) Raid 1
  ML 330.
  Procesador: 2 Procesadores Dual-Core Intel® Xeon® Processor L5240 (3.00 GHz, 40W, 1333 MHz FSB)
  Memoria: 2 gigabytes (GB) de RAM
  Disco duro: 2 discos de 36 gigabytes (GB) Raid 1

La Arquitectura física de sistema es Web. El lugar físico de la implantación es un edificio de 7 pisos, en los cuales se encuentran las instalaciones de CAEP, una red en estrella por piso conectada por Router Cisco a la sala de servidores ubicada en el 7º piso.

En cuanto a los conocimientos de las herramientas, se cuenta con una experiencia en el desarrollo de software en plataformas Web y desarrollo con .Net.

Con este equipamiento el desarrollo e implementación del modulo es completamente factible técnicamente.

5.1.2 Factibilidad Económica

La Institución cuenta actualmente con las licencias de desarrollo y el hardware de plataforma donde sustentarlo.

Los beneficios obtenidos en largo plazo por este modulo serán medidos en las disminuciones progresivas de los tiempos que los responsables tomaran para la administración.

Según entrevistas, cuantitativamente podemos decir que actualmente la administración de contratos le toma en promedio al responsable entre media a 2 horas por contrato, dependiendo de los problemas que este tenga ya sea en el pago por errores en la facturación, disponibilidad de la información del contrato relacionado, disponibilidad de la entrega de factura y el viaje que hace esta desde su área a contabilidad.

Si tomamos como referencia el promedios de esta variable obtendremos que en un mes un responsable ocupe cerca 1.25 horas en la administración de un contrato.

Promedio horas destinadas: 1.25 horas.

Si el valor hora promedio de un responsable es cercano a los $4.500 tenemos:

4.500 * 1.25 = $ 5.625 por mes

22.500 * 12 = $67500 en un año por contrato

Calculando un promedio de 50 contratos

$67500* 50 = $3.375.000 en un año.

Esto sin considerar los errores por facturación, que por falta de tiempo no son detectados, o los contratos que continúan su vigencia a pesar de no ser ya útiles y que su detección se realiza con posterioridad.

El tiempo estimado de desarrollo es de 2 meses y medio, con un recurso destinando un 100% de un día laboral, cuyo costo es de:

$4.166 * 8 = $33.328 diario

$33.328 * 30 = $999.840 mensual

$999.840 * 2.5 = $2.499.600 Valor HH del proyecto.

El tiempo destinado por el responsable una vez implementado el módulo será entre 15 a 45 minutos, considerando un promedio de 30 minutos los nuevos valores serán:

$4.500*0.5=$2.250

$2.250*12=$27.000

$27.000*50=$135.0000

Calculando:

Obtendremos que la reducción de costos estimada sea de $ 3.240.000.-

Con una inversión Inicial de $2.499.600.-

Calculando por concepto de costo de Operación de $50.000.- anual.

Se obtiene un flujo anual de $3.190.000

Una Tasa de descuento del 10%

VAN de $8.792.210.-

TIR de 125%

Por lo tanto se puede concluir que el proyecto es factible económicamente.

5.1.3 Factibilidad Operacional

El sistema fue aprobado por el administrador, señor Gordon, el cual confeccionará una orden de servicio mediante la cual, ordenará que todos los contratos deban ser mantenidos por el presente sistema. Y dada las características jerárquicas presentadas en la caja, la factibilidad de la utilización está el software queda cubierta. No obstante, las interfaces se realizaron pensadas en el facilitar la comprensión por parte del usuario, de manera de disminuir el factor de rechazo. Por lo tanto el proyecto es factible operacionalmente.

5.1.4 Factibilidad Legal

La institución cuenta con las licencias de desarrollo y explotación necesarias para el proyecto. Cabe recordar que el objetivo es desarrollar un modulo del sistema integrado SIIN. Por lo tanto es factible legalmente.

Dado los escenarios anteriores, el proyecto es factible.

5.2 PARADIGMA Y METODOLOGÍA

5.2.1 Análisis de Paradigmas

Se detallan los paradigmas a Analizados:

5.2.1.1 Modelo de Cascada

[2]Está compuesto de 5 etapas:

Análisis y definición de requerimientos.
Diseño de sistemas y de software.
Implementación y prueba de unidades.
Integración y prueba del sistema.
Operación y mantenimiento.

La desventaja más conocida y divulgada de este modelo es su inflexibilidad, ya que no es posible comenzar con una etapa antes de haber terminado con la anterior. Cuando los requerimientos están bien definidos y se establecen compromisos de no alteración de estos, el modelo fluye de la forma deseada, pero lamentablemente la realidad se encarga de traicionar estos compromisos y los requerimientos aparecen una vez diseñado o aun peor una vez construido el software, los costos asociados a integrar estos nuevos requerimientos son muy altos y casi siempre componen nuevos proyectos.

La ventaja de este modelo es su estructura y su simple administración. Ya que al ser estructurado no representa problemas para sus flujos.

5.2.1.2 Desarrollo Evolutivo

[2]Este se basa en la idea de confeccionar una implementación inicial, la cual es evaluada por el o los usuarios y se refinará con sucesivas versiones las que permitirán ir definiendo los requerimientos hasta la obtención del software deseado.

Se pueden identificar tres instancias por las cuales cada una de las versiones irá pasando hasta la versión final:

Especificación.
Desarrollo.
Validación.

La desventaja principal de este tipo de desarrollo es el poco control del tiempo que tomará realizar el software, se recomienda que este sea aplicado a proyectos de pequeña envergadura y cuyo periodo de vida sea relativamente corto. Según Ian Sommerville, se pueden identificar 3 problemas fundamentales: El proceso no es visible, a menudo los sistemas tienen una estructura deficiente y Se requieren herramientas y técnicas especiales.

5.2.1.3 Desarrollo Incremental

[2]Este tipo de desarrollo combina las ventajas de ambos enfoques descritos con anterioridad, el cascada y el evolutivo. Según Ian Sommerville, en su libro Ingeniería de Software, las ventajas de este tipo de desarrollo las podríamos sintetizar en:

Los clientes en un principio definen los requerimientos mas importantes o críticos y se establecen prioridades de desarrollo, con esto se definen los incrementos a realizar y los productos entregables en cada uno de estos incrementos.

Para el desarrollo de cada uno de estos incrementos, es posible definir distintos tipos de desarrollo, eligiendo en cada uno el más adecuado.

Las ventajas principales son:

El cliente no esperará mucho tiempo antes de recibir el primer producto, el cual satisface su requerimiento más crítico.

Los incrementos iniciales sirven de prototipos para evaluar la experiencia de usuario que estos tengan con el software, esta experiencia es traspasable a los nuevos incrementos.

El riesgo de falla es relativamente bajo debido al reiterado feedback que se posee del cliente. En otras palabras, las fallas del software se reducen en cada una de las iteraciones, con lo que el producto final tendrá menos errores y la percepción del usuario final, con respecto al producto es mejor.

5.2.1.4 Proceso Unificado

El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología orientada a objetos en el desarrollo de software de misión crítica en una variedad de industrias por la compañía Rational.[3]

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

Interacción con el usuario continua desde un inicio

Mitigación de riesgos antes de que ocurran

Liberaciones frecuentes

Aseguramiento de la calidad
Involucramiento del equipo en todas las decisiones del proyecto
Anticiparse al cambio de requerimientos
El Proceso Unificado se enfoca en la arquitectura como el centro del desarrollo, para asegurar que el desarrollo basado en componentes sea clave para un alto nivel de reuso.

Las características primordiales del Proceso Unificado son:

Iterativo e incremental
Centrado en la arquitectura
Guiado por casos de uso
Confrontación de riesgos

Según Jacobson el Proceso Unificado es un proceso porque “define quién está haciendo qué, cuándo lo hace y cómo alcanzar cierto objetivo, en este caso el desarrollo de software”.

[4] Booch Plantea en 1998, los conceptos clave del Proceso Unificado son:

Fase e iteraciones	¿Cuándo se hace?
Flujos de trabajo de procesos (actividades y pasos)	¿Qué se está haciendo?
Artefactos (modelos, reportes, documentos)	¿Qué se produjo?
Trabajador: un arquitecto	¿Quién lo hace?

5.2.1.4.1 El Ciclo de Vida del Software en el Proceso Unificado

Las fases del ciclo de vida del software son: concepción, elaboración, construcción y transición. La concepción es definir el alcance del proyecto y definir el caso de uso. La elaboración es proyectar un plan, definir las características y cimentar la arquitectura. La construcción es crear el producto y la transición es transferir el producto a sus usuarios.

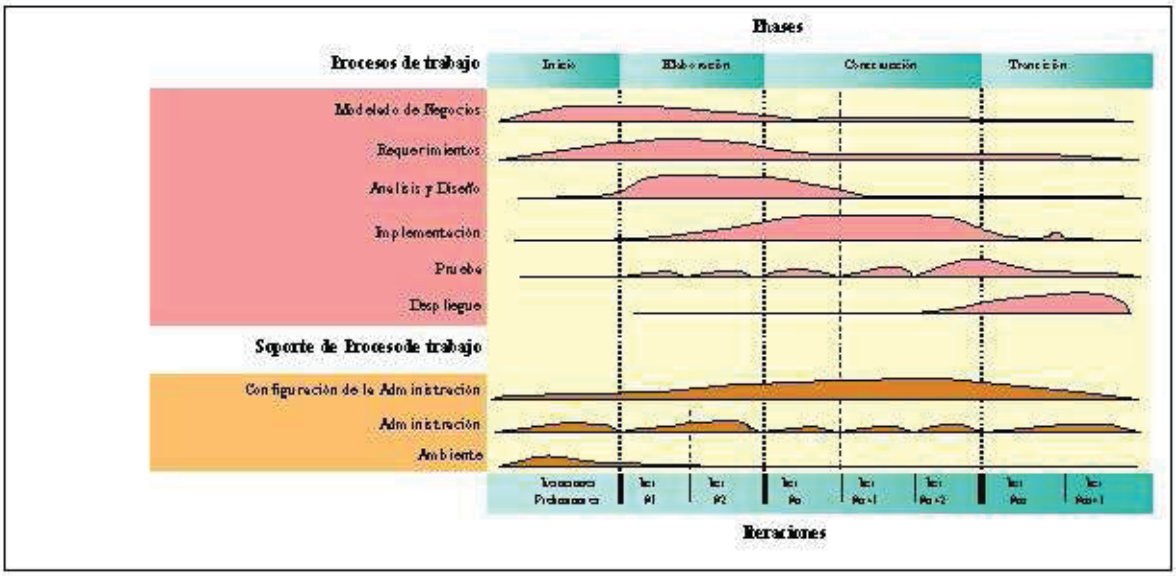


Ilustración 5.1 Estructura del Proceso Unificado

5.2.2 Selección del Paradigma a Utilizar

Se ha seleccionado el Paradigma Desarrollo Incremental y la orientación a objeto como metodología de análisis.

Esta elección es debido a las particularidades de esta implantación, que requiere que los usuarios no expertos, se familiaricen en forma paulatina con el nuevo estándar para la administración de contratos y puedan ser capaces de visualizar nuevos requerimientos, los cuales serán integrados al modelo en una próxima iteración del proyecto la cual incluirá la construcción de los requerimientos de integración al sistema SIIN.

[5]Según la Dra. Lioubov Dombrovskaia, en sus apuntes de ingeniería de software, estipula que las directrices de un proceso eficiente de desarrollo esta basado en los siguientes principios:

Desarrollo iterativo e incremental.

Desarrollo orientado a los casos de uso. Desarrollo centrado en la arquitectura.

Si bien son ciertos estos principios, están en estrecha relación con el Proceso unificado de desarrollo de software, el de paradigma seleccionado facilita la combinación de uno o más paradigmas de desarrollo en cada uno de sus iteraciones. Además, estos principios sumados a la metodología orientada a objetos ya han sido probados en el desarrollo del sistema de Contabilidad Central de CAEP, con excelentes resultados.

Por lo tanto se espera que realizar el proyecto bajo estos parámetros favorezca además de la integración, la unificación de los sistemas en cuanto a su documentación y elaboración.

5.3 ANÁLISIS DE RIESGOS

[6]“Risk can be defined as the combination of the probability of an event and its consequences”

El propósito del plan de gestión de riesgos es asegurar que los riesgos del proyecto en desarrollo son identificados adecuadamente, analizados, documentados, mitigados, monitoreados y controlados. Además describe el enfoque que se utilizará para identificar, analizar, priorizar, monitorear y mitigar los riesgos.

El plan de gestión de riesgos debe ser actualizado cuando los riesgos o estrategias de mitigación cambien. El responsable de dicha actualización es el Jefe de Proyecto que debe estar permanentemente monitoreando el estado del proyecto, y lo posibles riesgos que se presenten, para ejecutar el plan de mitigación o contingencia según sea el caso.

5.3.1 Organización y Responsabilidades

El Equipo de personas involucradas en la Gestión del Riesgo es:

- ***Jefe de Proyecto***

Mantener el control de cada una de las tareas del proyecto, efectuar revisiones de las entregas, ejecutar las tareas trasversales al proyecto, además del análisis de requerimientos y el diseño de la solución, a nivel de negocio como técnico.

- ***Analista Desarrollador***

Participar en el análisis de requerimientos, diseño de solución e implementación de la solución en la herramienta seleccionada.

5.3.2 Grupos Responsables del Riesgo:

Los grupos involucrados en la Gestión del Riesgo son:

- ***Equipo Técnico***

Corresponde a los responsables de efectuar análisis, diseño, implementación y puesta en marcha el sistema, además de controlar tareas, productos y el cumplimiento de los plazos estipulados. Sus integrantes son: Jefe de Proyecto; Analista Desarrollador.

- ***Contraparte Usuaria***

Corresponde a los usuarios validadores y auditores de los requerimientos entregados, además de ser fuente de información, o el medio de comunicación entre el equipo técnico y los usuarios expertos. Sus integrantes son

- ***Usuarios Expertos***

Corresponden a usuarios operativos de las funcionalidades involucradas por lo que entregan el detalle de los procesos y actividades que se realizan en la institución. Sus integrantes son personas que son jerárquicamente responsables de las actividades en las diferentes secciones, áreas y agencias.

5.3.3 Riesgos a Controlar

Los riesgos identificados y que están sujetos a control son los siguientes:

Asociados al Equipo Técnico

Conocimiento integral del negocio.
Tiempos mayores a lo planificado.
Conocimiento de la Herramienta de desarrollo. (asociado al diseño y posteriores)
Documentación asociada al Sistema incompleta.

Asociados a la Contraparte Usuaria y a los Usuarios Expertos

Entrega de información inexacta.
Incumplimiento de los plazos de entrega de información o de observaciones.

5.3.4 Control y Detalle de los Riesgos

[8]La posibilidad de que existan riesgos está siempre presente, pudiendo interferir en gran medida en la planificación de todo tipo de actividades. Es así como debe hacerse todo lo posible porque el impacto de un cierto hecho negativo sea mínimo, pudiendo, mediante un plan de contingencia, manejarlo adecuadamente.

Este plan de contingencia debe estar orientado a detectar posibles riesgos que puedan hacer fracasar el proyecto si no son enfrentados correctamente.

La evaluación y el control de riesgos se muestran en la hoja de Información de Riesgo.

La descripción de la hoja de Información de Riesgo es la siguiente:

- **Prioridad:** Importancia del riesgo con respecto a los otros riesgos identificados y analizados, asignándole a cada uno un número de entre 1 y 10 según su importancia.

- **Probabilidad :** Muestra la posibilidad de ocurrencia de cada riesgo (en porcentaje), y que al igual que la prioridad se han definido tres niveles:
  - **Alta:** Entre 100% y 81%.

- **Media:** Entre 80% y 41%.
- **Baja:** Entre 40% y 0%.

- **Impacto:** Indica la magnitud de pérdida en cuanto a costo y tiempo. Definiendo para esto tres niveles:
  - **Alto:** El impacto del riesgo es altamente perjudicial, ocasionando graves problemas.
  - **Medio:** El impacto es de mediano cuidado**.**
  - **Bajo:** El impacto no es grave.

- **Timeframe:** Se refiere a cuando podría ocurrir el riesgo, pudiendo ser a Corto, Mediano o Largo plazo.

<table>
<tr><td>ID NEG_01</td><td colspan="3">Proyecto Mantenedor de Contratos

Etapa: Análisis, Diseño, Desarrollo, Pruebas.</td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td>Prioridad: 1</td><td colspan="3" rowspan="3">Declaración

Conocimiento integral del negocio, a partir de los requerimientos entregados por los usuario se obtiene el diseño lógico de la solución a construir, por lo que, si toda la información no está clara o inexacta puede llevar a que el resultado final no sea el adecuado, implementando erróneamente los requerimientos del sistema.</td></tr>
<tr><td>Probabilidad: Media</td></tr>
<tr><td>Impacto: Alto</td></tr>
<tr><td>Timeframe
Mediano Plazo</td><td>Origen

Jefe de Proyecto</td><td>Categoría Riesgo

Del Negocio</td><td>Asignado a

Equipo Técnico</td></tr>
</table>

Contexto Si bien es cierto el proceso de administración de contratos en CAEP no es muy complejo pero si con varias particularidades que lo hacen diferente a cualquier otra organización, los conocimientos de las actividades están en la mente de los funcionarios que han trabajado por mucho tiempo en la institución, más que como una norma, por lo que capturar todos los requerimientos y reglas del negocio se hace complejo.
Estrategias de mitigación o Se efectuarán reuniones con todas las personas que están involucradas, o que participan en alguna parte del proceso. o Esta información se validará con la Contraparte Usuaria de manera de conocer en detalle la realidad actual para poder entregar una nueva solución en el caso que sea necesario. o La información proveniente de usuarios expertos se validará en forma cruzada con los documentos archivados y otras entidades que tengan incidencia en dicha información.
Plan de Contingencia o En caso de no comprender en detalle alguna actividad se solicitarán nuevas reuniones a los usuarios para aclarar los temas, solicitando información en detalle, ya sea acompañada con documentos u otro tipo de anexo que sirva para el entendimiento completo. Se debe contemplar que esto requiere un mayor tiempo de análisis y se debe dimensionar el tiempo involucrado.
Trigger o Información incompleta o inexacta.

<table>
<tr><td>ID PRO_1</td><td colspan="3">Proyecto Mantenedor de Contratos

Etapa: Todas las etapas.</td></tr>
<tr><td>Prioridad: 2</td><td colspan="3" rowspan="3">Declaración

Tiempos mayores a lo planificado, según el plan de trabajo establecido, lo que trae consigo que el tiempo total de desarrollo se incremente.</td></tr>
<tr><td>Probabilidad: Alta</td></tr>
<tr><td>Impacto: Medio</td></tr>
<tr><td>Timeframe

Mediano</td><td>Origen

Jefe Proyecto</td><td>Categoría Riesgo

Del Proyecto</td><td>Asignado a

Jefe Proyecto</td></tr>
<tr><td colspan="4">Contexto

La implementación del Sistema está estimada en un rango de tiempo que se puede ver afectado si el análisis o el diseño se extienden más de lo estimado.</td></tr>
<tr><td colspan="4">Estrategias de mitigación

o Se debe hacer una planificación detallada de cada una de las actividades a realizar, con un margen de tiempo para posibles atrasos que se produzcan.</td></tr>
<tr><td colspan="4">Plan de Contingencia

o En caso de que los atrasos sean mayores a los estimados se debe efectuar una re planificación informándole a todas las partes involucradas, se deben entregar tareas a cumplir a las partes involucradas de manera de disminuir el impacto asociado.

o Es posible comenzar a construir las partes que están claras, para no retrasar la implementación.</td></tr>
</table>

Trigger o Realizar tareas no programadas o programadas sub o sobre dimensionadas.

<table>
<tr><td>ID TEC_1</td><td colspan="3">Proyecto Mantenedor de Contratos

Etapa: Diseño, Desarrollo, Pruebas y Marcha Blanca.</td></tr>
<tr><td>Prioridad: 1</td><td colspan="3" rowspan="3">Declaración

Conocimiento de la Herramienta de desarrollo (asociado al diseño y posteriores), el trabajo con nuevas tecnologías puede traer problemas por no conocer las herramientas que se utilizan, para lo cual puede ser posible necesitar de tiempos de aprendizaje superiores.</td></tr>
<tr><td>Probabilidad: Media</td></tr>
<tr><td>Impacto: Alto</td></tr>
<tr><td>Timeframe

Mediano</td><td>Origen

Jefe de Proyecto</td><td>Categoría Riesgo

Técnico</td><td>Asignado a:

Jefe de Proyecto</td></tr>
<tr><td colspan="4">Contexto

La herramienta de desarrollo que se defina no es parte del conocimiento de los desarrolladores, por lo que su etapa de inducción puede ser mayor a una herramienta con la que ya se haya trabajado.</td></tr>
</table>

Estrategias de mitigación o Se entregarán los espacios para el estudio de la herramienta seleccionada. o Creación de vínculos con personas con experiencia en la herramienta. o Asistencia a charlas técnicas o Investigación que cada desarrollador debe hacer como parte de su formación.
Plan de Contingencia o Realizar una capacitación formal.
Trigger o Imposibilidad de construir algún artefacto en las herramientas utilizadas.

ID PRO_2	**Proyecto Mantenedor de Contratos** Etapa: Desarrollo, Puesta en Marcha, Mantención.		
Prioridad: 2 **Probabilidad**: Media **Impacto**: Alto	**Declaración** Documentación asociada al Sistema incompleta, lo que trae consigo una dependencia a las personas involucradas en el proyecto que no debe existir, pudiendo cualquier persona con conocimientos en la herramienta efectuar mantenciones a la aplicación.		
Timeframe: Largo Plazo	**Origen:** Jefe Proyecto	**Categoría Riesgo** Del Proyecto	**Asignado a** Jefe Proyecto

Contexto Cada una de las actividades y etapas debe estar documentada para que a futuro el sistema sea de fácil mantención y no dependa de las personas que trabajaron en él.
Estrategias de mitigación o Identificar cada uno de los documentos que deben ser construidos durante las diferentes etapas y actividades desarrolladas a los largo del proyecto, velando por que todas la información necesaria se encuentre descrita en detalle.
Plan de Contingencia o Documentar cada información que no se encuentre disponible una vez que se haya encontrado algún punto pendiente. o Documentación disponible y conocida en Servidor central.
Trigger o No disponibilidad de la Información requerida.

ID PRO_3	**Proyecto Mantenedor de Contratos** Etapa: Análisis.
Prioridad: 2	**Declaración**
Probabilidad: Alta	Incumplimiento de los plazos de entrega de información o de observaciones, lo que conlleva a que los plazos definidos se atrasen.
Impacto: Medio	

Timeframe:	**Origen:**	**Categoría Riesgo**	**Asignado a**
Corto Plazo	Jefe de proyecto	Del Proyecto	Equipo Técnico
Contexto Cada una de las tareas que es planificada durante el proyecto tiene un plazo definido y es predecesora para otra tarea, por lo que el atraso de una, lleva a que todas las restantes que se encuentran enlazadas a ella tengan retraso, y en definitiva el proyecto completo.			
Estrategias de mitigación o Monitoreo permanente de cada una de las tareas y actividades. o Aviso al responsable cuando no se cumpla el plazo de entrega.			
Plan de Contingencia o Replanificación de las tareas para minimizar el impacto. o Reasignar requerimientos según conocimientos y disponibilidad del usuario.			
Trigger o No entrega respuesta el día planificado.			

5.4 HERRAMIENTA DE DESARROLLO

Como fue mencionado con anterioridad en el Estudio de Factibilidad, CAEP cuenta con las licencias para el desarrollo de software con la herramienta de Microsoft Visual Studio

.NET, con servidor HP ProLiant ML 370 con Windows 2003 Server, administrador de base de datos SQL SERVER 2000, IIS 6.0.

A continuación se entrega una descripción de la herramienta seleccionada.

5.4.1 Visual Studio .NET

[7]Visual Studio .NET es un completo conjunto de herramientas para construir e integrar aplicaciones ASP Web, XML Web Services, aplicaciones de escritorio y aplicaciones móviles. Se compone de Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J#, los cuales utilizan el mismo ambiente integrado de desarrollo (IDE), lo cual les permite compartir herramientas y facilita la creación de soluciones de lenguaje mixto. Adicionalmente, estos lenguajes potencian la funcionalidad del denominado .NET Framework, entorno de programación ofrecido por Microsoft, el cual provee acceso a tecnologías claves que simplifican el desarrollo de aplicaciones ASP Web y XML Web Services.

Visual J#: En el caso de Visual J#, esta corresponde a una herramienta de desarrollo para programadores que están familiarizados con la sintaxis del lenguaje Java y puedan construir aplicaciones y servicios que se ejecutarán exclusivamente sobre el .NET Framework. Visual J# ha sido desarrollado independientemente por Microsoft y no está avalado por Sun Microsystems, Inc. Entre sus características relevantes se encuentran:

Soporte de biblioteca de clases: Bibliotecas de clases independientemente desarrolladas para proveer la mayoría de la funcionalidad de las bibliotecas de clases JDK (Java Development Kit) nivel 1.1.4, y muchas de las clases del JDK 1.2 java.util

Propiedades, delegados y eventos: J# tiene soporte para consumir y mostrar propiedades, delegados y eventos al estilo .NET, mientras mantiene la consistencia con la sintaxis tradicional de Java.

Comentarios Javadoc: J# soporta comentarios de código al estilo Javadoc. Visual J# .NET, incluye una herramienta que los usuarios pueden invocar para generar

documentación HTML de las API basado en comentarios Javadoc (Herramienta de J2SE que permite documentar API a partir de comentarios en el código fuente).

Asistente para migrar Visual J++: los desarrolladores de Visual J++ podrán migrar sus proyectos a Visual J# .NET al mismo tiempo de contar con ayuda sensible al contexto para relacionado a cualquier problema potencial.

Convertidor Binario: Esta herramienta convierte archivos Java bytecode (lenguaje de máquina de Java) a assemblies de Microsoft .NET para utilizarlos en aplicaciones .NET.

Visual C# .NET: Es un set de herramienta para crear XML Web Services, y aplicaciones .NET para Windows y la Web utilizando el lenguaje de desarrollo orientado a componentes C#. Pretende convertirse en un lenguaje moderno para desarrolladores con experiencia en C++ o Java. Visual C# entrega mayor funcionalidad incluyendo:

o Diseño rápido, desarrollo y soporte de explotación creando y consumiendo Web Services.

o Diseñadores de formularios y controles visuales para crear aplicaciones basadas en Windows.

o Herramientas de confección y servicios para construir soluciones poderosas basadas en el lado del servidor.

o Herramientas de migración para convertir proyectos basados en Java al ambiente de desarrollo Microsoft .NET.

Visual C++ .NET: Conjunto de herramientas para la construcción de aplicaciones basadas en MS Windows y MS .NET, aplicaciones Web dinámicas XML Web Services utilizando el lenguaje de programación C++. Ofrece características que ayudan a optimizar el proceso de programación de software con C++. Es un lenguaje basado en estándares y se pueden aprovechar los conocimientos de este lenguaje en varias comunidades y entornos informáticos.

Visual Basic .NET: Ofrece a los programadores orientados a las tareas, una sintaxis humanamente comprensible, una interfase de usuario intuitiva, herramientas y asistentes mejorados que aceleran el desarrollo de aplicaciones Microsoft .NET. Algunas características que lo distinguen son:

Definición de Variables por Omisión: no requiere que las variables sean inicializadas previo a su uso.

Tipificación Implícita: No requiere que especificar el tipo de variable antes de utilizarla.

Funcionamiento Enumerado: Ofrece un comportamiento más intuitivo utilizando tipos de datos enumerados.

Acceso público por omisión: Por defecto los miembros de una clase Visual Basic .NET son declarados para ser públicos, lo que permite una sintaxis más intuitiva.

Utilización de miembros compartidos: Pueden se accesados en VB .NET a través del nombre, la instancia de la variable del tipo al cual pertenecen o ambos.

Bloques de captura filtrada: Ofrece un estructurado y flexible manejo de excepciones de errores. Permite a los desarrolladores filtrar los errores basados en la clase de la excepción, cualquier expresión condicional o un número de error explícito.

Compilación en Background: Trabaja detrás de la escena, silenciosamente compila el código mientras se trabaja. De esta manera los desarrolladores inmediatamente detectan si tienen errores en el código.

Editor de código mejorado: Automáticamente formatea el código mientras se escribe, ahorrando tiempo. Puede automáticamente alinear el código a la indentación correcta, colocar mayúsculas a palabras claves y variables, colocar un “Then” olvidado a una instrucción “If”, entre otros.

Adicionalmente, el ambiente integrado de desarrollo Visual Studio .NET, incluye:

Herramientas para desarrollar aplicaciones para dispositivos inteligentes: como el Pocket PC, teléfonos celulares y PDAs, que se ejecutarán sobre un .NET Compact Framework.

RAD (Rapid Application Development) para Servidor: Server Explorer y varias herramientas incorporan herramientas de aplicaciones del lado del servidor. Esto permite acceder e integrar herramientas de administración del servidor, registro de eventos, bases de datos y servicios Web XML.

Diseñador HTML: Para programar HTML, ASP y MS ASP.NET Web Forms de manera gráfica sin tener que escribir HTML o scripts.

Crystal Reports: Para crear reportes muy completos e interactivos para aplicaciones basadas en Windows, Web y dispositivos móviles así como para Servicios Web.

Herramientas visuales de bases de datos: para diseñar visualmente stored ooleanas , ooleanas s, tablas, triggers, funciones definidas por el usuario y otros elementos de bases de datos.

Microsoft Visio®: Para capturar y comunicar requerimientos de negocio a través de modelamiento de bases de datos conceptuales, lógicas y físicas.

Microsoft Visual SourceSafe® 6.0c: Para proporcionar un ambiente colaborativo de desarrollo en equipo con control de versiones de código fuente, componentes, aplicaciones y documentos de diseño.

Visual Studio Analyzer: Para localizar cuellos de botella de desempeño de manera visual en aplicaciones COM distribuidas.

Microsoft BizTalk™ Server: Para construir procesos de negocios distribuidos que integran aplicaciones y socios. Para establecer rápidamente relaciones comerciales confiables y seguras con clientes y socios a través de la Web

Si bien es cierto que a través de Microsoft Visual Studio .NET, se está cambiando la filosofía de desarrollo respecto a sus versiones predecesoras, aún mantiene características

vinculadas especialmente al lenguaje Visual Basic .NET que permiten a desarrolladores que manejaron tecnologías anteriores sentirse capacitados para implementar soluciones con esta tecnología y al mismo tiempo motivados por sacar el máximo de provecho a las nuevas características.

Características Claves

Las características claves de este producto radican en el denominado .NET Framework.

El .NET Framework es un ambiente multilenguaje para construir, explotar y ejecutar XML Web services y aplicaciones. Consiste de tres partes principales:

Common Language Runtime (CLR): Se encarga de gestionar la ejecución del código y provee servicios para facilitar el desarrollo de componentes. A pesar de su nombre, el tiempo de ejecución (runtime) actualmente tiene ambos roles de soportar componentes en tiempo de ejecución y situaciones en tiempo de desarrollo, administrando en ambos casos la memoria y resolviendo la dependencia que puedan tener los componentes, entre otras tareas que resultan, incluso, en código reutilizable.

Unified Programming Classes: El denominado framework provee a los desarrolladores de un conjunto de librerías de clases (APIs) que están unificadas, orientadas a objetos, jerárquicas y extensibles. Actualmente, los programadores de C++ utilizan Microsoft Foundation Classes y los programadores de Java utilizan el Windows Fundation Classes. El framework unifica esos modelos dispares y les da a los programadores Visual Basic y Jscript acceso a bibliotecas de clases. Todos los lenguajes de programación desde Jscript hasta C++ tienen acceso similar al

ASP .NET: Construye sobre las clases de programación del .NET Framework, suministrando un modelo de aplicaciones Web con un conjunto de controles e infraestructura que simplifica construir aplicaciones Web. Utilizando las características XML Web Services, los desarrolladores de ASP .NET pueden escribir su lógica del negocio y utilizar la infraestructura de ASP .NET para entregar este servicio a través de SOAP (Simple Object Access Protocol).

Adicionalmente .NET Framework incluye:

Asistente de Migración: Un asistente permite migrar a .NET aplicaciones implementadas en Visual Studio 6, Visual C++, Visual InterDev, Visual J++ 6.0

MSDN: Una completa biblioteca MSDN, que contiene toda la documentación sobre estas herramientas de desarrollo. Más un sitio completo dedicado exclusivamente a compartir los conocimientos con otros desarrolladores del mundo y donde se encuentra la información oficial sobre características de Visual Studio .NET.

5.5 ANÁLISIS

El propósito de esta etapa es entregar el detalle para la implementación del sistema "Mantenedor de Contratos". Esta basada en la especificación de los requerimientos mediante la construcción de los casos de uso que están compuestos de tres partes:

Diagrama: detalle del diagrama general de casos de uso
Descripción: Especificación formal del requerimiento
Prototipo: Especificación de la interfaz que se deberá implementar

5.5.1 Alcance

El sistema esta enfocado en el control y mantención de los contratos con proveedores, además de proporcionar acceso controlado a la información de estos contratos. Este sistema será un modulo del sistema integrado, por lo tanto no esta contemplado la confección de funcionalidades que permitan manejar los perfiles, proveedores y usuarios, estos serán obtenidos del enlace con el sistema integral, el cual contempla estas funcionalidades.

5.5.2 Diagrama de Contexto

Mediante el siguiente diagrama se explica el ámbito del sistema:

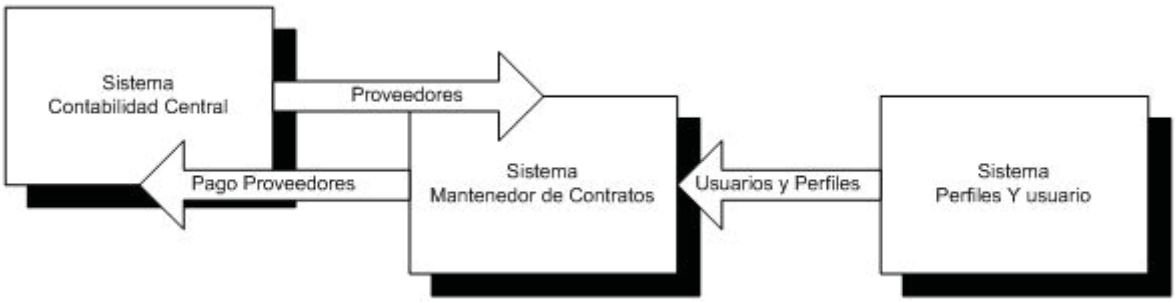


Ilustración 5.2 Diagrama de Contexto

5.5.3 Requerimientos Específicos

5.5.3.1 Identificación de Actores

Nombre Actor	**Cargo**	**Perfil**
Encargado	Jefes	Personal que tendrá a cargo la mantención de uno o más contratos. Es el encargado de velar por el cumpliendo de las obligaciones del contrato. Solo tiene jurisdicción sobre los contratos que son de su responsabilidad.
Web Master	*Analista Programador*	Responsable de mantener actualizada y correcta la información que el sistema entregará a los funcionarios.
Auditor	*Contraloría*	Personal encargado de revisar los

	Administrador	pagos y la mantención de los contratos

5.5.3.2 Requerimientos Funcionales

A. Pago de Proveedores con Contrato: realizar el pago de facturas de los contratos con los proveedores; debe generar un comprobante de pago con un identificar único que permita obtener los datos directamente en el sistema de contabilidad.

B. Buscador de Contratos: facilita la búsqueda de contratos en el sistema, limitado las acciones según los privilegios del usuario.

C. Ingresar contratos: permite el ingreso de la información de los contratos y su asignación a un responsable o encargado.

D. Indicadores Económicos: los contratos manejan fórmulas las cuales se utilizan para calcular el valor de la cuota a pagar, éstas fórmulas están asociadas a indicadores económicos como son UF o IPC. El indicador económico debe contener el valor ya sea mensual o diario según corresponda. Se debe permitir el ingreso, eliminación y actualización de los valores. Cabe destacar que no se podrá eliminar un valor asociado a un pago, es decir anteriores a la fecha de actual.

E. Formulas: mantener las fórmulas asociadas a los cálculos de cuotas, por ejemplo, promedio de los últimos 3 IPC.

F. Actualizar contrato: debe permitir renovar y modificar los datos de un contrato. La renovación se realiza aumentado el plazo de vigencia del contrato.

G. Bitácora de Pago Proveedor: proporcionar un listado de los pagos asociados a un contrato.

5.5.3.3 Requerimientos No Funcionales

A. Desarrollo Modular: efectuar un desarrollo modular que permita la evolución del mantenedor ante nuevas necesidades.

B. Alineación de columnas: todas las columnas que contengan Números en su despliegue deben estar a nivel central, los textos alineados a la izquierda.

C. Tiempos de respuestas: los tiempos de respuesta de los procedimientos almacenados en combinación con el despliegue de la información en pantalla no debe superar el minuto.

D. Listados Superior a 15 registros: los listados desplegados deben estar paginados, mostrar 15 registros por página.

E. Detalles Gráficos: los Input deben tener el largo del registro que contiene, por ejemplo, un campo Código no puede tener el mismo largo que un campo que contiene una descripción.

5.5.3.4 Diagrama de Casos de Uso General

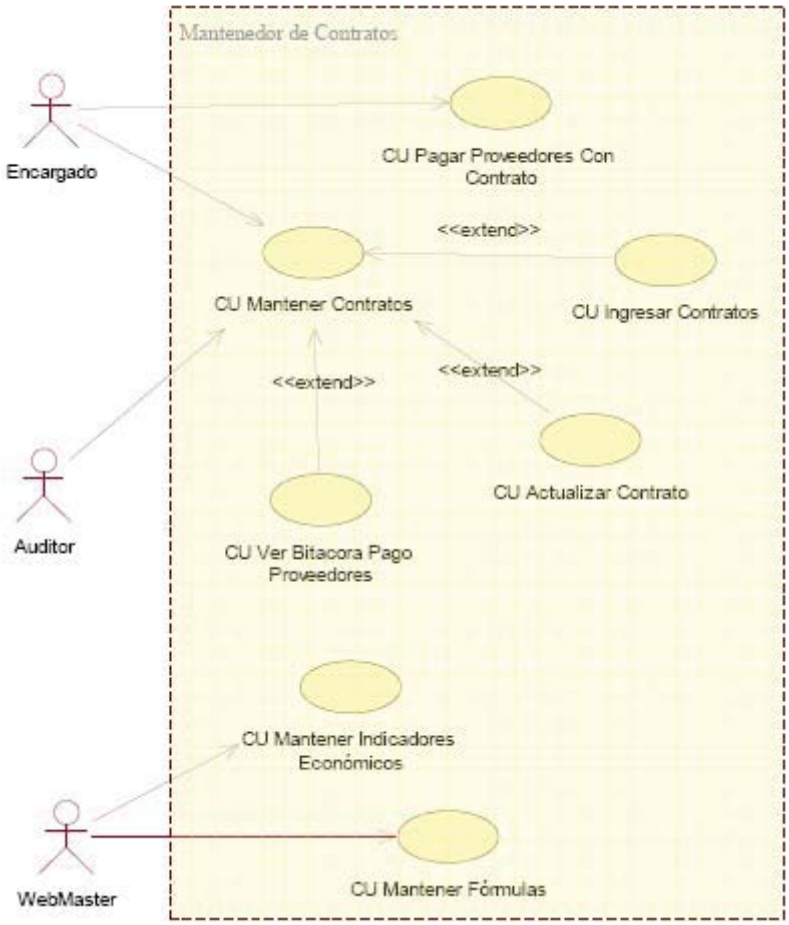


Ilustración 5.3 Diagrama de Casos de Uso

5.5.3.5 Convenciones

Casos de uso	
(*)	Campo de Ingreso Obligatorio
(B)	Campo con Buscador, se debe seleccionar información desde el buscador
Combo	Lista de selección de datos.
Check	Opción de selección Múltiple o Exclusivo
Lectura	Dato no Modificable
Formato Fecha	Dato Formateo automático
Texto	Dato Formato texto
Entero	Dato Tipo entero sin decimales
Numero	Dato Tipo Numérico con dos decimales
Ruta	Path de Archivo
DAS/M	Despliegue Automático sin posible Modificación.
DAC/M	Despliegue Automático con posible Modificación.
Prototipo	
	Opción Actualizar
	Opción Ver Calendario
	Opción Bitácora
	Menos Un Mes o Año
	Mas Un Mes o Año

5.5.3.6 Casos de Uso

5.5.3.6.1 CU Mantener Contratos

a) Diagrama

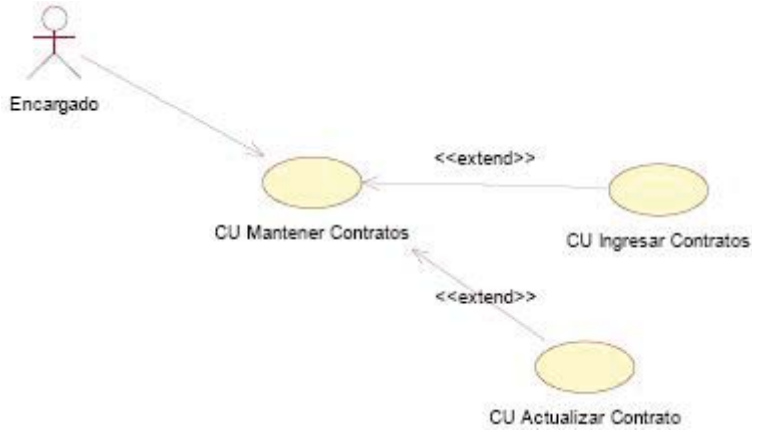


Ilustración 5.4 Diagrama C.U. Mantenedor Contratos

b) Descripción

Caso de Uso:	CU Mantener de Contratos
Actores:	Auditor, Encargado
Propósito	Ingresar, Actualizar y Renovar los Contratos de Proveedor con la institución.
Resumen:	El caso de uso comienza cuando el usuario selecciona opción Contratos, el sistema despliega un buscador, el usuario ingresa parámetros de búsqueda, el sistema despliega listado correspondiente a los datos ingresados, el usuario revisa listado y selecciona un contrato, el sistema despliega detalle con los datos del contrato.
Tipo:	Primario
Pre condición:	Usuario autentificado
Post condición:	
Referencias Cruzadas:	CU Ver Bitácora Pago Proveedores, CU Eliminar

Escenario principal

Acción(Actor)	Respuesta(Sistema)
1. El usuario ingresa al Mantenedor de Contratos, previamente autenticado como usuario.	2. El sistema despliega buscador de personas con las parámetros:
3. El usuario ingresa filtros de búsqueda y selecciona opción Buscar.	

Nombre Dato	Validación
Rut Proveedor	Texto
Nombre Proveedor	Texto
Caducidad (Inicio - Fin)	Formato Fecha
Estado	Combo
Responsable	Texto
Pagos	Combo
Opción Buscar	
Opción Nuevo Contrato.	

	4. El sistema rescata los parámetros ingresados. 5. El sistema efectúa consulta según datos ingresados. 6. El sistema despliega un listado con : o Identificación Contrato o Descripción o Rut Proveedor o Nombre Proveedor o Fecha Caducidad o Responsable o Opción Actualización o Opción Eliminación. o Opción Bitácora Las opciones Actualización y Eliminación solo se mostrarán cuando el contrato sea de la jurisdicción del encargado.
7. El usuario revisa la información desplegada. 8. El usuario selecciona opción Actualización de un contrato determinado.	9. El sistema rescata identificación del Contrato seleccionado. 10. El sistema consulta la información del Contrato. 11. El sistema despliega detalle del Contrato, que contiene:

Nombre dato	Validación
Identificación	Lectura
Descripción	Texto
Clasificación	Combo
Rut Proveedor	Texto (con validador de Rut)
Nombre Proveedor	Texto
Inicio Vigencia	Formato Fecha
Fin vigencia : Fecha Caducidad	Formato Fecha
Monto Total Contrato	Numero
Moneda	Combo
Documento de Pago	Combo
Fórmula Reajuste	Combo
Periodo de Pago	Combo
Día de Pago Mensual	Entero
Responsable	
Sección	Combo
Funcionario	Combo
Renovación automática	Check
Aviso Caducidad (Nro. De días de aviso anticipado a la caducidad)	Entero
Estado	Combo
Opción Renovar	
Opción Actualizar	
Opción Volver	

12. El usuario modifica los datos.
13. El usuario selecciona opción Actualizar.

14. El sistema rescata los datos.
15. El sistema verifica que los datos se encuentren completos y correctos.
16. El sistema registra los datos.
17. El sistema envía mensaje "*Actualización exitosa.*"

Extensiones

Nuevo Contrato - C.U. Ingresar Contrato

3.1 El usuario selecciona opción Nuevo Contrato.

3.1.2 El sistema despliega formulario con los siguientes campos a ingresar:

Nombre dato	Validación
Identificación	DAS/M
Descripción	Texto
Clasificación	Combo
Rut Proveedor	Texto (con validador de Rut)
Nombre Proveedor	Texto
Inicio Vigencia	Formato Fecha
Fin vigencia : Fecha Caducidad	Formato Fecha
Monto Total Contrato	Numero
Moneda	Combo
Documento de Pago	Combo
Fórmula Reajuste	Combo
Opción Ver Fórmula	
Periodo de Pago	Combo
Día de Pago Mensual	Entero
Responsable	
Sección	Combo
Funcionario	Combo
Renovación automática	Check
Aviso Caducidad (Nro. De días de aviso anticipado a la caducidad)	Entero
Estado	Combo
Opción Grabar	
Opción Volver	

3.1.3 El usuario ingresa los datos solicitados.
3.1.4 El usuario selecciona opción Grabar.
3.1.5 El sistema rescata los datos ingresados.
3.1.6 El sistema verifica el ingreso de todos los datos.
3.1.7 El sistema graba los datos.
3.1.8 El sistema envía mensaje "Ingreso Exitoso".

8.1 El usuario selecciona la Opción Eliminar Contrato.
8.1.2 El sistema despliega el siguiente mensaje:
"Esta seguro de eliminar el contrato <Id Contrato> (Si/No)
8.1.3 El usuario selecciona la opción Si
8.1.4 El sistema comprueba realiza las siguientes validaciones:
8.1.4.1 El contrato no este tenga pagos pendientes.
8.1.4.2 El contrato se encuentre Inactivo.

8.1.5.1 El contrato es valido para eliminación.
8.1.5.1.1 El sistema cambia el estado del contrato a eliminado
8.1.5.1.2 El sistema despliega el mensaje "Eliminación Exitosa".

8.1.5.2 El contrato no es valido para su eliminación
8.1.5.2.1 el sistema despliega el siguiente mensaje : "No se puede eliminar el contrato: <Contrato posee pagos pendientes> <Contrato esta Activo>"

8.1.6 El sistema vuelve al Buscador de Contratos.

8.3 El Usuario selecciona la opción Bitácora
8.3.1 Se gatilla caso de uso CU Bitácora de Pago Proveedores

13.1 Opción Renovar Contrato
13.1.1 El usuario selecciona opción Renovar Contrato.
13.1.2 El sistema muestra ventana tipo pop- up que contiene:

Nombre dato	Validación
Descripción	Texto
Fecha Inicio	Formato Fecha
Fecha Fin	Formato Fecha
Monto	Entero
Moneda	Combo
Fecha de Renovación	Fecha del día
Opción Renovar	

13.1.3 El usuario digita los datos y selecciona opción Renovar.
13.1.4 El sistema valida el ingreso de todos los datos.
13.1.5 El sistema graba los datos nuevos asociados a la renovación y los anteriores quedan como parte de la historia del contrato.
13.1.6 El sistema actualiza la información desplegada en el formulario del contrato.
13.1.7 El sistema envía mensaje "Renovación exitosa" y cierra la ventana.
13.1.8 El sistema vuelve al formulario del Contrato con la información actualizada.

12.1 El usuario selecciona la opción Ver Fórmula
12.1.1 El sistema abre una ventana emergente con la siguiente información de la fórmula:

Nombre dato	Validación
Nombre Fórmula	Lectura
Indicador Económico	Lectura
Promedio	Lectura
Mes	Lectura
Incluir Mes Actual	Lectura
Mismo Mes	Lectura
Mes Anterior	Lectura
Opción Volver	

12.1.2 El usuario selecciona la opción Volver 12.1.3 El sistema cierra la ventana. 15.1 El sistema valida el ingreso de datos 15.1.1 Datos obligatorios. El sistema detecta que existen campos obligatorios en blanco, correspondientes a Descripción, Clasificación, Rut Proveedor, nombre proveedor, Fechas Inicio y Fin, Monto Moneda, Documento de Pago, Día de Pago, Fórmula Reajuste, Periodo de Pago, Responsable, Aviso de Caducidad. 15.1.2 Validaciones según el tipo de dato. 15.1.3 Validaciones particulares definidas para cada dato. 15.1.3.1 El Rut y el Nombre debe corresponder al mismo Proveedor 15.1.3.2 El Rut y el Nombre del proveedor deben existir. 15.1.3.3 Fecha Inicio debe ser menor que la Final. 15.1.3.4 Aviso de caducidad debe estar dentro del rango de vigencia del contrato. 15.2 El sistema envía mensajes según la validación aplicada. 15.2.1 "<*Dato N*> es obligatorio, por favor ingréselo". 15.2.2 "<*Dato N*> no corresponde al formato solicitado, por favor ingréselo nuevamente". 15.2.3 "<*Dato N*> con problemas, por favor verifique <*Descripción del problema*>".	
Requisitos especiales	En el listado de contratos estos presentarán tres tipos de colores en las fuente, que permitirán observar la proximidad de la fecha de caducidad del contrato, alertando al usuario a tomar las medidas correspondientes: Verde, para contratos que su fecha de caducidad se encuentre a mas de 6 meses Amarillo para los contratos que se encuentren entre los 6 y 3 meses. Rojo para los contratos cuya fecha de caducidad se encuentra a menos de tres meses de la fecha actual.
Tecnología y variaciones de datos	

c) Prototipo

Ilustración 5.5 Prototipo Buscador Contratos

Listado Contratos						
Id Contrato	Descripción	Rut Proveedor	Nombre Proveedor	Fecha Caducidad	Responsable	Acción
12	Mantencion Servicios Instalados	5.442.369-8	ADT Segurity Services	15/12/2005	Juan Reyes	
10	Transporte de carga	70.598.635-6	Lan CurrieR	30/10/2006	Juan Reyes	

Ilustración 5.6 Prototipo Listado Contratos

Nuevo Contrato				
Identificación				
Descripción				
Clasificación	Seleccione			
Rut Proveedor				
Nombre Proveedor				
Inicio Vigencia				
Fin Vigencia				
Monto Total Contrato				
Moneda	Seleccione			
Documento de Pago	Seleccione			
Formula Reajuste	Seleccione	*Ver Formula*		
Periodo de Pago	Seleccione			
Día de Pago Mensual	Dia			
Responsable	Sección	Seleccione	Funcionario	Seleccione
Renovación automática	SI	No		
Aviso Caducidad		Dias		
Estado	Seleccione			
Grabar	*Volver*			

Ilustración 5.7 Prototipo Buscador Contratos

Actualizar Contrato			
Identificación	10		
Descripción	Mantencion Servicios Instalados		
Clasificación	Mantención		
Rut Proveedor	77.658.325.-9		
Nombre Proveedor	ADT Segurity Services		
Inicio Vigencia	31/12/2004		
Fin Vigencia	31/12/2006		
Monto Total Contrato	100		
Moneda	UF		
Documento de Pago	Factura		
Formula Reajuste	Ninguno		
Periodo de Pago	Trimestral		
Día de Pago Mensual	Dia 15		
Responsable	**Sección**	Administración	**Funcionario** Juan Reyes
Renovación automática	SI No		
Aviso Caducidad	60 Dias		
Estado	Activo *Renovar*		
Actualizar	*Volver*		

Ilustración 5.8 Prototipo Buscador Contratos

Renovación Contrato	
Descripcion	Mantencion Servicios Instalados
Fecha Inicio	
Fecha Fin	
Monto	
Moneda	Seleccione
Fecha Fenovación	15/09/2005
Actualizar	*Volver*

Ilustración 5.9 Prototipo Renovación Contratos

Formulas	
Nombre Formula	Uf Mes Anterior
Indicador	UF
Promedio	☐
Meses	
Incluir Mes Actual	☐
Mes Anterior	☑
Mismo Mes	☐
Volver	

Ilustración 5.10 Prototipo Visor Formulas

5.5.3.6.2 CU Pagar Proveedores con Contrato

a) Diagrama

Ilustración 5.11 Diagrama C.U. Pagar Proveedores con Contratos

b) Descripción

Caso de Uso:	CU Pagar Proveedores con Contrato
Actores:	Encargado
Propósito	Dar el visto bueno al pago de una factura
Resumen:	El caso de uso comienza cuando el usuario selecciona la opción Pago Proveedores con Contrato, el sistema le despliega los Pagos pendientes para su aprobación.
Tipo:	Primario
Pre condición:	Contrato creado
Post condición:	Pago Aprobado o Rechazado
Referencias Cruzadas:	

Escenario principal

Acción(Actor)	Respuesta(Sistema)
1. El usuario ingresa al Pago Proveedores con Contrato previamente identificado como usuario	2. El sistema muestra un listado con los siguientes campos:
3. El usuario selecciona la opción Pago.	Los contratos listados son los de exclusiva responsabilidad del usuario.

Nombre dato	Validación
Id. Contrato	Lectura
Proveedor	Lectura
Descripción	Lectura
Fecha vencimiento Pago	Lectura
Estado	Lectura
Opción de Pago	

	4. El sistema rescata el código y presenta una nueva pantalla que contiene la siguiente información:

Nombre dato	Validación
Identificación	Lectura
Descripción	Lectura
Rut Proveedor	Lectura
Nombre Proveedor	Lectura
Inicio Vigencia	Lectura
Fin vigencia : Fecha Caducidad	Lectura
Monto	Lectura
Moneda	Lectura
Documento de Pago	Lectura
Periodo de Pago	Lectura
Responsable	
Sección	Lectura
Funcionario	Lectura
Renovación automática	Lectura
Estado	Lectura
Detalle del Pago	
Numero Factura	Texto
Periodo	
Fecha Inicio	DASM
Fecha Fin	DASM
Monto A Cancelar	texto
Opción Aceptar	
Opción Volver	

5. El usuario verifica la información presentada y la compara con la factura. 6. El usuario aprueba el pago.	
	7. El Sistema almacena el pago como aceptado y genera un comprobante con un Número correlativo el cual será adjuntado a la factura, la que será enviada por mano a Tesorería para generar la Orden de Egreso para realizar el pago. El sistema de contabilidad reconocerá este número de comprobante y obtendrá la información del pago correspondiente.
Extensiones	
Requisitos especiales	
Tecnología y variaciones de datos	

c) Prototipo

Pago Proveedores con Contrato					
Identificación Contrato	**Proveedor**	**Descripción**	**Fecha Vencimiento Pago**	**Estado**	**Acción**
12	ADT Segurity Services	Mantencion Servicios Instalados	15/12/2005	Pendiente	Pagar
10	Lan CurrieR	Transporte de carga	30/10/2005	Rechazado	Pagar

Ilustración 5.12 Prototipo Pago Proveedores con Contrato

Pago Proveedor				
Identificación	12			
Descripción	Mantencion Servicios Instalados			
Rut Proveedor	77.658.325.-9			
Nombre Proveedor	ADT Segurity Services			
Inicio Vigencia	31/12/2004			
Fin Vigencia	31/12/2006			
Monto	100			
Moneda	UF			
Documento de Pago	Factura			
Periodo de Pago	Trimestral			
Responsable	**Sección**	Administración	**Funcionario**	Juan Reyes
Renovación automática	SI			
Estado	Activo			
Detalle Pago				
Numero factura				
Periodo Inicio	01/08/2005			
Periodo Fin	30/08/2005			
Monto Mensual Neto	1 (UF)			
Reajuste	0 ($)			
Impuesto - Iva 19%	10.088 ($)			
Monto A Cancelar	53.950 ($)			
Aceptar	***Rechazar***			

Ilustración 5.13 Prototipo Pago Proveedor

Comprobante de Pago Proveedores Nº 000012
Fecha Autorización 13/12/2005

Identificación Contrato 10
Mantencion Servicios Instalados
ADT Segurity Services

Juan Reyes - Administración

Ilustración 5.14 Prototipo Comprobante de Pago

5.5.3.6.3 CU Bitácora Pago Proveedores

a) Diagrama

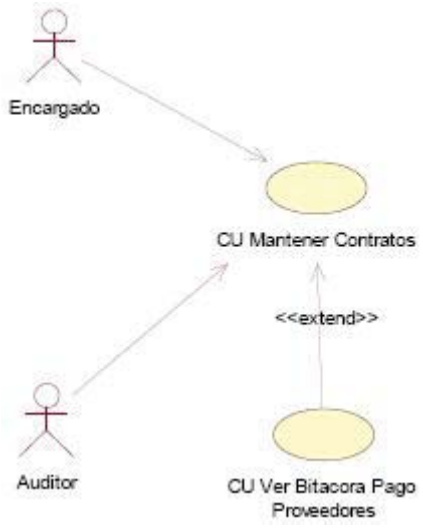


Ilustración 5.15 Diagrama C.U. Mantenedor Contratos

b) Descripción

Caso de Uso:	CU Ver Bitácora Pago Proveedores
Actores:	Auditor, Encargado
Propósito	Mostrar el Historial de pagos efectuados en un contrato
Resumen:	El caso de uso comienza cuando el usuario selecciona la opción Bitácora desde el CU Buscador Contratos, el sistema despliega un listado ordenado de los todos los pagos realizados para ese contrato.
Tipo:	Secundario
Pre condición:	Contrato Existente
Post condición:	
Referencias Cruzadas:	Buscador de Contratos

Escenario principal

Acción(Actor)	Respuesta(Sistema)
1. El usuario selecciona la opción ver bitácora desde el caso de uso Buscador Contratos.	2. el sistema rescata el código del contrato seleccionado y muestra una lista que tiene las siguientes columnas:
3. El usuario revisa el listado y selecciona opción Volver.	Al pie de este listado se muestra el total Cancelado. 4. El sistema regresa al CU Buscador Contratos

Nombre dato	Validación
Identificación Contrato	Lectura
Descripción	Lectura
Rut Proveedor	Lectura
Nombre Proveedor	Lectura
Fecha inicio	Lectura
Fecha fin	Lectura
Estado	Lectura
Fecha Vencimiento Pago	Lectura
Fecha Pagado	Lectura
Responsable	Lectura
Estado	Lectura
N° Factura	Lectura
Monto Pagado	Lectura
Opción Volver	

Extensiones	
Requisitos especiales	
Tecnología y variaciones de datos	

c) Prototipo

Bitacora Pago Proveedor	
Identificación	10
Descripción	Mantencion Servicios Instalados
Rut Proveedor	77.658.325.-9
Nombre Proveedor	ADT Segurity Services
Inicio Vigencia	31/12/2004
Fin Vigencia	31/12/2006
Estado	Activo

Fecha Caducidad	Responsable	Estado	Motivo
14/03/2005 11:30	Juan Reyes	Aprobado	
05/03/2005 17:30	Juan Reyes	Rechazado	Cálculo de reajuste erroneo

Volver

Ilustración 5.16 Prototipo Bitácora Pago Proveedor

5.5.3.6.4 CU Mantener Indicadores Económicos

a) Diagrama

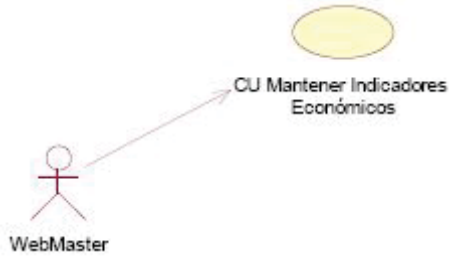


Ilustración 5.17 Diagrama C.U. Mantenedor Contratos

b) Descripción

Caso de Uso:	CU Mantener Indicadores Económicos
Actores:	Web Master
Propósito	Ingresar, Eliminar y Actualizar los distintos Indicadores económicos utilizados en los cálculos de montos y valores de los contratos y pagos.
Resumen:	El caso de uso comienza cuando el usuario selecciona la opción Indicadores Económicos, el sistema despliega un Listado con los indicadores económicos del sistema, el usuario selecciona un indicador, el sistema presenta un formulario para el ingreso de los valores, mensuales, diarios o únicos. El sistema rescata los datos y los almacena.
Tipo:	Primario
Pre condición:	
Post condición:	Índice Actualizado
Referencias Cruzadas:	

Escenario principal

Acción(Actor)	**Respuesta**(Sistema)
1. El Usuario selecciona la opción Indicadores Económicos.	2. El sistema despliega un Listado con los Indicadores Económicos:
3. El usuario selecciona la Opción Actualizar.	4. El Sistema presenta un calendario divido por meses o días de un mes según la periodicidad de ingreso del índice seleccionado.

Nombre dato	Validación
Nombre Indicador	Lectura
Periodicidad de Ingreso	Lectura
Opción Actualizar	Lectura
Opción Nuevo	Lectura

5. El usuario selecciona un año y/o un mes luego ingresa los valores para el periodo seleccionado y elige la opción Actualizar.	6. El sistema valida los datos ingresados. 7. El sistema almacena los cambios.

Nombre dato	Validación
Nombre	Lectura
Periodicidad de ingreso	Lectura
Año	Texto
Mes	Numero
Día	Numero
Opción Actualizar	

Extensiones

Nuevo Índice Económico

3.1 El usuario selecciona la opción Nuevo.

3.1.1 el sistema presenta el un formulario con las siguientes columnas:

Nombre dato	Validación
Nombre	Texto
Periodicidad de ingreso	Combo
Opción Grabar	

3.1.2 El usuario ingresa los datos.

3.1.3 El sistema valida los datos.

3.1.4 El sistema almacena los datos y vuelve al listado de Indicadores.

Requisitos especiales	
Tecnología y variaciones de datos	Los Datos ingresados en las tablas involucradas estarán disponibles mediante vistas para los demás sistemas de la institución, Intranet corporativa, Sistemas Prestamos, etc.

c) Prototipo

Mantenedor Indicadores Económicos

Nombre Indicador	Periodicidad de Ingreso	Acción
UF	Diario	
IPC	Mensual	
Iva 19%	Mensual	
		Nuevo

Ilustración 5.18 Prototipo Mantenedor Indica

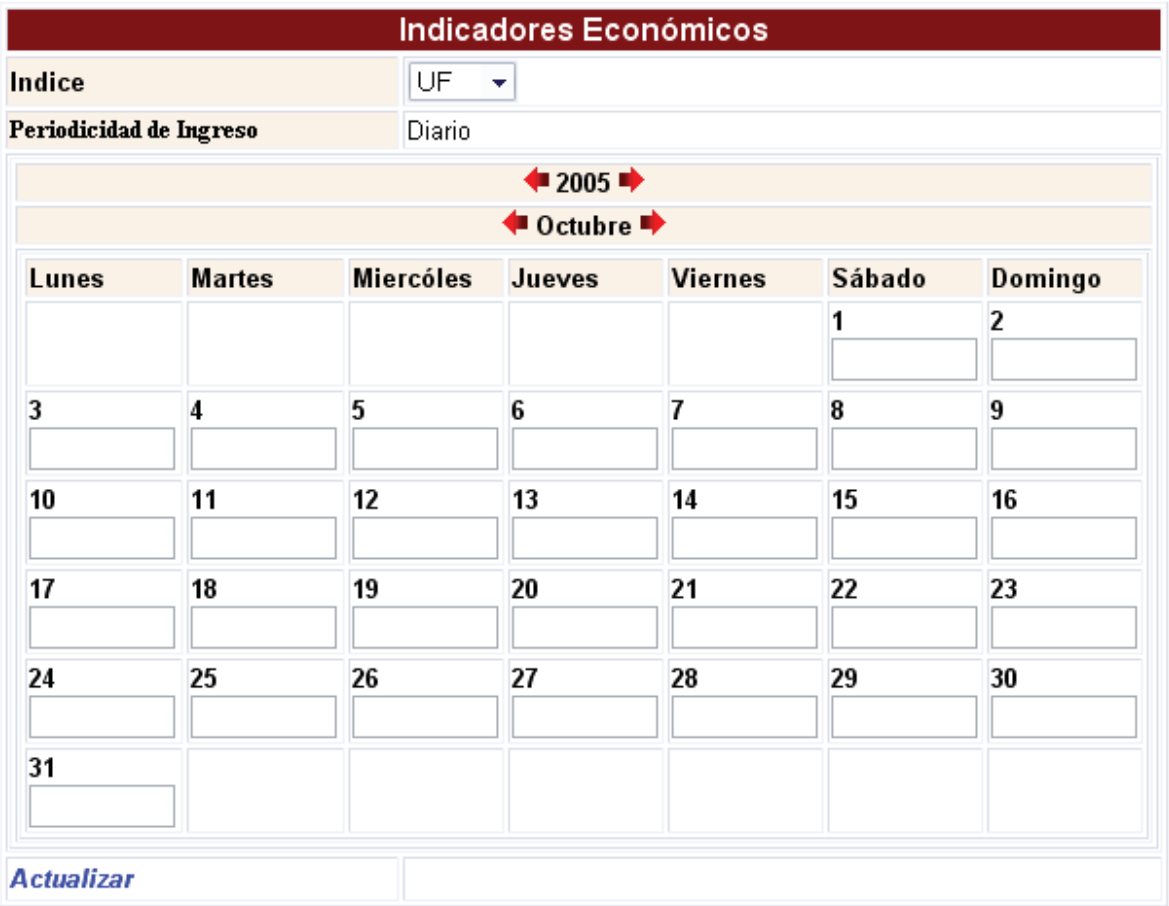


Ilustración 5.19 Prototipo Ingresar Indicador Económico Diario

Ilustración 5.20 Prototipo Indicador Económico Mensual

Ilustración 5.21 Prototipo Nuevo Índice Económico

5.5.3.6.5 CU Mantener Formulas

a) Diagrama

Ilustración 5.22 Diagrama C.U. Mantener Fórmulas

b) Descripción

Caso de Uso:	CU Mantener Fórmulas
Actores:	Web Master
Propósito	Ingresar, Eliminar, Actualizar las fórmulas para el cálculo de los reajustes e impuestos para los distintos pagos de facturas a proveedores.
Resumen:	El caso de uso comienza cuando el usuario selecciona la opción fórmulas, el sistema despliega un formulario para el ingreso de los datos. El usuario ingresa la fórmula y graba los datos en la BD.
Tipo:	Primario
Pre condición:	
Post condición:	Fórmula Almacenada
Referencias Cruzadas:	

Escenario principal

Acción(Actor)	Respuesta(Sistema)
1. El caso de uso comienza cuando el usuario selecciona la opción fórmulas.	2. El Sistema presenta un listado de las fórmulas existentes:

Nombre dato	Validación
Nombre Fórmula	Lectura
Opción Actualizar	
Opción Nueva Fórmula	

3. El usuario selecciona la opción Actualizar.

4. El sistema muestra un formulario con las siguientes columnas:

Nombre dato	Validación
Nombre Fórmula	Texto
Fórmula	
Indicador Económico	Combo
Promedio	Check
Meses	Numero
Incluir Mes Actual	Check
Mismo Mes	Check
Mes Anterior	Check
Opción Actualizar	
Opción Volver	

5. El usuario ingresa los datos y selecciona la opción Actualizar.	6. El sistema rescata los datos y almacena la fórmula.
Extensiones	

Nueva Fórmula

3.1.1 Usuario selecciona la opción Nueva Fórmula.

3.1.2 El sistema presenta un formulario para el ingreso de la nueva fórmula:

Nombre dato	Validación
Nombre Fórmula	Texto
Fórmula	
Indicador Económico	Combo
Promedio	Check
Mes	Numero
Incluir Mes Actual	Check
Mismo Mes	Check
Mes Anterior	Check
Opción Aceptar	
Opción Volver	

El ingreso de los datos se debe efectuar en el siguiente orden:

Seleccionar el Indicador económico a utilizar

Seleccionar dentro de estas tres opciones excluyentes:

Promedio: Si la fórmula corresponde al promedio de una cantidad x de indicadores. Con esta acción se habilita el ingreso de las siguientes opciones:

Mes: Cantidad de meses a considerar para el calculo del promedio.

Incluir mes actual: Seleccionar si se desea que en el cálculo del promedio se considere el mes Actual, que corresponderá al mes en el cual se realice el pago.

Mismo Mes: Si la fórmula utiliza el mismo mes para el calculo

Mes anterior: Si la fórmula utiliza el mes anterior para el cálculo.

3.1.3 El usuario ingresa los valores y selecciona la opción Aceptar.

3.1.4 El sistema graba los datos y vuelve al listado de fórmulas reflejando los cambios.

Requisitos especiales	
Tecnología y variaciones de datos	

c) Prototipo

Mantenedor Formulas

Nombre Formula	Acción
UF Mes Anterior	
IPC	
Iva 19%	

Nueva Formula

Ilustración 5.23 Prototipo Mantenedor Fórmulas

Actualizar Formula	
Nombre Formula	UF Mes Anterior
Indicador Económico	UF
Promedio	☐
Meses	
Incluir Mes Actual	☐
Mes Anterior	☑
Mismo Mes	☐
Actualizar	Volver

Ilustración 5.24 Prototipo Actualizar Fórmula

Nueva Formula	
Nombre Formula	UF Mes Anterior
Indicador Económico	UF
Promedio	☐
Meses	
Incluir Mes Actual	☐
Mes Anterior	☑
Mismo Mes	☐
Aceptar	Volver

Ilustración 5.25 Prototipo Nueva Fórmula

5.6 DISEÑO LÓGICO

En esta sección se expone el diseño lógico del sistema, el cual debe satisfacer los requerimientos funcionales del software, señalados por medio de los casos de uso.

5.6.1 Arquitectura del sistema

El sistema esta basado en la construcción de sitios Web de tres capas:

- Presentación
- Negocio
- Datos

Cada una de estas unidas esta destinada desarrollar funciones particulares y específicas:

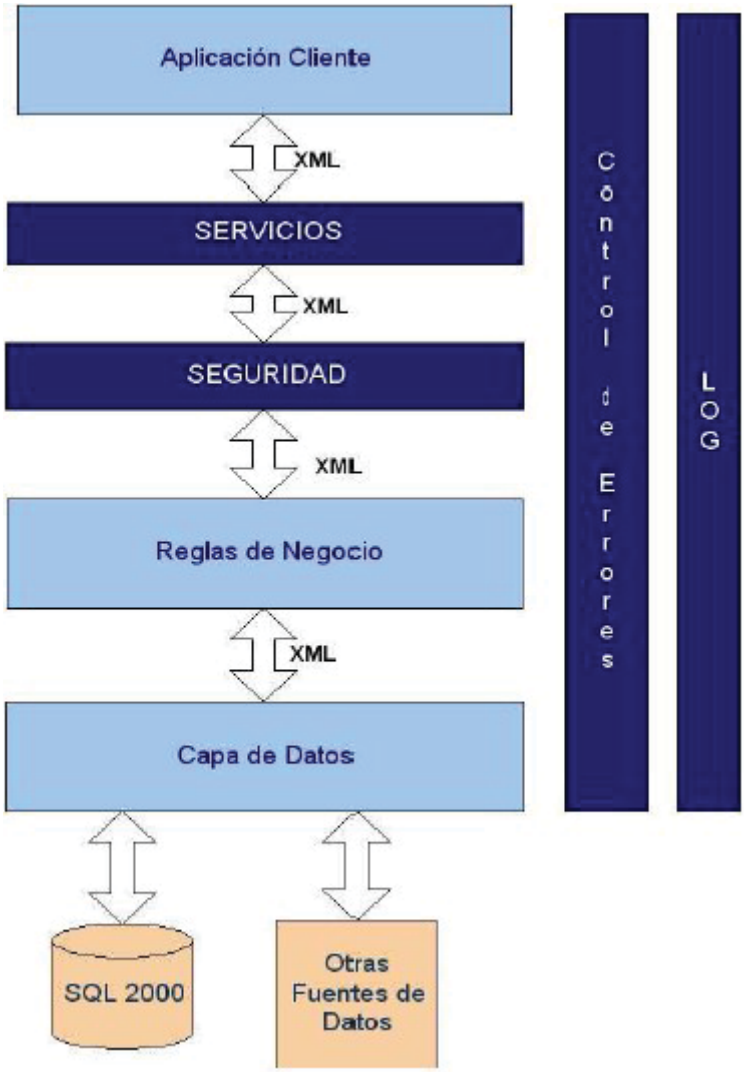


Ilustración 5.26 Arquitectura del Sistema

5.6.1.1 Descripción Capa de datos

Se entrega una descripción de la funcionalidad que debe poseer la Capa de Datos de la solución que se implementará, componente crucial para el funcionamiento del sistema, ya que su función es la de establecer la comunicación con las fuentes de datos que el sistema requiera.

5.6.1.2 Diagrama Solución

A continuación se presenta el esquema de solución que se implementará para la Capa de Datos de la arquitectura de solución.

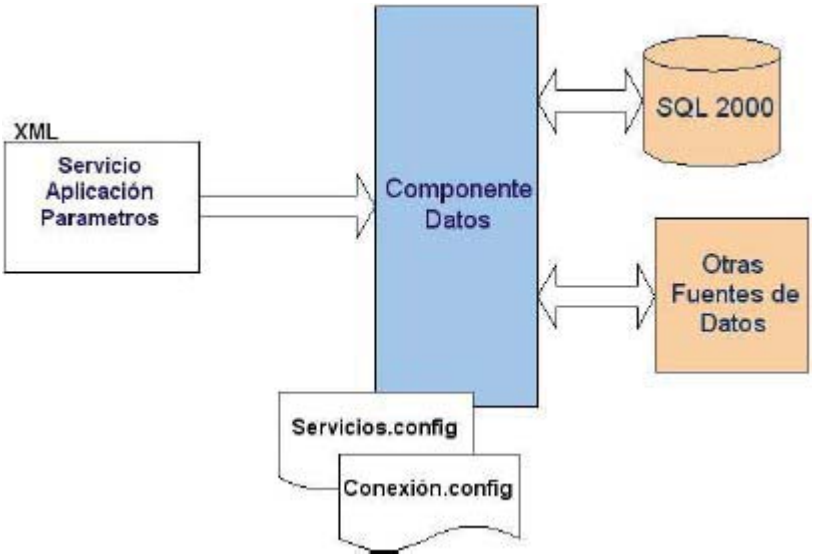


Ilustración 5.27 Especificación Capa de Datos

5.6.1.2.1 Descripción

A continuación se entrega una descripción de cada una de las partes que participan en el funcionamiento de la Capa de Datos del sistema, su objetivo y la información que entregan.

XML Entrada

La estructura XML corresponde al medio de comunicación que se utilizará entre las componentes de Negocio y Dato, la primera es la responsable de enviar la información necesaria para que se efectúen las consultas requeridas a la fuente de datos.

La estructura XML a utilizar es la siguiente:

```
<XML_Entrada>
  <Servicios>
        <Servicio>
              <Nombre></Nombre>
              <Conexión></Conexión>
              <Descripcion></Descripcion>
              <Parámetros>
                    <Parametro>
                          <Nombre></Nombre>
                          <Tipo></Tipo>
                          <Largo></Largo>
                          <Valor></Valor>
                    </Parametro>
              </Parámetros>
        </Servicio>
  </Servicios>
</XML_Entrada>
```

Componente Datos

La componente de datos debe ser capaz de leer el XML de entrada, con el tag '<Aplicación>' ir al archivo 'Conexión.config' y buscar el string de conexión a la fuente de datos correspondiente. Además con el tag '<Servicio><Nombre>' recuperar del archivo 'Servicios.config' el nombre del procedimiento almacenado que se ejecutará.

Después de recuperar la información necesaria debe ejecutar la consulta en la fuente de datos indicada, y recuperar los datos, formando un XML de salida.

La estructura XML que devuelve a la capa de Negocio es la siguiente:

```
<XML_Salida>
  <Servicios>
        <Servicio>
              <Nombre></Nombre>
              <Conexión></Conexión>
              <Descripcion></Descripcion>
              <Registros>
                    <Registro>
                          <Campos>
                                <Campo>
                                      <Nombre></Nombre>
                                      <Valor></Valor>
                                </Campo>
                          <Campos>
                    </Registro>
              </Registros>
        </Servicio>
  </Servicios>
</XML_Salida>
```

Servicios.config

Archivo encargado de almacenar el nombre de los procedimientos almacenados que se ejecutarán según el servicio que se haya solicitado desde la componente de negocio, su estructura es la siguiente:

```
<Servicios>
  <Servicio>
        <Nombre></Nombre>
        <Procedimiento></Procedimiento>
  </Servicio>
</Servicios>
```

Conexión.config

Archivo con estructura XML, encargado de registrar los String de conexión necesarios para las diferentes fuentes de datos.

```
<Conexiones>
  <Conexión>
        <Conexión></Conexión>
        <String></String>
  </Conexión>
</Conexiones>
```

Fuente Datos

Dentro de una aplicación pueden existir diferentes fuentes de información que entreguen a la aplicación los datos necesarios para su correcto funcionamiento. Dichas fuentes pueden ser Bases de Datos, archivos, Web Service, otras aplicaciones en diferentes lenguajes, entre otras.

5.6.2 Diagramas de Secuencia

Un diagrama de secuencia, es una representación que muestra, en un determinado escenario, los eventos generados por actores externos, su orden y los eventos externos del sistema.[10]

5.6.2.1 Diagrama de Secuencia 'CU Ver Bitácora Pago proveedores

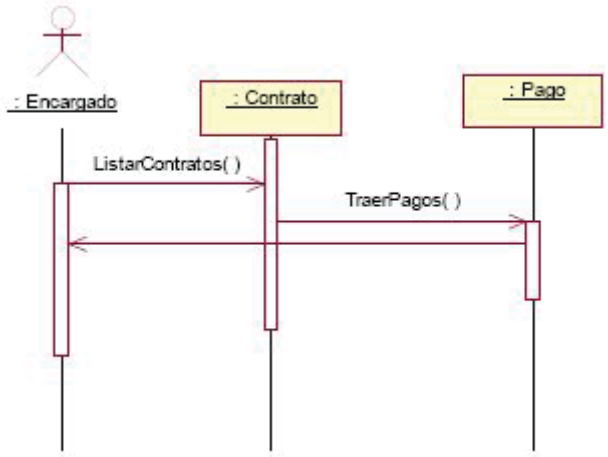


Ilustración 5.28 Diagrama de Secuencia Ver Bitácora Pago Proveedores

5.6.2.2 Diagrama de Secuencia ‘CU Mantener Contratos’

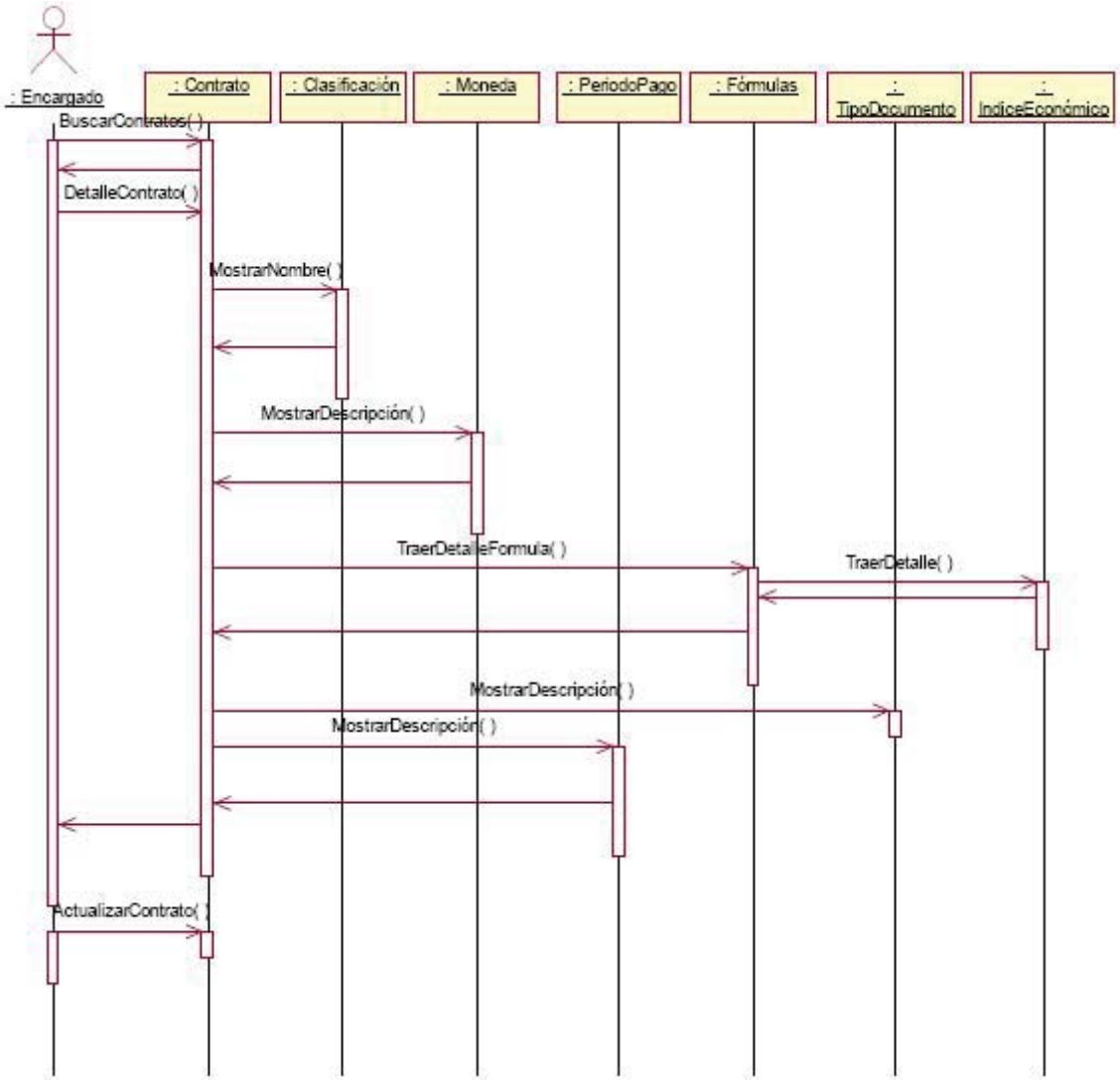


Ilustración 5.29 Diagrama de Secuencia Mantener Contratos

5.6.2.2.1 Extensión Nuevo Contrato

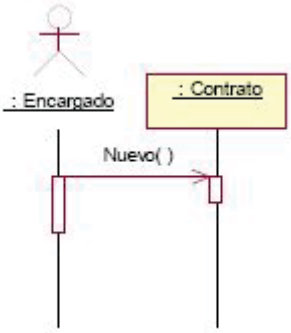


Ilustración 5.30 Diagrama de Secuencia Nuevo Contrato

5.6.2.3 Diagrama de Secuencia 'CU Mantener Fórmulas'

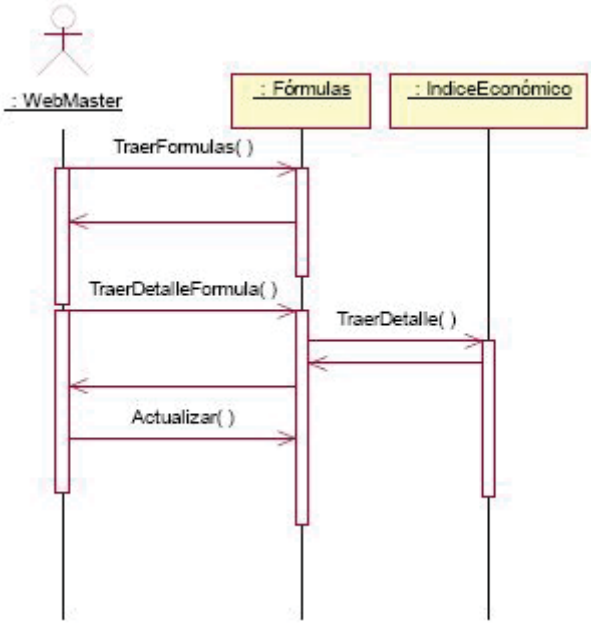


Ilustración 5.31 Diagrama de Secuencia Mantener Fórmulas

5.6.2.4 Diagrama de Secuencia 'CU Mantener Indicadores Económicos'

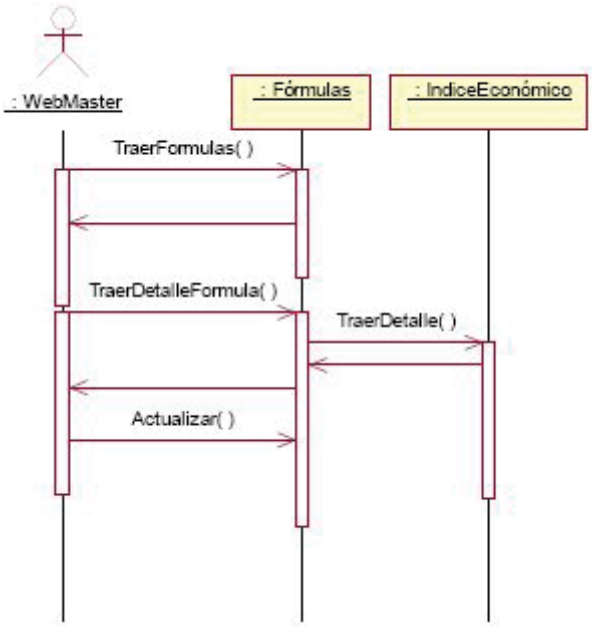


Ilustración 5.32 Diagrama de Secuencia Mantener Indicadores Económicos

5.6.2.5 Diagrama de Secuencia ‘CU Pagar Proveedores con Contrato’

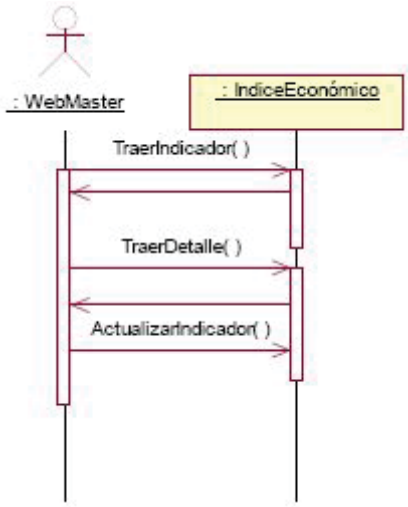


Ilustración 5.33 Diagrama de Secuencia Pagar Proveedores con Contrato

5.6.3 Diagrama de Clases

Un diagrama de clases, ilustra la definición de clases que será implementado en el software, identificando las clases, métodos y asociaciones.

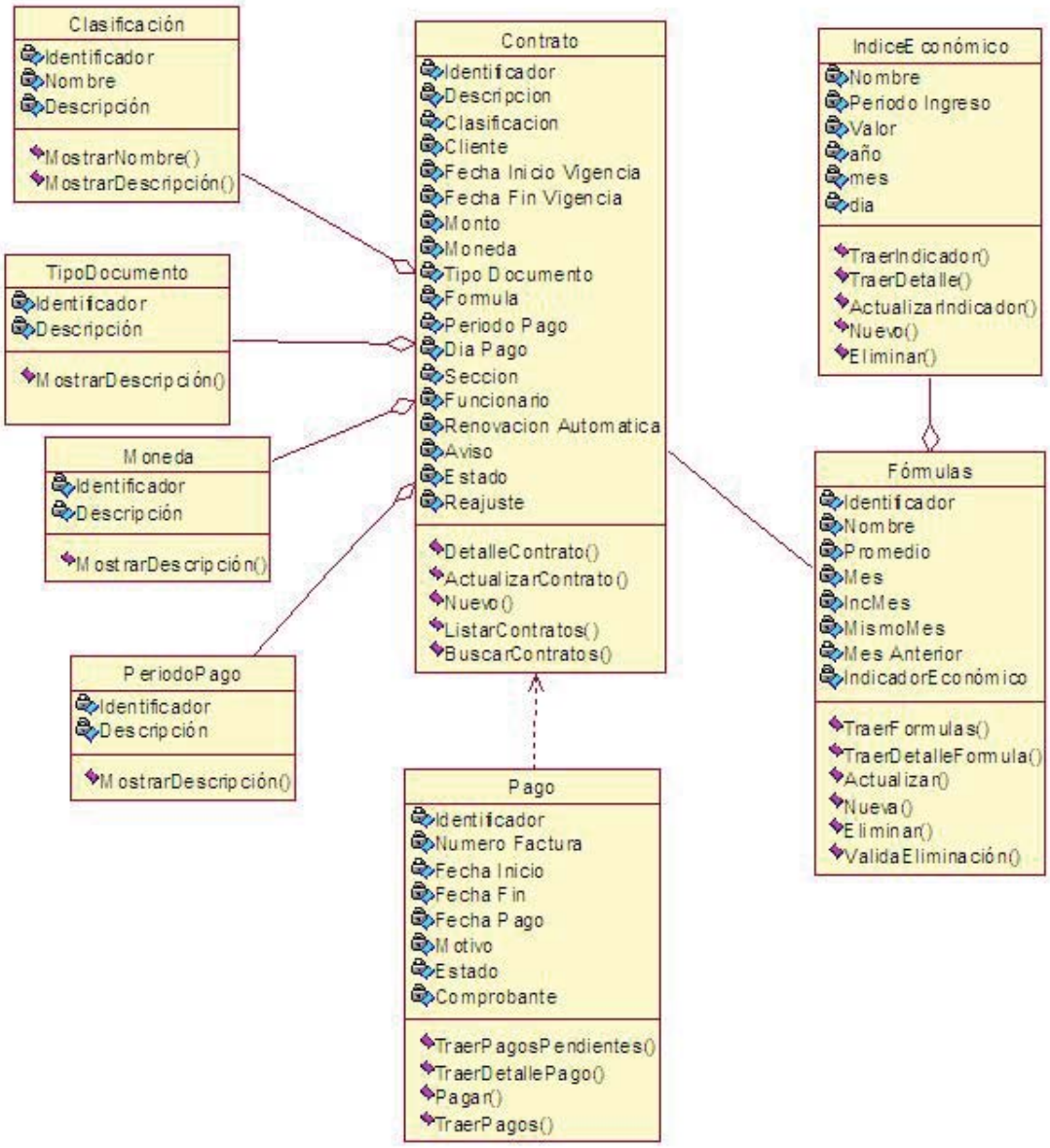


Ilustración 5.34 Diagrama de Clases

5.6.4 Diagrama de Componentes del Sistema

Se diseña la arquitectura del sistema basado en componentes físicos, los cuales interactúan para la obtención de los resultados esperados.

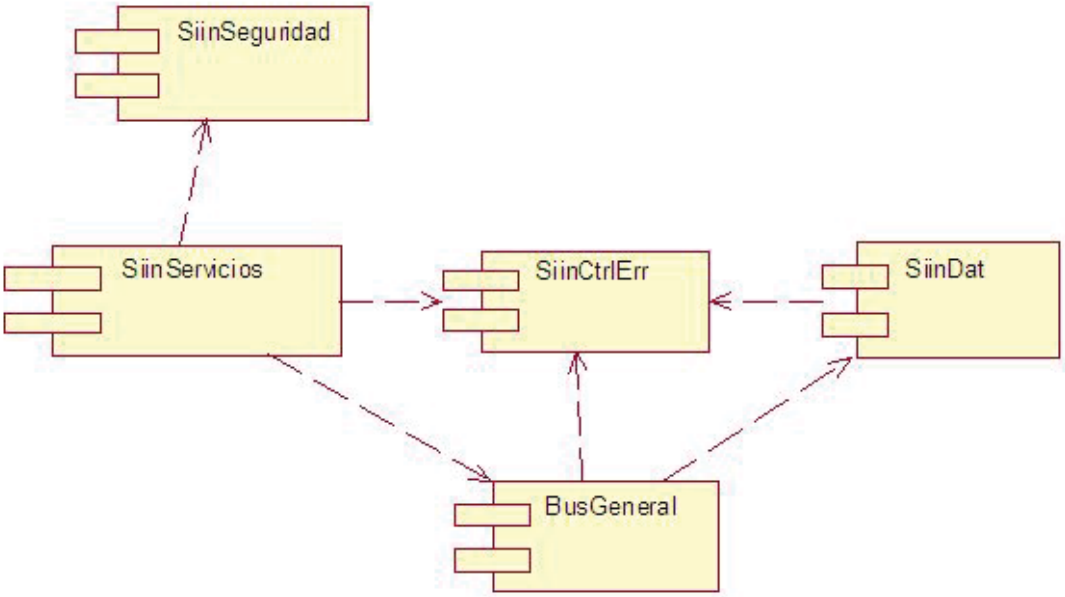


Ilustración 5.35 Diagrama de Componentes

Componente SIINSeguridad: es el encargado de proporcionar la seguridad al sitio, es transversal a toda a la aplicación, mediante la autentificación, el manejo de sesiones y el control de acciones no permitidas.

Componente SIINCtrlErr: componente encargado de controlar los errores y el manejo de excepciones en todos los niveles, evitando que el sistema se vuelva inestable y da continuidad al trabajo. Además informa al administrador de estos errores mediante un LOG que será almacenado en el servidor de aplicación.

5.6.5 Contratos

Corresponden a las funciones que se deben cumplir para efectuar el cambio que efectúan las operaciones del sistema, se incluyen cuando se produzca un cambio, en caso contrario no es necesario. Se definen para las operaciones del sistema definidas en los diagramas de secuencia

5.6.5.1 CU Pago Proveedores con Contrato

Contrato: Pagar

Referencias:

Precondiciones: Pago de proveedor pendiente

Poscondiciones: Se cambia estado a pagado. Se crea comprobante de pago.

5.6.5.2 CU Mantenedor Fórmulas

Contrato: Actualizar

Referencias:

Precondiciones: Fórmula Creada

Poscondiciones: Se modificó una instancia Fórmula.

5.6.5.3 CU Mantenedor Indicadores Económicos

Contrato: Actualizar

Referencias:

Precondiciones: Indicador creado

Poscondiciones: Se modificó una instancia Indicador

5.6.5.4 CU Mantenedor de Contratos

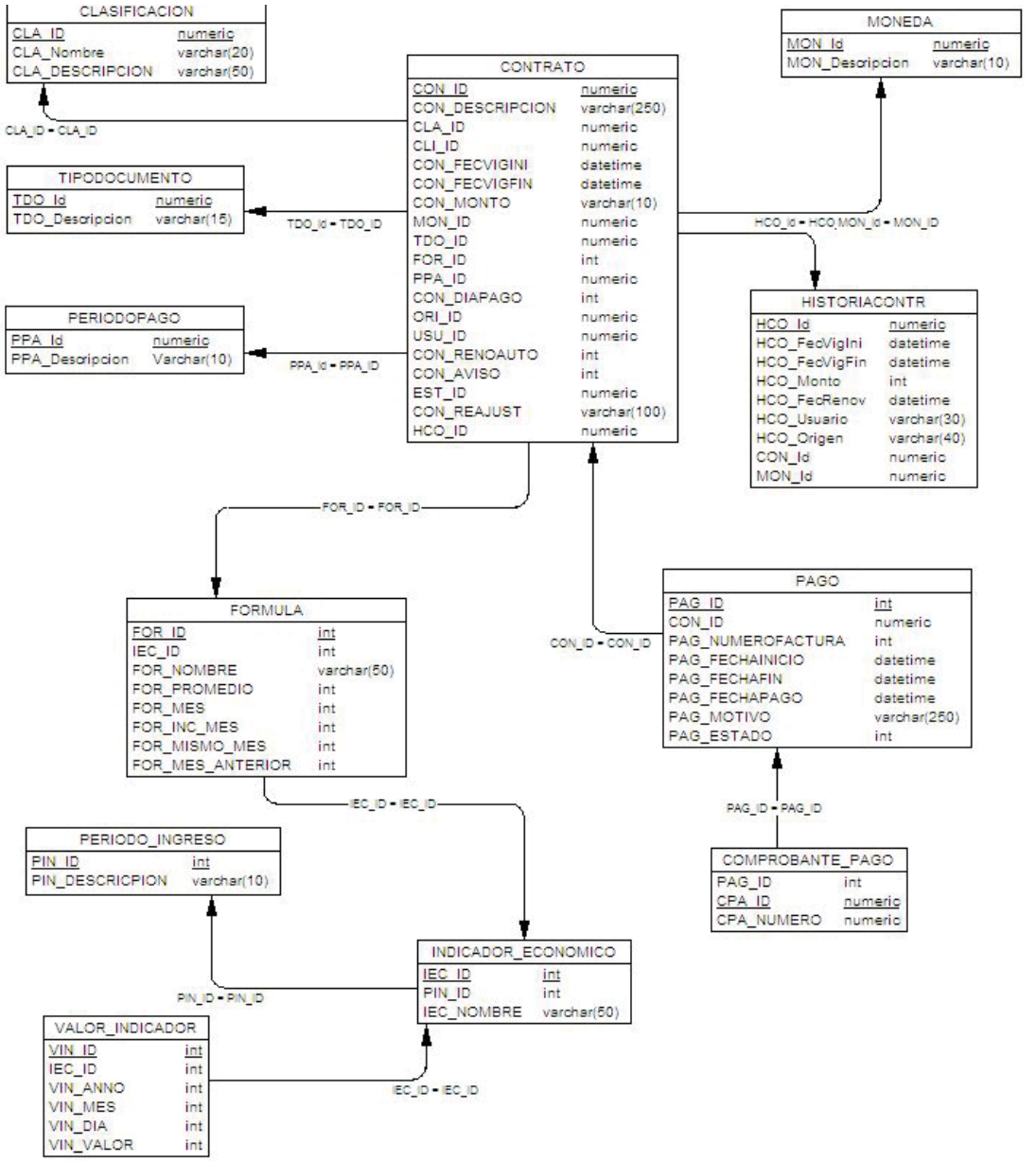
Contrato: ActualizarContrato

Referencias:

Precondiciones: Contrato Creado

Poscondiciones: Se modificó una instancia Contrato. Se creó una instancia de Historia Contrato.

5.7 DISEÑO DE DATOS



VISTA_PROVEEDORES

SISTCONTABLE.DBO.CLIENTE.COM_Id
SISTCONTABLE.DBO.CLIENTE.CLI_Id
SISTCONTABLE.DBO.CLIENTE.CLI_Rut
SISTCONTABLE.DBO.CLIENTE.CLI_ApePaterno
SISTCONTABLE.DBO.CLIENTE.CLI_ApeMaterno
SISTCONTABLE.DBO.CLIENTE.CLI_Nombres
SISTCONTABLE.DBO.CLIENTE.CLI_Direccion
SISTCONTABLE.DBO.CLIENTE.CLI_Telefono
SISTCONTABLE.DBO.CLIENTE.CLI_Celular
SISTCONTABLE.DBO.CLIENTE.CLI_Email
SISTCONTABLE.DBO.CLIENTE.CLI_Fax
SISTCONTABLE.DBO.CLIENTE.CLI_Sexo
SISTCONTABLE.DBO.CLIENTE.TPE_Id
SISTCONTABLE.DBO.CLIENTE.EST_Id
SISTCONTABLE.DBO.CLIENTE.CLI_CodigoWang
SISTCONTABLE.DBO.CLIENTE.CLI_DiaCierre
SISTCONTABLE.DBO.CLIENTE.CLI_PagoSueldo
SISTCONTABLE.DBO.CLIENTE.CLI_Contacto
SISTCONTABLE.DBO.CLIENTE.CLI_IdReparticion
SISTCONTABLE.DBO.CLIENTE.CLI_Registro

VISTA_FUNCIONARIO

PERFILES.DBO.USUARIO.USU_Id
PERFILES.DBO.USUARIO.USU_Rut
PERFILES.DBO.USUARIO.USU_Nombres
PERFILES.DBO.USUARIO.USU_ApePaterno
PERFILES.DBO.USUARIO.USU_ApeMaterno
PERFILES.DBO.USUARIO.USU_NombreUsu
PERFILES.DBO.USUARIO.USU_Email
PERFILES.DBO.USUARIO.USU_Clave
PERFILES.DBO.USUARIO.USU_Modifica
PERFILES.DBO.USUARIO.ORI_Id
PERFILES.DBO.USUARIO.EST_ID
PERFILES.DBO.USUARIO.TUS_Id

VISTA_SECCION

PERFILES.DBO.ORIGEN.ORI_Id
PERFILES.DBO.ORIGEN.ORI_Nombre
PERFILES.DBO.ORIGEN.ORI_Depende
PERFILES.DBO.ORIGEN.EST_ID

VISTA_ORIGEN_USUARIO

PERFILES.DBO.ORI_USU.ORI_Id
PERFILES.DBO.ORI_USU.USU_Id

Ilustración 5.36 Diagrama de Datos

5.7.1 Descripción de Tablas

A continuación se detallan las tablas del sistema, que comprenderán la base datos del mismo, cabe destacar que las vistas no son detalladas.

Nombre Tabla	**TipoDocumento**			
Descripción	Contiene los diferentes documentos que registrará el sistema.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
TDO_Id	SI	Entero		Identificador del Tipo de Documento dado por el sistema.
TDO_Descripcion	NO	Varchar	15	Nombre o texto descriptivo del Tipo de Documento.

Nombre Tabla	**PeriodoPago**			
Descripción	Registra los posibles periodos de pago que existirán, por ejemplo, Mensual, Trimestral, etc.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**

PPA_Id	SI	Entero		Identificador del Periodo de Pago dado por el sistema.
PPA_Descripcion	NO	Varchar	10	Nombre o texto descriptivo del Periodo de Pago.

Nombre Tabla	**Moneda**			
Descripción	Registra las diferentes monedas que se manejarán en el sistema.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
MON_Id	SI	Entero		Identificador de la Moneda dado por el sistema.
MON_Descripcion	NO	Varchar	10	Nombre o texto descriptivo de la Moneda.

Nombre Tabla	**Clasificación**			
Descripción	Registra las diferentes clasificaciones que pueden existir en los contratos.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
CLA_Id	SI	Entero		Identificador de la

				Clasificación dada por el sistema.
CLA_Nombre	NO	Varchar	20	Nombre de la clasificación
CLA_Descripcion	NO	Varchar	50	Texto descriptivo de la clasificación

Nombre Tabla	**Contrato**			
Descripción	Registra la información asociada a los Contratos de la Institución.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
CON_Id	SI	Entero		Identificador del Contrato dado por el sistema.
CON_Descripcion	NO	Varchar	80	Texto descriptivo del contrato, puede incluir Objeto y especificación del servicio
CON_FecVigIni	NO	Fecha		Fecha inicio de vigencia del contrato
CON_FecVigFin	NO	Fecha		Fecha de termino del contrato
CON_Monto	NO	Varchar	10	Monto total de contrato
MON_Id	NO	Entero		Identificador del tipo de moneda

TDO_Id	NO	Entero		Identificador del tipo de documento
FOR_Id	NO	Entero		Identificador de la fórmula
PPA_Id	NO	Entero		Identificador del período de pago
CON_DiaPago	NO	Entero		Día de pago del contrato
ORI_Id	NO	Entero		Sección del funcionario
USU_Id	NO	Entero		Identificador del Funcionario
CON_RenovAuto	NO	Entero		Si posee o no renovación automática
CON_Aviso	NO	Entero		Periodo en días de aviso previo de vencimiento de contrato
EST_Id	NO	Entero		Identificador del estado del contrato

Nombre Tabla	**Fórmula**			
Descripción	Registra las diferentes Fórmulas que se utilizarán en el calculo de los reajustes e impuestos de los pagos a proveedores			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
FOR_Id	SI	Entero		Identificador de la Fórmula

IEC_Id	NO	Entero		Identificador del indicador Económico
FOR_Nombre	NO	Varchar	50	Nombre de la Fórmula
FOR_Promedio	NO	Entero		Si es el promedio
FOR_Mes	NO	Entero		Promedio de cuantos meses
FOR_Inc_Mes	NO	Entero		Si se incluye el mes actual en el calculo
FOR_Mismo_Mes	NO	Entero		Si se incluye el mismo mes
FOR_Mes_Anterior	NO	Entero		Si es el del mes anterior

Nombre Tabla	**Pago**			
Descripción	Almacena los pagos realizados a los proveedores con contrato			
Campos				
Nombre	Clave Primaria	Tipo	Dimensión	Descripción
PAG_Id	SI	Entero		Identificador del pago
CON_Id	NO	Entero		Identificador del contrato
PAG_NumeroFactura	NO	Entero		Numero del documento asociado al pago
PAG_FechaInicio	NO	Fecha		Fecha inicio periodo de pago

PAG_FechaFin	NO	Fecha		Fecha fin de periodo de pago
PAG_FechaPago	NO	Fecha		Fecha de la realización del pago
PAG_Motivo	NO	Varchar	250	Motivo por el cual el pago sigue pendiente
PAG_Estado	NO	Entero		Estado del pago

Nombre Tabla	**Periodo_Ingreso**			
Descripción	Registra los posibles periodos de Ingreso que existirán, por ejemplo, Mensual, Diario, etc.			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
PIN_Id	SI	Entero		Identificador del periodo de ingreso
PIN_Descripcion	NO	Varchar	10	Nombre o texto descriptivo del periodo

Nombre Tabla	**Comprobante _ pago**
Descripción	Registra los comprobantes de los pagos efectuados a los proveedores

Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
PAG_Id	NO	Entero		Identificador del Pago
CPA_Id	SI	Entero		Identificador del Comprobante del pago
CPA_Numero	NO	Entero		Numero correlativo del comprobante

Nombre Tabla	**Indicador_Economico**			
Descripción	Registra los posibles indicadores económico utilizados por las Fórmulas			
Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
IEC_Id	SI	Entero		Identificador del Indicador
PIN_Id	NO	Entero		Identificador del periodo de ingreso
IEC_Nombre	NO	Varchar	50	Nombre del Indicador

Nombre Tabla	**Valor_Indicador**
Descripción	Almacena los valores de los indicadores económicos

Campos				
Nombre	**Clave Primaria**	**Tipo**	**Dimensión**	**Descripción**
VIN_Id	SI	Entero		Identificador del valor
IEC_Id	NO	Entero		Identificador del indicador
VIN_Anno	NO	Entero		Año del valor
VIN_Mes	NO	Entero		Mes del valor
VIN_Dia	NO	Entero		Día del valor
VIN_Valor	NO	Entero		Valor

5.7.2 Diseño de Interfaces

A continuación se detalla la plantilla a utilizar como modelo en el desarrollo de los formularios Web a desarrollar, las demás pantallas se ajustaran a los prototipos ya validados en los casos de uso de la etapa de análisis.

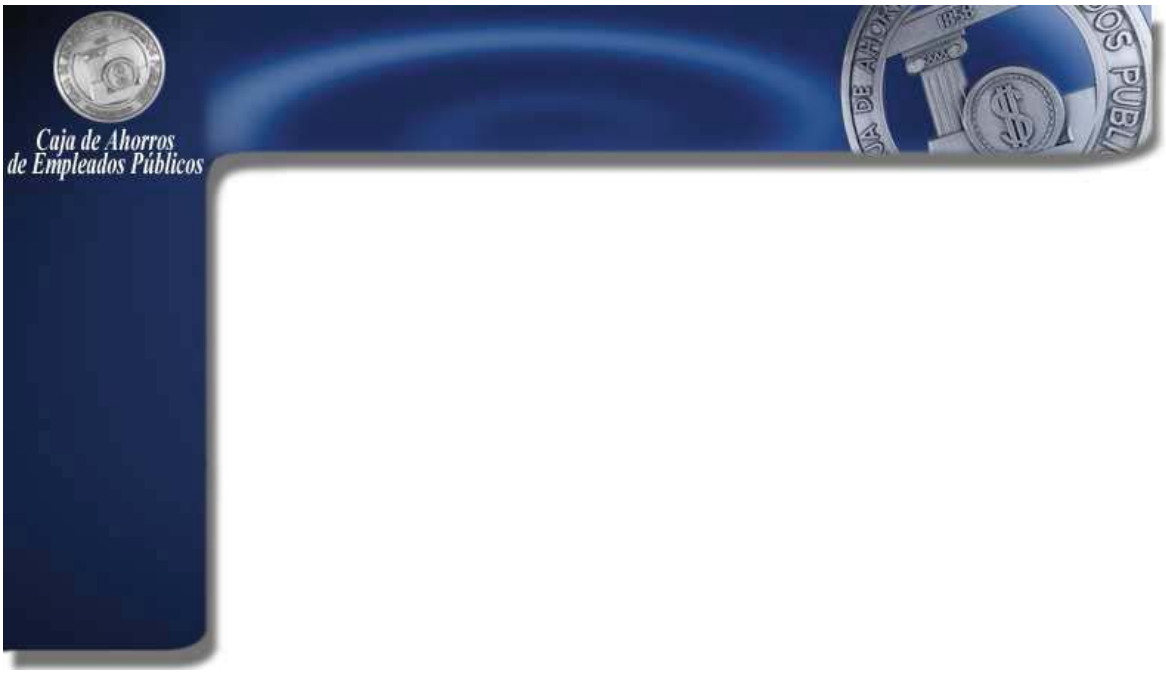


Ilustración 5.37 Frame superior y lateral

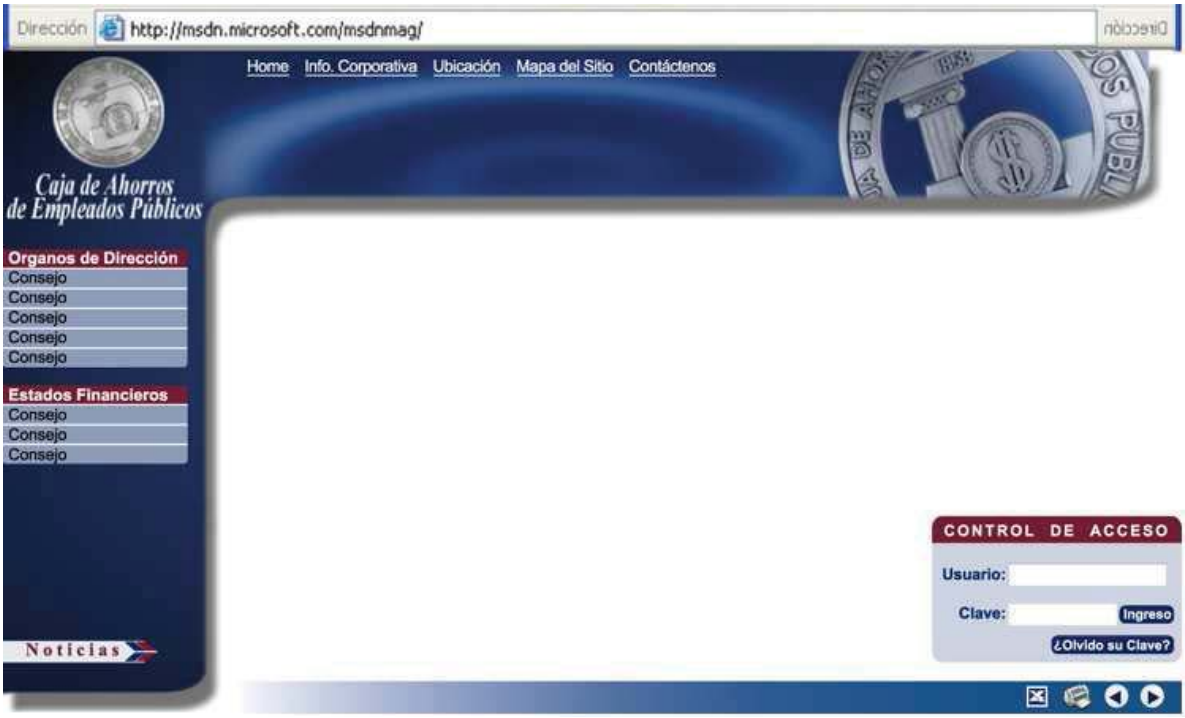


Ilustración 5.38 Pantalla Inicial

En el costado izquierdo se puede apreciar el formato del menú.

5.8 PRUEBAS

Una de las características típicas del desarrollo de software basado en el ciclo de vida es la realización de controles periódicos, normalmente coincidiendo con los hitos del proyecto o la terminación de documentos. Estos controles pretenden una evaluación de la calidad de los productos generados (especificación de requisitos, documentos de diseño, etc.) para poder detectar posibles defectos cuanto antes. Sin embargo, todo sistema o aplicación, independientemente de estas revisiones, debe ser probado mediante su ejecución controlada antes de ser entregado al cliente. Estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan habitualmente pruebas.

Las pruebas constituyen un método más para poder verificar y validar el software. Se puede definir la verificación como [9] “el proceso de evaluación de un sistema o de uno de sus componentes para determinar si los productos de una fase dada satisfacen las condiciones impuestas al comienzo de dicha fase”. Por ejemplo, verificar el código de un módulo significa comprobar si cumple lo marcado en la especificación de diseño donde se describe. Por otra parte, la validación es “el proceso de evaluación de un sistema o de uno de sus componentes durante o al final del proceso de desarrollo para determinar si satisface los requisitos especificados”. Así, validar una aplicación implica comprobar si satisface los requisitos marcados por el usuario. Podemos recurrir a la clásica explicación informal de Boehm de estos conceptos:

Verificación: ¿estamos construyendo correctamente el producto?

Validación: ¿estamos construyendo el producto correcto?

Como hemos dicho, las pruebas permiten verificar y validar el software cuando ya está en forma de código ejecutable.

5.8.1 Proceso de Pruebas

[8]En el libro “Ingeniería de Software” de Ian Sommerville podemos encontrar una figura que gráfica el proceso de pruebas:

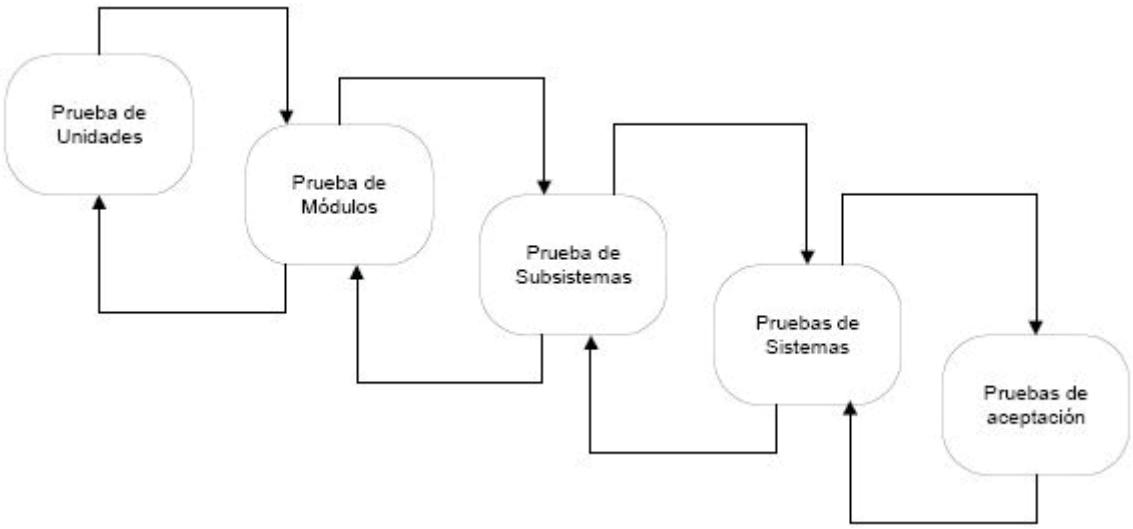


Ilustración 5.39 Proceso de Pruebas

En estas cinco etapas, en donde se prueban los componentes del sistema, la integración del sistema y por ultimo el sistema con los datos del usuario, se describirán los defectos y errores del sistema en construcción. Este es un proceso iterativo, ya que si un problema es encontrado en la última de las etapas este puede afectar perfectamente a cualquier componente y por ende el proceso de pruebas se deberá realizar nuevamente desde el principio, es si, guardando las proporciones del problema.

Pruebas de unidades, se prueban los componentes individuales asegurando su correcto comportamiento

Pruebas de módulos, un modulo comprende una colección de componentes dependientes, estos módulos son probados independientemente para ver si existen problemas en la comunicación.

Pruebas de subsistemas, en estas pruebas se verifican las interfaces entre módulos.

Pruebas de sistema, se prueban la integración de los subsistemas

Pruebas de aceptación, en esta etapa se verifica que el sistema cumpla con los requerimientos.

Una vez terminado el proceso de pruebas el sistema estará en condiciones de ser implantado y liberado para su explotación.

El diseño de casos de prueba está totalmente mediatizado por la imposibilidad de probar exhaustivamente el software. Por ejemplo, si se piensa en un programa muy sencillo que sólo suma dos números enteros de dos cifras (del 0 al 99). Si se quisiera probar, simplemente, todos los valores válidos que se pueden sumar, se debería probar 10.000 combinaciones distintas de cien posibles números del primer sumando y cien del segundo. Ahora aún quedarían por probar todas las posibilidades de error al introducir los datos (por ejemplo, teclear una letra en vez de un número). La conclusión es que, si para un programa tan elemental se deben realizar tantas pruebas, pensemos en lo que supondría un programa medio.

En consecuencia, las técnicas de diseño de casos de prueba tienen como objetivo conseguir una confianza aceptable en que se detectarán los defectos existentes (ya que la seguridad total sólo puede obtenerse de la prueba exhaustiva, que no es practicable) sin necesidad de consumir una cantidad excesiva de recursos (por ejemplo, tiempo para probar o tiempo de ejecución). Toda la disciplina de pruebas debe moverse, por lo tanto, en un equilibrio entre la disponibilidad de recursos y la confianza que aportan los casos para descubrir los defectos existentes. Este equilibrio no es sencillo, lo que convierte a las pruebas en una disciplina difícil que está lejos de parecerse a la imagen de actividad rutinaria que suele sugerir.

Ya que no se pueden probar todas las posibilidades de funcionamiento del software, la idea fundamental para el diseño de casos de prueba consiste en elegir algunas de ellas que, por sus características, se consideren representativas del resto. Así, se asume que, si no se detectan defectos en el software al ejecutar dichos casos, podemos tener un cierto nivel de confianza (que depende de la elección de los casos) en que el programa no tiene defectos. La dificultad de esta idea es saber elegir los casos que se deben ejecutar. De hecho, una elección puramente aleatoria no proporciona demasiada confianza en que se puedan detectar todos los defectos existentes. Por ejemplo, en el caso del programa de suma de dos números, elegir cuatro pares de sumandos al azar no aporta mucha seguridad a la prueba

(una probabilidad de 4/10.000 de cobertura de posibilidades). Por eso es necesario recurrir a ciertos criterios de elección. Veremos los 2 enfoques principales para el diseño de casos:

El enfoque **estructural** o de **caja blanca**, consiste en centrarse en la estructura interna (implementación) del programa para elegir los casos de prueba. En este caso, la prueba ideal (exhaustiva) del software consistiría en probar todos los posibles caminos de ejecución, a través de las instrucciones del código, que puedan trazarse.

El enfoque **funcional** o de **caja negra**, consiste en estudiar la especificación de las funciones, la entrada y la salida para derivar los casos. Aquí, la prueba ideal del software consistiría en probar todas las posibles entradas y salidas del programa.

Estos enfoques no son excluyentes entre sí, ya que se pueden combinar para conseguir una detección de defectos más eficaz.

5.8.2 Pruebas Estructurales. (Prueba de la caja blanca)

Como hemos dicho, las pruebas exhaustivas son impracticables. Podemos recurrir al clásico ejemplo de Myers, de un programa de 50 líneas con 25 sentencias IF en serie, en el que el número total de caminos contiene 33,5 millones de secuencias potenciales (contando dos posibles salidas para cada IF tenemos 225 posibles caminos). El diseño de casos tiene que basarse en la elección de caminos importantes que ofrezcan una seguridad aceptable en descubrir defecto, y para ello se utilizan los llamados criterios de cobertura lógica

Una posible clasificación de criterios de cobertura lógica es la que se ofrece abajo. Hay que destacar que los criterios de cobertura que se ofrecen están en orden de exigencia y, por lo tanto, de coste económico. Es decir, el criterio de cobertura de sentencias es el que ofrece una menor seguridad de detección de defectos, pero es el que cuesta menos en número de ejecuciones del programa.

Cobertura de sentencias. Se trata de generar los casos de prueba necesarios para que cada sentencia o instrucción del programa se ejecute al menos una vez.

Cobertura de decisiones. Consiste en escribir casos suficientes para que cada decisión tenga, por lo menos una vez, un resultado verdadero y, al menos una vez, uno falso. En general, una ejecución de pruebas que cumple la cobertura de decisiones cumple también la cobertura de sentencias.

Cobertura de condiciones. Se trata de diseñar tantos casos como sea necesario para que cada condición de cada decisión adopte el valor verdadero al menos una vez y el falso al menos una vez. No podemos asegurar que si se cumple la cobertura de condiciones se cumple necesariamente la de decisiones.

Criterio de decisión/condición. Consiste en exigir el criterio de cobertura de condiciones obligando a que se cumpla también el criterio de decisiones.

Criterio de condición múltiple. En el caso de que se considere que la evaluación de las condiciones de cada decisión no se realiza de forma simultánea (por ejemplo, según se ejecuta en el procesador), se podría considerar que cada decisión multicondicional se descompone en varias decisiones unicondicionales. Es decir, una decisión como **IF((a=1)AND(c=4)) THEN** se convierte en una concatenación de dos decisiones: **IF(a=1)** y **IF(c=4)**. En este caso, debemos conseguir que todas las combinaciones posibles de resultados (verdadero/falso) de cada condición en cada decisión se ejecuten al menos una vez.

Estas pruebas son realizas por el programador y en la práctica se efectúan durante la codificación y pruebas de unidades.

5.8.3 Prueba Funcional. (Prueba de la caja negra)

La prueba funcional o de caja negra se centra en el estudio de la especificación del software, del análisis de las funciones que debe realizar, de las entradas y de las salidas. Lamentablemente, la prueba exhaustiva de caja negra también es generalmente impracticable: pensemos en el ejemplo de la suma del programa antes mencionado, de nuevo, ya que no podemos ejecutar todas las posibilidades de funcionamiento y todas las combinaciones de entradas y de salidas, debemos buscar criterios que permitan elegir un

subconjunto de casos cuya ejecución aporte una cierta confianza en detectar los posibles defectos del software

Veremos a continuación distintas técnicas de diseño de casos de caja negra.

5.8.3.1 Particiones o Clases de Equivalencia

Cada caso debe cubrir el máximo número de entradas.

Debe tratarse el dominio de valores de entrada dividido en un número finito de clases de equivalencia que cumplan la siguiente propiedad: la prueba de un valor representativo de una clase permite suponer “razonablemente” que el resultado obtenido (existan defectos o no) será el mismo que el obtenido probando cualquier otro valor de la clase.

El método de diseño de casos consiste entonces en:

Identificación de clases de equivalencia.

Creación de los casos de prueba correspondientes.

Para identificar las posibles clases de equivalencia de un programa a partir de su especificación se deben seguir los siguientes pasos:

Identificación de las condiciones de las entradas del programa, es decir, restricciones de formato o contenido de los datos de entrada.

A partir de ellas, se identifican clases de equivalencia que pueden ser:

De datos válidos.

De datos no válidos o erróneos.

La identificación de las clases se realiza basándose en el principio de igualdad de tratamiento: todos los valores de la clase deben ser tratados de la misma manera por el programa.

Existen algunas reglas que ayudan a identificar clases:

Si se especifica un rango de valores para los datos de entrada (por ejemplo, “el número estará comprendido entre 1 y 49”), se creará una clase válida (1 ≤ número ≤ 49) y dos clases no válidas (número < 1 y número > 49).

Si se especifica un número de valores (por ejemplo, “se pueden registrar de uno a tres propietarios de un piso”), se creará una clase válida (1 ≤ propietarios ≤ 3) y dos no válidas (propietarios < 1 y propietarios > 3).

Si se especifica una situación del tipo “debe ser” o booleanas (por ejemplo, “el primer carácter debe ser una letra”), se identifican una clase válida (“es una letra”) y una no válida (“no es una letra”).

Si se especifica un conjunto de valores admitidos (por ejemplo, “pueden registrarse tres tipos de inmuebles: pisos, chalés y locales comerciales”) y se sabe que el programa trata de forma diferente cada uno de ellos, se identifica una clase válida por cada valor (en este caso son tres: piso, chalé y local) y una no válida (cualquier otro caso: por ejemplo, plaza de garaje).

En cualquier caso, si se sospecha que ciertos elementos de una clase no se tratan igual que el resto de la misma, deben dividirse en clases menores.

La aplicación de estas reglas para la derivación de clases de equivalencia permite desarrollar los casos de prueba para cada elemento de datos del dominio de entrada. La división en clases deberían realizarla personas independientes al proceso de desarrollo del programa, ya que, si lo hace la persona que preparó la especificación o diseñó el software, la existencia de algunas clases (en concreto, las no consideradas en el tratamiento) no será, probablemente, reconocida.

El último paso del método es el uso de las clases de equivalencia para identificar los casos de prueba correspondientes. Este proceso consta de las siguientes fases:

Asignación de un número único a cada clase de equivalencia.

Hasta que todas las clases de equivalencia hayan sido cubiertas por (incorporadas a) casos de prueba, se tratará de escribir un caso que cubra tantas clases válidas no incorporadas como sea posible.

Hasta que todas las clases de equivalencia no válidas hayan sido cubiertas por casos de prueba, escribir un caso para una única clase no válida sin cubrir.

La razón de cubrir con casos individuales las clases no válidas es que ciertos controles de entrada pueden enmascarar o invalidar otros controles similares. Por ejemplo, en un programa donde hay que “introducir cantidad (1 99) y letra inicial (A Z)” ante el caso “105 &” (dos errores), se puede indicar sólo el mensaje “105 fuera de rango de cantidad” y dejar sin examinar el resto de la entrada (el error de introducir “&” en vez de una letra).

5.8.3.2 Análisis de Valores Límite (AVL).

Mediante la experiencia (e incluso a través de demostraciones) se ha podido constatar que los casos de prueba que exploran las condiciones límite de un programa producen un mejor resultado para la detección de defectos, es decir, es más probable que los defectos del software se acumulen en estas condiciones. Podemos definir las condiciones límite como las situaciones que se hallan directamente arriba, abajo y en los márgenes de las clases de equivalencia.

El análisis de valores límite es un técnica de diseño de casos que complementa a la de particiones de equivalencia. Las diferencias entre ambas son las siguientes:

Más que elegir “cualquier” elemento como representativo de una clase de equivalencia, se requiere la selección de uno o más elementos tal que los márgenes se sometan a prueba.

Más que concentrarse únicamente en el dominio de entrada (condiciones de entrada), los casos de prueba se generan considerando también el espacio de salida.

El proceso de selección de casos es también heurístico, aunque existen ciertas reglas orientativas. Aunque parezca que el AVL es simple de usar (a la vista de las reglas), su

aplicación tiene múltiples matices que requieren un gran cuidado a la hora de diseñar las pruebas. Las reglas para identificar clases son las siguientes:

Si una condición de entrada especifica un rango de valores (“ l.0 valor +1.0”) se deben generar casos para los extremos del rango (1.0 y +1.0) y casos no válidos para situaciones justo más allá de los extremos (1.001 y + 1.001, en el caso en que se admitan 3 decimales).

Si la condición de entrada especifica un número de valores (“el fichero de entrada tendrá de 1 a 255 registros”), hay que escribir casos para los números máximo, mínimo, uno más del máximo y uno menos del mínimo de valores (0, 1, 255 y 256 registros).

Usar la regla 1 para la condición de salida (“el descuento máximo aplicable en compra al contado será el 50%, el mínimo será el 6%”). Se escribirán casos para intentar obtener descuentos de 5,99%, 6%, 50% y 50,01 %.

Usar la regla 2 para cada condición de salida (“el programa puede mostrar de 1 a 4 listados”). Se escriben casos para intentar generar 0, 1, 4 y 5 listados.

En esta regla, como en la 3, debe recordarse que:

Los valores límite de entrada no generan necesariamente los valores límite de salida (recuérdese la función seno, por ejemplo).

No siempre se pueden generar resultados fuera del rango de salida (pero es interesante considerarlo).

Si la entrada o la salida de un programa es un conjunto ordenado (por ejemplo, una tabla, un archivo secuencial, etc.), los casos se deben concentrar en el primero y en el último elemento.

Es recomendable utilizar el ingenio para considerar todos los aspectos y matices, a veces sutiles, en la aplicación del AVL.

Para este desarrollo utilizaremos estos dos enfoques en su completitud, lo que evitará posibles fallas del software. El plan de pruebas estará basado en el enfoque funcional, mientas que las pruebas de componentes serán tratadas mediante el enfoque estructural.

5.8.4 Plan de Pruebas

A continuación se detalla el plan de Pruebas realizado, mediante el cual fueron posibles encontrar fallas en el sistema. Se utilizó clases de Equivalencia para estos efectos; primero se definieron las clases y luego por modulo fueron probadas y registrado los resultados obtenidos, éstos son reflejados en tablas respectivas

5.8.4.1 Pruebas de Caja Negra por Módulo

5.8.4.1.1 Clases de Equivalencia

1.- Fechas de vigencia: deben ser Mayor el Fin que el inicio.

Valido Inicio 20/01/2006

Fin 20/02/2006

Inválido Inicio 26/02/2007

Fin 31/12/2006

2.- Día de Pago: entre 1 y 31.

Válido 25

Inválido 35

3.- Número de Monto Cancelado: debe ser numérico.

Valido 35000

Inválido 'hhh'

4.- Índice económico: debe ser numérico

Valido 18000

Inválido 'hh'

5.- Monto Contrato: deber ser numérico.

Valido 3000000

Inválido 'hh'

6.- Aviso Caducidad: debe ser numérico.

Valido 60

Inválido 'hh'

5.8.4.1.1.1 Casos de Uso Mantenedor de Contratos – Clase de Equivalencia 1

Buscador

Valor Ingresado	**Resultado Esperado**	**Resultado**
Fecha Caducidad Inicio 20/01/2006 Fecha Caducidad Fin 20/02/2006 Opción Buscar	Listado de contratos que cumplen con las fechas indicadas	Listado de contratos que cumplen con las fechas indicadas

Fecha Caducidad Inicio 26/2/2007 Fecha Caducidad Fin 31/12/2006 Opción Buscar	Error Fechas inconsistentes	Mensaje de Error en Fechas: Fecha de inicio y Fin Inconsistentes, favor ingréselas nuevamente

Nuevo Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Fecha Caducidad Inicio 20/01/2006 Fecha Caducidad Fin 20/02/2006 Opción Grabar	Ingreso de nuevo contrato	Ingreso de nuevo contrato
Fecha Caducidad Inicio 26/2/2207 Fecha Caducidad Fin 31/12/2006 Opción Grabar	Error Fechas inconsistentes	Mensaje de Error en Fechas: Fecha de inicio y Fin Inconsistentes, favor ingréselas nuevamente

Actualizar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**

Fecha Caducidad Inicio 20/01/2006 Fecha Caducidad Fin 20/02/2006 Opción Grabar	Actualización Exitosa	Actualización Exitosa
Fecha Caducidad Inicio 26/2/2207 Fecha Caducidad Fin 31/12/2006 Opción Grabar	Error Fechas inconsistentes	Mensaje de Error en Fechas: Fecha de inicio y Fin Inconsistentes, favor ingréselas nuevamente

Renovar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Fecha Caducidad Inicio 20/01/2006 Fecha Caducidad Fin 20/02/2006 Opción Grabar	Actualización Exitosa	Actualización Exitosa
Fecha Caducidad Inicio 26/2/2207 Fecha Caducidad Fin 31/12/2006	Error Fechas inconsistentes	Error, en capa de negocios al crear los pagos. Estado: Solucionado

Opción Grabar		

5.8.4.1.1.2 Casos de Uso Mantenedor de Contratos – Clase de Equivalencia 2

Nuevo Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Día de Pago 25 Opción Grabar	Ingreso de nuevo contrato	Ingreso de nuevo contrato
Día de Pago 35 Opción Grabar	Error Día de pago debe estar en 1 y 31	Error Día de pago debe estar en 1 y 31

Actualizar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Día de Pago 25 Opción Grabar	Ingreso de nuevo contrato	Ingreso de nuevo contrato
Día de Pago 35 Opción Grabar	Error Día de pago debe estar en 1 y 31	Error Día de pago debe estar en 1 y 31

5.8.4.1.1.3 Casos de Uso Pago Proveedores con Contrato – Clase de Equivalencia 3

Pago Proveedores

Valor Ingresado	Resultado Esperado	Resultado
Monto cancelado 35000 Opción Aceptar	Pago exitoso	Pago exitoso
Monto cancelado “hhh” Opción Aceptar	Error monto debe ser numérico	Error Capa de Negocio. Estado: Pendiente

5.8.4.1.1.4 Casos de Uso Mantenedor indicadores Económicos – Clase de Equivalencia 4

Actualizar

Valor Ingresado	Resultado Esperado	Resultado
Valor indicador 18000 Opción Aceptar	Almacenar valor	Valor Almacenado
Valor indicador “hhh” Opción Aceptar	Error valor debe ser numérico	No fue almacenado Estado: Pendiente

5.8.4.1.1.5 Casos de Uso Mantenedor de Contratos – Clase de Equivalencia 5

Nuevo Contrato

Valor Ingresado	Resultado Esperado	Resultado
Monto Contrato 3000000 Opción Grabar	Ingreso de nuevo contrato	Ingreso de nuevo contrato

Monto Contrato “hh” Opción Grabar	Error monto debe ser numérico	Error capa de negocio Estado: Pendiente

Actualizar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Monto Contrato 3000000 Opción Grabar	Actualización Exitosa	Actualización Exitosa
Monto Contrato “hh” Opción Grabar	Error monto debe ser numérico	Error capa de negocio Estado: Pendiente

Renovar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Monto Contrato 3000000 Opción Grabar	Actualización Exitosa	Actualización Exitosa
Monto Contrato “hh” Opción Grabar	Error monto debe ser numérico	Error capa de negocio Estado: Pendiente

5.8.4.1.1.6 Casos de Uso Mantenedor de Contratos – Clase de Equivalencia 6

Nuevo Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Aviso Caducidad 60 Opción Grabar	Ingreso de nuevo contrato	Ingreso de nuevo contrato
Aviso Caducidad “hh” Opción Grabar	Error días de caducidad debe ser numérico	Error capa de negocio Estado: Pendiente

Actualizar Contrato

Valor Ingresado	**Resultado Esperado**	**Resultado**
Aviso Caducidad 60 Opción Grabar	Actualización Exitosa	Actualización Exitosa
Aviso Caducidad “hh” Opción Grabar	Error días de caducidad debe ser numérico	Error capa de negocio Estado: Pendiente

Capítulo Seis

6 Conclusiones

6.1 CONCLUSIONES

Durante el desarrollo de un proyecto informático es muy importante no perder el horizonte, para esto es necesario contar con objetivos bien definidos y diferenciarlos de los requerimientos, ya que el límite entre ambos es bastante delgado.

En el desarrollo de este proyecto se presentaron diversas dificultades en la obtención de los requerimientos, ya que, en una institución cuya pirámide organizacional es bastante elevada y por ende las jerarquías están bien definidas, el personal de niveles mas bajos no puede tomar decisiones sin consultar a sus jefes, éstos a sus jefes y así replicándose sucesivamente hasta el último nivel superior. Esto fue solucionado realizando las consultas en el orden inverso, partiendo del nivel más alto, de esta forma cuando se llegaba a un usuario, ya se contaba con la aprobación de su superior para entregar la información correspondiente.

Además los usuarios se veían preocupados por la sustitución de sus labores por el software, por ende se mostraban reacios a entregar ciertos datos o en dar su aprobación a la reestructuración de los procesos organizacionales.

La falta de estándares provocaba contradicciones y dobles versiones para un mismo proceso.

Estos problemas mencionados fueron satisfactoriamente solucionados estableciendo mesas de diálogos en las cuales se discutían los casos de uso y se sancionaban, de esta forma, todos hablan el mismo lenguaje al final del proceso.

Con respecto al uso de las metodologías y paradigmas, es fundamental contar con una sola línea de documentos de referencia, ya que, según lo visto y estudiado, existen variadas y amplias diferencias entre autores y países con respecto a un mismo tema.

La metodología orientada a objetos proporciona herramientas de análisis y diseño que favorecen la comprensión natural de este proceso por parte de usuario no experimentados, ya que los casos de uso son auto explicativo y favorecen el rápido desarrollo de las aplicaciones.

También fue posible de apreciar que le proceso de planificación es uno de los pilares fundamentales del éxito de un proyecto, ya que acá es donde se establece: qué hacer, cómo

hacerlo, cuándo hacerlo y quién va ha hacerlo. Aquí es donde se realiza la proyección del trabajo y sus distintas complicaciones, se establecen las tareas y sus responsables; las tareas que se deben tener en cuenta, ya que si ellas se atrasan, el proyecto podrá sufrir una demora considerable.

Si bien es cierto es un proceso largo y tedioso, que requiere una conciencia amplia del marco del proyecto, se puede deferir, en todo el ámbito de la palabra, que es un proceso necesario y no se debe menospreciar al momento de realizar un futuro trabajo.

Por otro lado se puede concluir que el desarrollo de un sistema aporta a una organización una estructura renovada del proceso, facilitando y generando nuevas formas de visualizarlo por parte de los usuarios, los que no logran dejar de sorprenderse de la utilidad que el sistema les entrega. Lo que ellos en las reuniones solicitaron, ahora lo ven plasmado en funcionalidades bien definidas y de simple uso.

Los informáticos son entes generadores de cambio dentro de las organizaciones. Los sistemas informáticos son más que un conjunto de aplicaciones, son un grupo de procesos automáticos o no, que interactúan para cumplir con los objetivos de los negocios. La misión del ingeniero, es lograr la orquestación integral de estos procesos, así al tomar uno de ellos, modificarlo y volverlo a insertar dentro de la organización, es necesario minimizar los problemas de integración que este pueda provocar. Es aquí en donde la aplicación de la

técnica favorece a este proceso, entregando una visión amplia y segura de los pasos a seguir para completar con éxito las tareas.

REFERENCIAS

[1] PMI Project Management Institute, Guía de los Fundamentos de la Dirección de Proyectos, Tercera Edición, Project Management Institute, pp 103 – 156, EEUU, 2004.

[2] Sommerville Ian, Ingeniería de software, Pearson Educación, pp 44 – 53, México, 2002.

[3] Zavala Ruiz J, Ingeniería de Software Extracto de Tesis, publicado en http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html, 2000.

[4] Booch G., Software Architecture and the UML, publicación disponible en: http://www.rational.com/uml, 1998.

[5] Dombrovskaia Lioubov, Apuntes de Ingeniería de Software, Universidad Federico Santa María, Valparaíso, 2003

[6] The Institute of Risk Management(IRM), et al, Risk Management Standard, AIRMIC, ALARM, IRM, Londres, 2002.

[7] Rodríguez María Esther, Revisión del estado actual de los servicios .NET y tendencias futuras, Universidad de Vigo, Vigo, 2002.

[8] Pressman Roger, Ingeniería de software, un enfoque práctico, V Edición, McGraw-Hill/Interamericana de España, Madrid, 2002.

[9] IEEE, Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas, IEEE Std 1233, 1990

[10] Booch Grady, et al, El lenguaje Unificado de Modelado, Addison Wesley Iberoamericana, Madrid, 1999