

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



**SISTEMA DE REGISTRO Y SEGUIMIENTO A NIVEL
NACIONAL DE PERSONAS CON DISCAPACIDAD
VISUAL**

**INSTITUTO BOLIVIANO DE LA CEGUERA
JEFATURA DE PLANIFICACIÓN, DESARROLLO
EDUCATIVO Y REHABILITACIÓN**

**PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE
LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS**

AUTOR: Silvia Eugenia Mújica Torrez
TUTOR: Lic. Mario Claudio Loayza Molina Mg.Sc.
REVISOR: Mg.Sc. Miguel Cotaña Mier

**2008
LA PAZ, BOLIVIA**

Dedicatoria

A Dios por guiarme y darme fortaleza.

A mis padres Eduardo y Maria que con su cariño y apoyo me ayudaron a formar mi carácter.

A mis hermanos Píther y Willy que me motivaron a seguir adelante.

Agradecimientos

Toda mi gratitud al Lic. Feliciano Flores G. quien me brindo su apoyo y comprensión.

Al Lic. Mario Loayza Molina y al Lic. Miguel Cotaña Mier, por su paciencia, conocimientos, sabiduría y apoyo, los cuales hicieron posible la elaboración de este proyecto.

Al Lic. Jesús Lalo Flores Rojas por guiarme y ayudarme con sus conocimientos.

A toda mi familia, de quienes aprendí que los errores juegan un papel importante para vivir la vida.

A mis amigos en especial mi amiga Olivia Blanca, quienes me motivaron e impulsaron a llegar a mi meta.

Al personal docente y administrativo de la Carrera de Informática, quienes me colaboraron en todo.

RESUMEN

El objetivo del presente proyecto, tiene la finalidad de realizar el registro y seguimiento de personas con discapacidad visual en el Instituto Boliviano de la Ceguera, ayudando a un mejor manejo, organización y administración de la información para la toma de decisiones en proyectos orientados a estas personas. La información que es registrada son: Datos personales de la persona con discapacidad visual; en la que se contemplan características tales como: causa de ceguera, grado de ceguera y el tiempo de ceguera, también se considera la dinámica familiar. Datos académicos; que se subdividen en dos: Datos de Educación Regular y los datos de rehabilitación.

Para el desarrollo del sistema; las etapas de análisis y diseño se aplicó la Metodología Orientada a Objetos Rational Unified Process (RUP), que es una metodología estándar para la construcción completa del ciclo de ingeniería de software, tanto para sistemas tradicionales como para sistemas Web, se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos. Para la construcción de modelos se aplicó el Lenguaje Unificado de Modelado (UML), este lenguaje permite especificar, visualizar y construir los artefactos de los sistemas de software.

En cuanto a la implementación del Sistema se utilizó el lenguaje de programación Active Server Pages (ASP), con el servidor Internet Information Server (I.I.S.) y como Gestor de Base de Datos SQL – SERVER 2000, todas tecnologías de Microsoft®.

Para la calidad de software se aplicó el estándar ISO 9126, tomando en cuenta las características: Funcionalidad, Instalación / Mantenibilidad y Portabilidad.

ABSTRACT

The objective of the present project, has the purpose of carrying out the registration and people's pursuit with visual disability in the Bolivian Institute of the Blindness, helping to a better handling, organization and administration of the information for the taking of decisions in projects guided these people. The information that is registered is: The person's personal data with visual disability; in the one that you/they are contemplated characteristic such as: cause of blindness, degree of blindness and the time of blindness, are also considered the family dynamics. Academic data; that they are subdivided in two: Data of Regular Education and the rehabilitation data.

For the development of the system; the analysis stages and design you applies the Guided Methodology to Objects Rational Unified Process (RUP) that is a standard methodology for the complete construction of the cycle of software engineering, so much stops traditional systems as for systems Web, is centered in the production and maintenance of models of the system more than in producing documents. For the construction of models you applies the Unified Language of Modeling (UML), this language allows to specify, to visualize and to build the devices of the software systems.

As for the implementation of the System the programming language was used Active Server Pages (ASP), with the servant Internet Information Server (I.I.S.) and I eat database Agent SQL - SERVER 2000, all technologies of Microsoft®.

For the software quality you applies the standard ISO 9126, taking into account the characteristics: Functionality, Installation / Maintenance and Portability.

ÍNDICE

CONTENIDO

PÁGINA

CAPÍTULO I: INTRODUCCIÓN

1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES.....	2
1.2.1 Antecedentes de la Institución	2
1.2.2 Antecedentes del Proyecto.....	2
1.3 IDENTIFICACIÓN DEL PROBLEMA.....	3
1.3.1 Análisis de la problemática	3
1.3.2 Planteamiento del Problema	3
1.4 OBJETIVOS.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos	4
1.5 JUSTIFICACIÓN.....	4
1.5.1 Social	4
1.5.2 Económica	5
1.5.3 Tecnológica.....	5
1.6 ALCANCES.....	5
1.7 APORTES.....	5
1.8 MÉTODOS Y TÉCNICAS	6

CAPÍTULO II: MARCO TEÓRICO

2.1 INTRODUCCIÓN	7
2.2 METODOLOGÍA.....	7
2.3 ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS	7
2.3.1 Tipos de modelos	8
2.3.2 Modelo de datos vs. Modelo de objetos.....	8
2.3.3 Relación entre Modelo de Objetos y Modelo Dinámico.....	9
2.4 INGENIERÍA DE REQUISITOS.....	10
2.4.1 Inicio	10
2.4.2 Obtención.....	10
2.4.3 Elaboración	10
2.4.4 Negociación	11
2.4.5 Especificación	11
2.4.6 Validación	11
2.4.7 Gestión de Requisitos	11
2.5 METODOLOGÍA ORIENTADA A OBJETOS RATIONAL UNIFIED PROCESS (RUP).....	12
2.5.1 Desarrollo iterativo del software	12
2.5.2 Administración de requerimientos	12

2.5.3 Arquitecturas basadas en componentes.....	12
2.5.4 Modelamiento Visual	13
2.5.5 Verificación de Cualidades.....	13
2.5.6 Control de Cambios.....	13
2.5.7 Ciclos y fases del RUP	13
2.5.7.a) Inicio.....	14
2.5.7.b) Elaboración	15
2.5.7.c) Construcción	15
2.5.7.d) Transferencia	15
2.6 LENGUAJE UNIFICADO DE MODELADO (UML).....	16
2.6.1 Clases de Diagramas	16
2.6.2 Tipos de modelo	17
2.7 CICLO DE VIDA DE DESARROLLO DEL SOFTWARE	19
2.8 DESARROLLO ITERATIVO INCREMENTAL	19
2.9 BASE DE DATOS	20
2.9.1 Sistema de manejo de Base de Datos (DBMS).....	20
2.9.2 Base de datos relacionales	22
2.9.3 Fases del diseño de bases de datos.....	22
2.9.4 El modelo entidad – relación (E-R)	23
2.10 MODELO CLIENTE – SERVIDOR.....	23
2.11 ESTRUCTURA CLIENTE – SERVIDOR.....	24
2.12 APLICACIONES WEB	25
2.12.1 Arquitectura de las aplicaciones Web.....	26
2.12.2 Usabilidad en la Web.....	28
2.12.3 Heurísticas de Usabilidad para la Web.....	29
2.13 ACTIVE SERVER PAGES (ASP)	30
2.13.1 Internet Information Server (IIS).....	30
2.13.2 JavaScript	31
2.14 MICROSOFT® SQL – SERVER 2000	31
2.14.1 Características del SQL – SERVER 2000	32
2.15 CALIDAD DE SOFTWARE.....	32
2.15.1 Factores de calidad del estándar ISO 9126.....	33

CAPÍTULO III: DISEÑO DEL SISTEMA

3.1 FASE DE INICIO	34
3.1.1 Situación actual del sistema	34
3.1.2 Casos de uso y su descripción.....	35
3.1.2.a) Definición de los casos de uso de alto nivel.....	36
3.1.2.b) Diagrama de caso de uso	38

3.2 FASE DE ELABORACIÓN	39
3.2.1 Modelo de casos de uso.....	39
3.2.2 Casos de Uso Expandidos	40
3.2.3 Modelo dinámico.....	43
3.2.3.1 Diagrama de Estado.....	44
3.2.4 Modelo funcional.....	47
3.2.4.1 Diagramas de secuencia	47
3.2.5 Modelo de dominio	50
3.2.5.1 Modelo conceptual.....	50
3.2.5.2 Diccionario Modelo	52
3.3 FASE DE CONSTRUCCIÓN.....	53
3.3.1 Modelo de aplicación	53
3.3.1.1 Contrato de operaciones	53
3.3.2 Diseño del sistema	56
3.3.2.1 Casos reales de uso.....	56
3.3.2.2 Secuencias de pantallas	57
3.3.2.3 Diagrama de componentes	62
3.3.2.4 Diagrama de despliegue.....	62
3.4 FASE DE TRANSFERENCIA.....	63
 CAPÍTULO IV: CALIDAD Y COSTO-BENEFICIO DE SOFTWARE	
4.1 CALIDAD DE SOFTWARE	64
4.1.1 Funcionalidad.....	64
4.1.2 Instalación / Mantenibilidad	65
4.1.3 Portabilidad	67
4.2 ANÁLISIS COSTO – BENEFICIO DEL SISTEMA.....	67
4.2.1 Costo de análisis de programación	67
4.2.2 Costos de capacitación del usuario.....	68
4.2.3 Costo de instalación	68
 CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	
5.1 CONCLUSIONES	69
5.2 RECOMENDACIONES	70
BIBLIOGRAFÍA	71
ANEXOS.....	73
DOCUMENTOS.....	79

ÍNDICE DE FIGURAS

CAPÍTULO II: MARCO TEÓRICO

Figura 2.1: Vista general de RUP.....	14
Figura 2.2: Caso de Uso.....	18
Figura 2.3: Diagrama de Clase.....	18
Figura 2.4: Diagrama de Secuencia.....	19
Figura 2.5: Interfase entre base de datos física y peticiones del usuario.....	21
Figura 2.6: Modelo de aplicación cliente/servidor.....	24
Figura 2.7: Esquema General de las Tecnologías Web.....	26
Figura 2.8: Arquitectura - Web Tradicional.....	26
Figura 2.9: Arquitectura - Web Dinámico.....	27
Figura 2.10: Arquitectura Web de tres niveles.....	27

CAPÍTULO III: DISEÑO DEL SISTEMA

Figura 3.11: Funcionamiento actual de la institución.....	34
Figura 3.12: Sistema actual para el registro de la PCDV.....	35
Figura 3.13: Diagrama de caso de uso para el Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual (SIRESE).....	38
Figura 3.14: Diagrama de caso de uso para el proceso de Registro.....	39
Figura 3.15: Diagrama de caso de uso para el proceso de Seguimiento Académico.....	40
Figura 3.16: Diagrama de estado correspondiente a la verificación de usuario.....	44
Figura 3.17: Diagrama de estado correspondiente a la generación ó actualización De la Ficha Personal de una PCDV.....	45
Figura 3.18: Diagrama de estado correspondiente al proceso de Seguimiento Académico.....	46
Figura 3.19: Diagrama de estado correspondiente a la solicitud de Reportes.....	47
Figura 3.20: Diagrama de secuencia correspondiente al control de usuario.....	48
Figura 3.21: Diagrama de secuencia correspondiente a la generación y actualización de la ficha personal.....	48
Figura 3.22: Diagrama de secuencia correspondiente al Seguimiento Académico.....	49
Figura 3.23: Diagrama de secuencia correspondiente a la generación de reportes.....	49
Figura 3.24 Conceptos.....	50
Figura 3.25: Modelo Conceptual.....	51
Figura 3.26 Pantalla de identificación de usuario.....	57
Figura 3.27: Interfaz para el administrador del sistema.....	58
Figura 3.28: Listado del Historial del sistema.....	59
Figura 3.29: Formulario de datos personales de una PCDV.....	60
Figura 3.30: Listado de PCDV registrados por el sistema.....	61
Figura 3.31: Diagrama de componentes general del sistema.....	62
Figura 3.32: Diagrama de despliegue físico del sistema.....	62

ÍNDICE DE TABLAS

CAPÍTULO III: DISEÑO DEL SISTEMA

Tabla 3.1 Descripción de los casos de uso de solicitud de registro.....	36
Tabla 3.2: Descripción del caso de uso de generación y actualización de la ficha personal..	37
Tabla 3.3: Descripción del caso de uso de Seguimiento Académico.....	37
Tabla 3.4: Descripción del caso de uso de generación de reportes.....	37
Tabla 3.5: Caso de uso expandido de solicitud de registro.....	40
Tabla 3.6: Caso de uso expandido de generación y actualización de Ficha Personal.....	41
Tabla 3.7: Caso de uso expandido de Seguimiento Académico.....	42
Tabla 3.8: Caso de uso expandido de generación de reportes.....	43
Tabla 3.9 Diccionario modelo.....	52
Tabla 3.10 Contrato para la verificación de usuario.....	54
Tabla 3.11: Contrato correspondiente a la generación ó actualización de la ficha personal..	54
Tabla 3.12: Contrato correspondiente para el seguimiento académico.....	55
Tabla 3.13: Contrato correspondiente a la generación de reportes.....	56
Tabla 3.14: Relación de cambios y modificaciones del SIRESE versión 1.0.....	66
Tabla 3.15: Relación de valores empleados en el modelo COCOMO.....	68

1. INTRODUCCIÓN

1.1 Introducción

En la actualidad nuestra sociedad esta compuesta por diferentes tipos de organizaciones, cada una de ellas enfocada a un área ó un grupo social, pero todas ellas tienen como elemento básico una meta, objetivo ó propósito, para llegar al cumplimiento de este elemento básico estas organizaciones deben interactuar con su ambiente externo e incluso con otras organizaciones para poder retroalimentarse.

En Bolivia, a partir de la década de los 60, empiezan a surgir organizaciones de personas con discapacidad como la Fraternidad Cristiana de Enfermos y Discapacitados, Deporte Integrado; siendo las personas ciegas las que mayor antigüedad de organización presentaban, con sus propias organizaciones.

La promulgación de la Ley 1678 en Bolivia, referido a las Personas con Discapacidad y la emisión de su Reglamento 24807 de 4 de agosto de 1997; donde se estipula la atención, con Seguro de Corto Plazo, a toda persona con discapacidad y el Derecho a recibir un tratamiento en este campo con la misma eficacia y eficiencia otorgadas al común de las personas. Hasta el presente dichos preceptos no se cumplen ya que por un lado no se cuenta con un registro de todas las personas que presentan diferentes discapacidades ni los grados de discapacidad. Por otro lado, el gobierno Nacional no prioriza la atención a las personas con discapacidad. La marginación y la segregación que sufren las personas con discapacidad en la familia y en la sociedad hacen que ellas no puedan acceder a una integración real dentro de nuestra sociedad.

A pesar de contar con la Ley de la Persona con Discapacidad y su Decreto Reglamentario, mencionados anteriormente; las disposiciones en ellas emanadas son cumplidas en un mínimo porcentaje tanto por el Gobierno Central como por los Gobiernos Departamentales y Municipales. Siendo una de las principales causas la falta de datos actualizados de las personas con discapacidad visual, información que es muy necesaria al momento de realizar proyectos destinados a su beneficio.

En la actualidad existe la necesidad de procesar grandes volúmenes de información de forma automática, este factor afecta a entidades públicas como a privadas, éste es el caso del Instituto Boliviano de la Ceguera (I.B.C.), en el cual todos los procesos son realizados de forma manual.

En el presente proyecto de grado se realizara un estudio sobre las necesidades: de registro y el seguimiento de la información de las personas con discapacidad visual a nivel Nacional. Con el propósito de desarrollar un Sistema de Registro y Seguimiento a Nivel Nacional de personas con discapacidad visual, que colabore en el desarrollo de tareas y el cumplimiento de objetivos en el Instituto Boliviano de la Ceguera.

1.2 ANTECEDENTES

1.2.1 Antecedentes de la Institución

Actualmente el Instituto Boliviano de la Ceguera (I.B.C.) es una entidad pública descentralizada del Ministerio de Salud, con personalidad jurídica y posee autonomía de gestión, responsable de capacitar a personas ciegas para que puedan lograr una forma individual de vida, desarrollar sus potencialidades para lograr sus objetivos, basados en la igualdad de oportunidades con la diferenciación a través de la Rehabilitación e Integración Laboral en la sociedad. [DS, 26973], [LOPE, 2003] , [REF2]

Misión

El Instituto Boliviano de la Ceguera es la entidad pública responsable de asumir la preparación, readaptación e integración de la persona ciega, encaminada a que logren ser productivos y venzan formas individuales de vida para su inserción social y laboral, a través de la adopción de medidas de apoyo y disposiciones legales para personas ciegas. [REF1]

Visión

Lograr la igualdad de oportunidades de las personas ciegas con la diferenciación, en la que puedan ser competitivos y puedan satisfacer sus necesidades y expectativas. [REF1]

1.2.2 Antecedentes del Proyecto

Al realizar una investigación referencial, se encontraron los siguientes trabajos relacionados con el “Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual”:

- “Sistema de Registro y control de Afiliación” en el Seguro Social Universitario, de la Lic. Maria Victoria Vila Cruz. Implementa la metodología Métrica versión 3, con un análisis estructurado, tiene los procesos de registro de afiliados y control de aportes.[VILA, 2004]

- “Sistema de Seguimiento y Control de Rehabilitación para el régimen penitenciario”, de la Lic. Sonia Cañizaire Zarate. Aplica la metodología Orientada a Objetos R.U.P. (Rational Unified Process), con UML y la herramienta Rational Rose, usando un ciclo de vida iterativo incremental. [CAÑIZAIRE, 2005]

El presente trabajo se desarrollará con la metodología orientada a objetos R.U.P. y también un enfoque orientado a redes para facilitar la presentación de reportes, aplicando mecanismos de seguridad para el acceso al sistema.

1.3 IDENTIFICACIÓN DEL PROBLEMA

1.3.1 Análisis de la problemática

Mediante un proceso de entrevistas y una observación de actividades desarrolladas dentro de la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C., se identificaron los siguientes problemas:

- El registro de las personas con discapacidad visual se lo realiza de manera manual.
- La derivación de personas con discapacidad visual a una Escuela¹ es un proceso lento y burocrático que se la realiza manualmente.
- No se tiene un seguimiento académico de los alumnos de manera oportuna.
- La información de cambio de escuela por un alumno, no es controlada ni oportunamente registrada.
- No se cuenta con un control de Ingreso-Egreso de un alumno a una escuela.
- Los reportes en cuanto a la cuantificación y detalle de las PCDV de acuerdo a ciertas características no son confiables² ni oportunos por la falta de información actualizada.

1.3.2 Planteamiento del Problema

El I.B.C., realiza el registro de personas con discapacidad visual de forma manual, la cantidad de información conlleva a una demora considerable en el procesamiento y actualización de los datos, esto ocasiona que no se pueda hacer un seguimiento a las personas registradas y que los reportes no sean confiables ni oportunos.

¹ Una escuela es un centro ó instituto de rehabilitación en diferentes áreas funcionales e integrales.

² Que posean las cualidades recomendables para el fin a que se destinan.

¿Será posible controlar los procesos de Registro y Seguimiento de Personas con Discapacidad Visual a Nivel Nacional, de tal forma que la organización, planificación, administración y toma de decisiones sea la óptima para proyectos orientados a estas personas?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema automatizado de Registro y Seguimiento de personas con discapacidad visual a Nivel Nacional para el Instituto Boliviano de la Ceguera, que permita una mejor administración y toma de decisiones para proyectos orientados a estas personas.

1.4.2 Objetivos Específicos

- Desarrollar el subsistema de Registro, considerando las características y antecedentes de la ceguera de la persona con discapacidad visual.
- Desarrollar un subsistema de Seguimiento académico, para controlar ingresos y egresos a escuelas de rehabilitación por los alumnos.
- Desarrollar un subsistema de generación de reportes, de acuerdo a características que sean requeridas y necesarias para otros procesos, cuya interfaz sea amigable para el usuario.
- Desarrollar un subsistema de Historial, para el seguimiento cronológico del acceso y uso del sistema.

1.5 JUSTIFICACIÓN

El presente proyecto se justifica social, económica y tecnológicamente.

1.5.1 Social

Este proyecto se justifica socialmente porque aporta una herramienta de apoyo al desempeño de las tareas de los funcionarios de la institución. El implementar el nuevo sistema que esta ajustado a las necesidades de la institución permite optimizar las tareas realizadas por los funcionarios, además de prestar un servicio a fin de colaborar para futuras planificaciones de proyectos que beneficien directamente a la población de personas con discapacidad visual de Bolivia.

1.5.2 Económica

El nuevo Sistema de Registro y Seguimiento a Nivel Nacional permite reducir el tiempo de ejecución del proceso de Registro de datos de personas con discapacidad visual y facilita la toma de decisiones para la institución mediante información actualizada, oportuna y confiable, reduciendo de esta forma los costos para cumplir los fines de la institución.

Mediante la generación de reportes se reduce los costos empleados en: la recopilación, elaboración, verificación, presentación de la información y su posterior envío a los solicitantes.

1.5.3 Tecnológica

El presente proyecto se justifica tecnológicamente, con el desarrollo de un software que servirá de apoyo a la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C., el software será desarrollado en un entorno amigable para el usuario, lo cual facilitará la comprensión y aceptación del software.

1.6 ALCANCES

El presente proyecto tiene como alcances: el Registro y Seguimiento de la información de la persona con discapacidad visual a nivel departamental en su primera etapa, mediante el Análisis, Diseño e Implementación del sistema de información.

Se implementará en la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del Instituto Boliviano de la Ceguera (I.B.C.) y brindará reportes a niveles superiores según su jerarquía.

1.7 APORTES

- Con este proyecto se evita la duplicación ó extravió de la información.
- La generación de reportes en una interfaz amigable para el usuario, brinda a los funcionarios información actualizada y confiable.
- La implementación del sistema reduce el tiempo para efectuar operaciones de registro, seguimiento de la información, elaboración de datos estadísticos.
- El proyecto reduce los costos de operación en cada proceso.
- Se obtiene mayor control en las tareas de cada proceso.
- La generación de reportes aplicada en un entorno de red.

1.8 MÉTODOS Y TÉCNICAS

Para el desarrollo del presente proyecto se emplearon las siguientes metodologías y herramientas:

- Análisis de Requerimientos
 - ✓ Entrevistas y Cuestionarios
 - ✓ Observación de las actividades desarrolladas
 - ✓ Revisión de documentos (Leyes, Fichas de registro,...)

- Análisis y diseño del Sistema
 - ✓ Metodología Orientada a Objetos Rational Unified Process (RUP)
 - ✓ Lenguaje unificado para la construcción de modelos (UML)

- Implementación del sistema
 - ✓ Lenguaje de Programación: Active Server Page (ASP)
 - ✓ Administrador de Base de Datos: SQL Server 2000
 - ✓ Sistema Operativo: Plataforma Windows XP Profesional

2. MARCO TEÓRICO

2.1 Introducción

En este capítulo se expone un coordinado y coherente conjunto de conceptos, definiciones y proposiciones, que permitan abordar y comprender el problema y llegar a los objetivos propuestos.

2.2 Metodología

Una Metodología es una colección de métodos aplicados a lo largo del ciclo de vida del desarrollo del software y unificados por alguna aproximación general o filosófica.

Los modelos dan la oportunidad de fallar en condiciones controladas; todos los métodos incluyen:

- Notación. El lenguaje para expresar cada modelo.
- Proceso. Las actividades que encaminan a la construcción ordenada de los modelos del sistema.
- Herramientas. Los artefactos que eliminan el tedio de construir el modelo y reafirman las reglas sobre propios modelos, de forma que sea posible revelar errores e inconsistencias. **[PRESSMAN, 2002]**

2.3 Análisis y diseño orientado a objetos

La etapa de análisis consiste en la descripción del problema y las necesidades o requerimientos; en qué consiste el conflicto y qué debe hacerse. El análisis se centra en la investigación del problema.

En la etapa de diseño es necesario contar con descripciones detalladas de la solución lógica y cómo esta solución cumple con los requerimientos detectados en la etapa de análisis.

En el análisis y diseño orientado a objetos, se toma al dominio de un problema y su solución lógica dentro de la perspectiva de los objetos (cosas, entidades).

- En el análisis orientado a objetos se identifican y describen los objetos ó entidades que pertenezcan al dominio del problema.
- En el diseño orientado a objetos se definen y construyen los objetos lógicos del software que serán implementados en un lenguaje de programación orientado a objetos.

2.3.1 Tipos de modelos

Los modelos primordiales son:

- El modelo funcional describe los cálculos existentes del sistema sin describir ni como ni cuando se calculan.
- El Modelo Dinámico, modela los aspectos del sistema que tienen que ver con el tiempo y los cambios. También modela el Control, aspecto del sistema que describe la secuencia de operaciones que ocurren como respuesta a unos estímulos. Los conceptos principales del Modelo Dinámico:
 - Sucesos o Eventos: representan los estímulos externos.
 - Estados: representan valores de los objetos.
- El Modelo de Objetos describe la estructura estática de las clases, los objetos y sus relaciones.
 - Conceptos de clase, atributo, objeto y operación.
 - Enlaces y asociaciones.
 - Agregación.
 - Generalización y herencia.
 - Características avanzadas: clases abstractas, herencia múltiple, metadatos, claves candidatas y restricciones.

En resumen: modelo funcional especifica lo que sucede, el modelo dinámico especifica cuando sucede y el modelo de objetos especifica a que le sucede.

2.3.2 Modelo de datos vs. Modelo de objetos

Una BD se desarrolla mediante un Modelo de Datos, siguiendo los siguientes pasos:

- Se construye el Modelo de Datos sobre el dominio de la aplicación.
- Se transforma del Modelo de Datos en un Diseño de la BD mediante la aplicación
- de una serie de transformaciones estándar (normalización).

Un Sistema de Objetos se construye modelando mediante técnicas diferentes, pues las técnicas del Modelo de Datos son bastante limitadas para soportar el Modelo de Objetos. Se deben considerar las siguientes recomendaciones:

- No comenzar construyendo diagrama de clases; primero, es necesario comprender el problema.
- Intentar mantener el Modelo sencillo.
- Seleccionar con cuidado los nombres.
- No introducir punteros o referencias a otros objetos como atributos.
- Intentar evitar asociaciones n-arias.
- No intentar establecer el grado de multiplicidad perfecto al principio.
- No introducir atributos de enlace dentro de la clase.
- Utilizar asociaciones cualificadas donde sea posible.
- Intentar evitar generalizaciones profundamente anidadas.
- Intentar asociaciones uno a uno.
- No se sorprenda si su modelo requiere una revisión.
- Documentar siempre los Modelos de Objetos.

2.3.3 Relación entre Modelo de Objetos y Modelo Dinámico

El Diagrama de Estados describe toda o una parte del comportamiento de un objeto de una clase. Los Estados son los equivalentes a los valores de los atributos y enlaces de las clases. Los Sucesos se pueden representar como operaciones en el Modelo de Objetos.

La estructura del Modelo Dinámico está relacionada y restringida por la del Modelo de Objetos (generalización por restricción).

Un Estado Compuesto es el agregado de más de un subestado concurrente. En el Modelo de Objetos hay tres fuentes de concurrencia:

- Concurrencia entre componentes: cada componente tiene su propio Estado independiente.
- Concurrencia entre partes de un componente: agregación dentro de un objeto.
- Concurrencia del objeto: comportamiento concurrente de un objeto.

El Modelo Dinámico de una clase lo heredan sus subclases. La jerarquía de Sucesos es independiente de la jerarquía de clases (pueden definirse a través de diferentes clases de objetos, y son más expresivos que las operaciones). [RUMBAUGH, 1998]

2.4 Ingeniería de requisitos

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software. El diseño y la construcción de un elegante programa de computadora que resuelva el problema incorrecto no satisfacen las necesidades de nadie. Por lo tanto, es muy importante comprender lo que el cliente quiere antes de comenzar a diseñar y construir un sistema basado en computadora. El proceso de la ingeniería de requisitos se lleva a cabo a través de siete distintas funciones: inicio, obtención, elaboración, negociación, especificación, validación y gestión. [PRESSMAN, 2006]

2.4.1 Inicio

El proyecto inicia con conversaciones en las cuales se realizan preguntas al cliente, con el objetivo de tener una comprensión básica del problema, la naturaleza de la solución que desea y la efectividad de la comunicación preliminar entre el cliente y el desarrollador.

2.4.2 Obtención

En esta función se debe tener mucho cuidado de no caer en problemas de: ámbito, comprensión y volatilidad. Para ayudar a superar estos problemas, se debe realizar en forma organizada la actividad de recopilación de requisitos.

2.4.3 Elaboración

Esta actividad se enfoca en el desarrollo de un modelo técnico refinado de las funciones, características y restricciones del software. Se debe describir el problema de una forma que establezca una base para el diseño. El resultado final es un modelo de análisis que define el dominio de la información, las funciones y el comportamiento del problema. Se desarrollan los casos de uso que se definen desde el punto de vista de un actor³.

³ Un actor es un papel que las personas (usuarios) ó los dispositivos representan cuando interactúan con el software.

2.4.4 Negociación

Se pide a los clientes, usuarios y otros interesados que ordenen sus requisitos, para discutir los conflictos relacionados con la prioridad, se identifican y analizan los riesgos asociados con cada requisito. Mediante un enfoque iterativo, los requisitos se eliminan, combinan o modifican de forma que cada parte alcance cierto grado de satisfacción.

2.4.5 Especificación

La formalidad y el formato de una especificación varía con el tamaño y la complejidad del software que se va a construir, lo que se realiza generalmente son los escenarios de uso, cuando un sistema reside en un ambiente técnico que se comprenda bien. La especificación describe la función y el desempeño de un sistema basado en computadora y las restricciones que regirán su desarrollo.

2.4.6 Validación

Examina la especificación para asegurar que todos los requisitos de software se han establecido de manera precisa, esto mediante una revisión técnica formal en la cual son participes: el ingeniero de software, los clientes, usuarios y otros interesados. Los cuales examinan la especificación y buscan errores en el contenido o la interpretación, áreas que tal vez requieran un clarificación, información faltante, inconsistencias, conflictos entre los requisitos ó requisitos irreales.

2.4.7 Gestión de Requisitos

La gestión de requisitos es un conjunto de actividades que ayudan a identificar, controlar y rastrear los requisitos y los cambios a éstos en cualquier momento mientras se desarrolla el proyecto. Una vez identificados los requisitos se desarrollan las tablas de rastreabilidad.

La gestión de requisitos se inicia sólo para proyectos grandes, los cuales tienen cientos de requisitos identificables. En los proyectos pequeños esta función de la ingeniería de requisitos es bastante menos formal.

2.5 Metodología Orientada a Objetos Rational Unified Process (RUP)

RUP es una metodología estándar para la construcción completa del ciclo de ingeniería de software, tanto para sistemas tradicionales como para sistemas web. Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos, es una guía de cómo usar UML de la forma más efectiva.

RUP pretende implementar las mejores prácticas actuales en ingeniería de software: Desarrollo iterativo del software, Administración de requerimientos, Uso de arquitecturas basadas en componentes, Modelamiento visual del software, Verificación de la calidad del software, y el Control de cambios.

2.5.1 Desarrollo iterativo del software

Un proceso iterativo permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema. RUP sigue un modelo iterativo que aborda las tareas más riesgosas primero. Con esto se logra reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente.

2.5.2 Administración de requerimientos

RUP describe cómo: Obtener los requerimientos, organizarlos, documentar requerimientos de funcionalidad y restricciones, rastrear y documentar decisiones, captar y comunicar requerimientos del negocio.

Los casos de uso y los escenarios indicados por el proceso han probado ser una buena forma de captar requerimientos y guiar el diseño, la implementación y las pruebas.

2.5.3 Arquitecturas basadas en componentes

El modelo de desarrollo basado en componentes incorpora muchas de las características del modelo en espiral. Es evolutivo por naturaleza y exige un enfoque iterativo para la creación de software. Sin embargo, el modelo configura aplicaciones a partir de componentes de software empaquetados en forma previa. El modelo de desarrollo basado en componentes conduce a la reutilización del software, la cual proporciona beneficios a los ingenieros de software. **[PRESSMAN, 2006]**

RUP apoya el desarrollo basado en componentes, tanto nuevos como preexistentes. El proceso se basa en diseñar tempranamente una arquitectura base ejecutable, que sea:

- ✓ Flexible
- ✓ Fácil de modificar
- ✓ Intuitivamente comprensible
- ✓ Promueve la reutilización de componentes

2.5.4 Modelamiento Visual

El lenguaje unificado para la construcción de modelos UML es la base del modelamiento visual de RUP. Considera el modelamiento visual de la estructura y el comportamiento de la arquitectura y los componentes. Bloques de construcción: Ocultan detalles, permiten la comunicación en el equipo de desarrollo y permiten analizar la consistencia: entre los componentes y entre diseño e implementación.

2.5.5 Verificación de Cualidades

RUP ayuda a planificar, diseñar, implementar, ejecutar y evaluar pruebas que verifiquen: No sólo la funcionalidad, sino también el rendimiento y la confiabilidad. El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente.

2.5.6 Control de Cambios

Los cambios son inevitables, pero es necesario evaluar si éstos son necesarios y rastrear su impacto. RUP indica como controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.

2.5.7 Ciclos y fases del RUP

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción y Transición.

En cada fase se ejecutarán una o varias iteraciones, dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

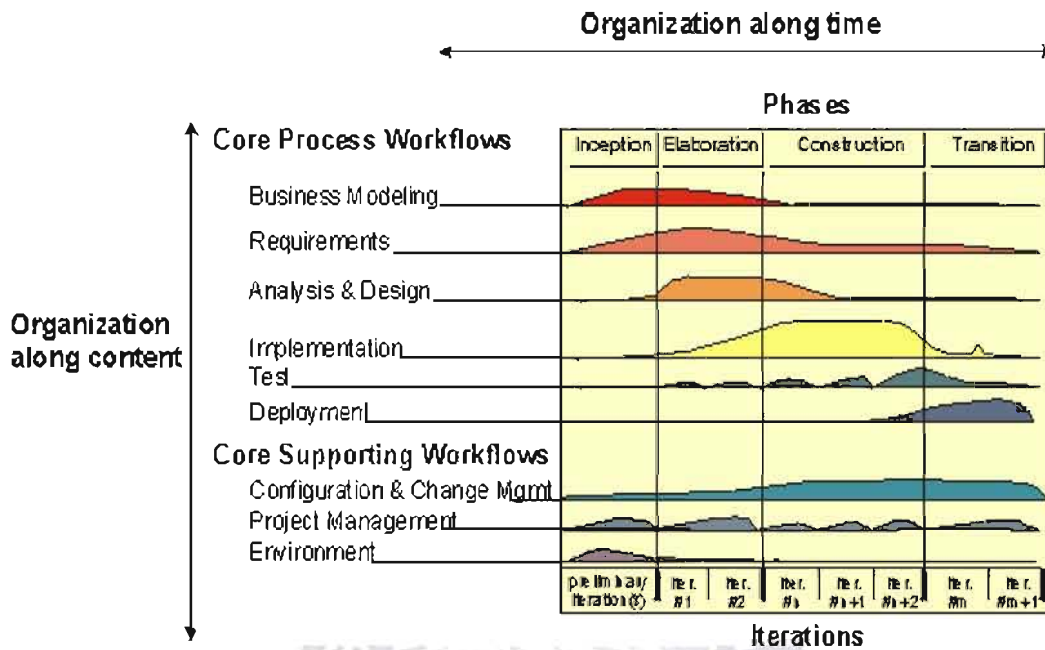


Figura 2.1: Vista general de RUP [REF3]

2.5.7.a) Inicio

En esta fase se establece la oportunidad y alcance el proyecto. Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción: Identificar todos los casos de uso y describir algunos en detalle.

Teniendo como objetivos particulares en esta fase:

- Definición del producto, aceptada por todas las partes interesadas involucradas en el proyecto.
- Discriminar los casos de uso (funcionalidades) prioritarios de los posibles pero no imprescindibles.
- Proponer una arquitectura inicial.
- Estimar los recursos necesarios para el desarrollo del proyecto y la distribución de roles y responsabilidades.
- Definir las herramientas a utilizar en cada proceso.

2.5.7.b) Elaboración

En esta fase se debe definir la arquitectura del sistema, tomando en cuenta los requerimientos obtenidos en la etapa de Inicio y proveer las alternativas para el control de riesgos. Teniendo como objetivos:

- Analizar el dominio del problema.
- Establecer una arquitectura base sólida.
- Desarrollar un plan de proyecto.
- Eliminar los elementos de mayor riesgo para el desarrollo exitoso del proyecto.

2.5.7. c) Construcción

En esta fase todos los componentes restantes se desarrollan e incorporan al producto. Todo es probado en profundidad. El énfasis está en la producción eficiente y no ya en la creación intelectual.

Los objetivos en esta etapa son:

- Obtener un sistema de calidad en un tiempo acotado.
- Completar para cada módulo las etapas de análisis, diseño, desarrollo y pruebas.
- Trabajar en paralelo en el desarrollo de subsistemas y módulos que pueden ser elaborados de forma independiente.
- Trabajar de forma iterativa e incremental en el desarrollo.
- El producto de software integrado como prototipo y corriendo en la plataforma adecuada.
- Realizar manuales de usuario.

2.5.7.d) Transferencia

El objetivo es traspasar el software desarrollado a la comunidad de usuarios. Una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).

Esta fase incluye:

- Pruebas Beta para validar el producto con las expectativas del cliente.
- Ejecución paralela con sistemas antiguos.
- Conversión de datos.
- Entrenamiento de usuarios.
- Distribuir el producto.

2.6 Lenguaje Unificado de Modelado (UML)

El UML es el lenguaje unificado para la construcción de modelos, este lenguaje permite especificar, visualizar y construir los artefactos de los sistemas de software. Es notacional (que, entre otras cosas incluye el significado de sus notaciones) y esta destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. El UML es un lenguaje para construir modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos, ni le indica cual proceso de desarrollo adoptar. UML es una notación estándar de diagramación y construcción de modelos. **[LARMAN, 1999]**

El lenguaje UML permite expresar el modelo de análisis utilizando una notación de modelado con reglas sintácticas, semánticas y pragmática. La sintaxis nos dice como mostrar y combinar símbolos, la sintaxis es comparable a las palabras del lenguaje natural. Las reglas semánticas nos dicen lo que significa cada símbolo y como interpretarlo. **[PRESSMAN, 2002]**

2.6.1 Clases de Diagramas

UML dispone de dos clases diferentes de diagramas: los que dan una vista estática del sistema y los que dan una visión dinámica. **[REF4]**

Los diagramas estáticos son:

- Diagrama de clases: muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- Diagrama de objetos: Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- Diagrama de componentes: Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- Diagrama de despliegue.: Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.
- Diagrama de casos de uso: Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Los diagramas dinámicos son:

- Diagrama de secuencia, Diagrama de colaboración: Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
- Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- Diagrama de actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Los diagramas a representar dependerán de las características del sistema a desarrollar, UML permite definir solo los necesarios, puede tomarse como recomendación lo siguiente:

- Aplicación monopuesto:
 - ✓ Diagrama de casos de uso.
 - ✓ Diagrama de clases.
 - ✓ Diagrama de interacción.
- Aplicación monopuesto, con entrada de eventos:
 - ✓ Añadir: Diagrama de estados.
- Aplicación cliente servidor:
 - ✓ Añadir: Diagrama de despliegue y diagrama de componentes, dependiendo de la complejidad.
- Aplicación compleja distribuida:
 - ✓ Todas las clases.

2.6.2 Tipos de modelo

UML cuenta con diferentes tipos de modelos cada uno de los cuales muestran diferentes aspectos de las entidades representadas, entre estos tenemos:

- Funcional: Muestra la funcionalidad del sistema desde el punto de vista del usuario, incluye:

- ✓ Diagramas de caso de uso. Por ejemplo: Este diagrama describe la funcionalidad (simplificada) de un sistema de restaurante, el comensal puede comer, tomar vino y pagar; sólo el cocinero puede preparar la comida.

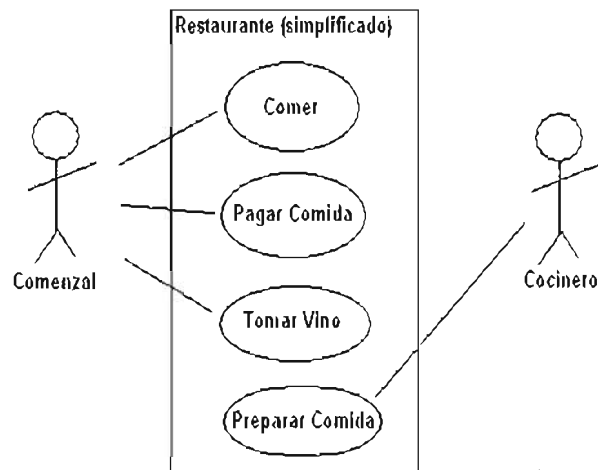


Figura 2.2: Caso de Uso [REF4]

- Objetos: Muestra la estructura y la subestructura del sistema usando objetos, atributos, operaciones y asociaciones, incluye:
 - ✓ Diagramas de clase. Por ejemplo: Este diagrama describe la estructura (simplificada) de un sistema de restaurante. El sistema tiene cualquier cantidad de platillos, una cocina, comedor y cualquier número de empleados, todos estos objetos asociados a un restaurante.

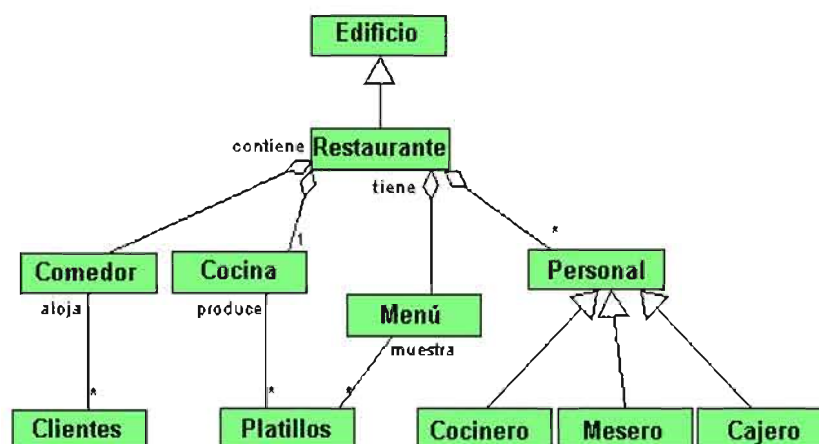


Figura 2.3: Diagrama de Clase [REF4]

- Dinámico: Muestra el comportamiento interno del sistema, incluye:
 - ✓ Diagramas de secuencia
 - ✓ Diagramas de actividad
 - ✓ Diagramas de estado

Por ejemplo: El siguiente diagrama describe la secuencia (simplificada) de mensajes de un sistema de restaurante. El diagrama representa a un cliente pidiendo comida y pagando.

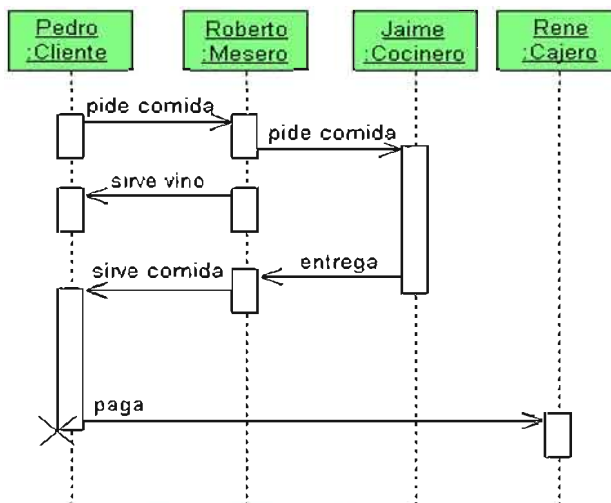


Figura 2.4: Diagrama de Secuencia [REF4]

2.7 Ciclo de vida de Desarrollo del Software

El proceso de desarrollo de software es un método para organizar las actividades relacionadas con la creación y mantenimiento de los sistemas de software.

El lenguaje UML estandariza los artefactos y la notación, pero no define un proceso oficial de desarrollo del software. [LARMAN, 1999]

2.8 Desarrollo iterativo incremental

Este ciclo de vida implica el crecimiento y perfeccionamiento gradual de un sistema mediante una secuencia de ciclos de desarrollo de análisis, diseño, implementación y pruebas. Para comenzar se realiza la planeación y especificación, luego el desarrollo pasa a la fase de construcción a través de una serie de ciclos de desarrollo, en cada ciclo se toma un conjunto pequeño de requerimientos, volviendo a desarrollar el análisis, diseño, implementación y pruebas, el sistema va creciendo con la finalización de cada ciclo de desarrollo.

Los ciclos de desarrollo se realizan hasta contemplar todos los requerimientos del sistema, estos ciclos iterativos de desarrollo se organizan a partir de los requerimientos del caso de uso y la duración de cada ciclo de desarrollo esta limitado en un marco temporal, que es un lapso rígidamente fijo.

El desarrollo iterativo incremental consta de las fases:

- **Fase de iniciación:** Es la primera fase del proceso de desarrollo, donde la idea inicial para el desarrollo debe encontrarse bien fundamentada para pasar a la fase de elaboración.
- **Fase de elaboración:** En esta fase se definen la visión del producto y su arquitectura, especificando los requisitos del sistema para crear una base arquitectónica sólida.
- **Fase de construcción:** Esta fase requiere varios ciclos de desarrollo mediante los cuales se va constituyendo el sistema. El objetivo final es obtener un sistema funcional de software que satisfaga y este conforme a los requerimientos de los usuarios.
- **Fase de transición:** Esta fase contempla la entrega del software a los usuarios. El proceso de desarrollo de software no termina con esta fase, porque durante esta fase el sistema puede necesitar mantenimiento para eliminar errores de programación.

2.9 Base de datos

Una base de datos es una colección integrada de datos almacenados, en distintos tipos de conjunto de datos relacionados pertenecientes a una entrada, de forma que sean accesibles para múltiples aplicaciones. [SENN, 1990]

2.9.1 Sistema de manejo de Base de Datos (DBMS)

El DBMS es considerado la parte medular de la base de datos que permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de reportes. [KENDALL, 1997]

Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

El DBMS es conocido también como Gestor de Base de datos.

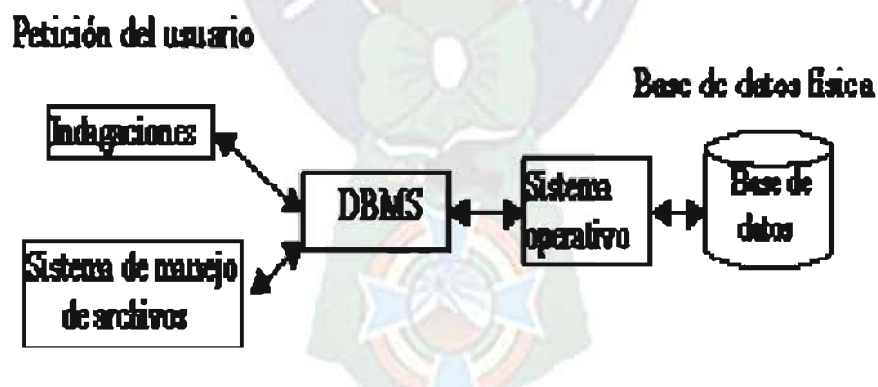


Figura 2.5: Interfase entre base de datos física y peticiones del usuario

El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

2.9.2 Base de datos relacionales

El objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información.

Para ello se deben diseñar los esquemas con una forma normal adecuada, para esto se requiere mayor información sobre la empresa del "mundo real" que se intenta modelar con la base de datos. La información adicional la proporciona una serie de limitantes que se denominan dependencias de los datos.

2.9.3 Fases del diseño de bases de datos

Para el diseño de Bases de Datos se consideran las siguientes fases:

- **Recolección y análisis de requerimientos:** Se realiza mediante entrevistas a los usuarios de la base de datos para recoger y documentar sus necesidades de información. Paralelamente se definen los requerimientos funcionales que consisten en operaciones (transacciones) que se aplicarán a la base de datos.
- **Diseño conceptual:** Cuando se tienen requerimientos, el siguiente paso es crear un esquema conceptual para la base de datos mediante un modelo de datos conceptual de alto nivel. El esquema conceptual contiene una descripción detallada de los requerimientos de información de los usuarios, y contiene descripciones de los tipos de datos, relaciones entre ellos y restricciones. El **modelo E-R** (entidad - relación), que describe los datos como entidades, vínculos (relaciones) y atributos, puede utilizarse para el diseño de esquemas conceptuales.
- **Diseño lógico de la base de datos:** Consiste en implementar la base de datos con un S.G.B.D. comercial, transformando el modelo conceptual al modelo de datos empleados por el S.G.B.D. (jerárquico, red o relacional).
- **Diseño físico de la base de datos:** En este caso se especifican las estructuras de almacenamiento internas y la organización de los archivos de la base de datos.

2.9.4 El modelo entidad – relación (E-R)

El modelo entidad relación (E-R) se basa en una percepción de un mundo real que consiste en una colección de objetos básicos llamados entidades, y relaciones entre estos objetos.

- Una Entidad: es un objeto que es distinguible de otros objetos por medio de un conjunto específico de atributos.
- EL Atributo: son las características asociadas al objeto.
- Una Relación: es una asociación entre varias entidades.
- Conjunto de entidades: entidades del mismo tipo.
- Conjunto de relaciones: relaciones del mismo tipo.

El modelo E-R representa ciertas restricciones a las que deben ajustarse los contenidos de una base de datos. Una restricción importante es la de cardinalidad de asignación, que expresa el número de entidades a las que puede asociarse otra entidad mediante un conjunto de relación.

La estructura lógica global de una Base de Datos puede expresarse gráficamente por el diagrama E-R:

- **Rectángulos:** representan conjunto de entidades.
- **Elipses:** representan atributos.
- **Rombos:** representan relaciones entre conjuntos de entidades.
- **Líneas:** que conectan atributos a conjuntos de entidades conjuntos de entidades a relaciones.

2.10 Modelo Cliente – Servidor

El modelo cliente – servidor tiene diversas definiciones, entre las cuales las más principales son:

Desde un punto de vista conceptual: «Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información».

Según términos de arquitectura se define como: «Los distintos aspectos que caracterizan a una aplicación (proceso, almacenamiento, control y operaciones de entrada y salida de datos) en el sentido más amplio, están situados en más de un computador, los cuales se encuentran interconectados mediante una red de comunicaciones».

IBM define al modelo Cliente/Servidor: «Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores». [REF5]

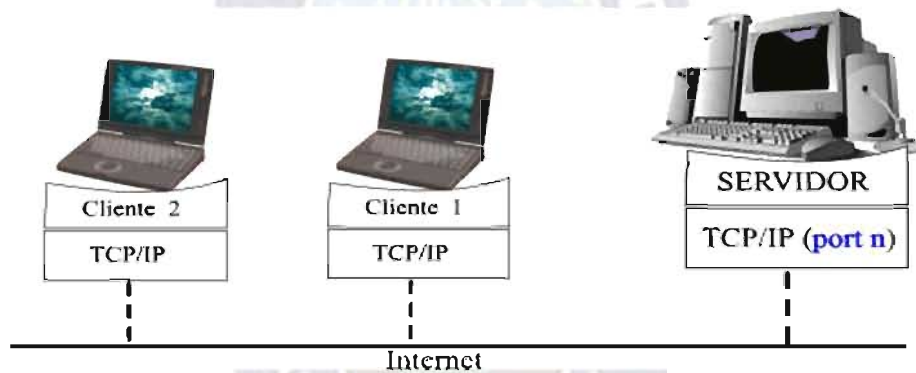


Figura 2.6: Modelo de aplicación cliente/servidor [REF6]

2.11 Estructura Cliente – Servidor

Dependiendo de que recursos ofrezca el servidor y cuales se mantienen en los clientes, se pueden hacer distinciones entre distintas estructuras cliente-servidor.

En estas estructuras se diferencia:

- Donde se encuentran los datos.
- Donde se encuentran los programas de aplicación.
- Donde se presentan los datos.

Tenemos los siguientes sistemas:

- a) **Sistema centralizado basado en el host (anfitrión).** Aquí, los datos, los programas de aplicación y la presentación se encuentran en el servidor. La imagen final se transmite a los terminales de los usuarios. Desde los terminales, las cadenas de caracteres de las entradas de los usuarios se reenvían al host. Este concepto es el que sirve de base para los mainframe.
- b) **Pc cliente y servidor host.** Los datos de aplicación se conservan de forma centralizada en el servidor. Con programas clientes de las aplicaciones, éstas se presentan en cada estación de trabajo. El lugar de trabajo suele ser una pc ejecutando, por ejemplo windows.
- c) **Estación de trabajo cliente y servidor de archivo.** Los datos se encuentran en el servidor (generalmente en una base de datos). Con una base de datos cliente se accede a esos datos desde cualquier computadora. En el cliente se procesan los datos utilizando la inteligencia del cliente. Cada computadora contiene aplicaciones con las que se puede procesar los datos.
- d) **Pc cliente y servidor de aplicaciones.** En esta red se dispone al menos de dos servidores distintos. Uno de ellos actúa únicamente como servidor de base de datos y el resto como servidor de aplicaciones. Los servidores de aplicaciones de esta red también son los responsables de acceso a las bases de datos. En las estaciones de trabajo funcionan los clientes de los programas de aplicación correspondientes.
- e) **Sistema cliente-servidor cooperativo descentralizado.** Las bases de datos están repartidas en distintos servidores o incluso clientes. Las aplicaciones funcionan igualmente en distintos servidores o en parte también en clientes. [REF7]

2.12 Aplicaciones Web

Con la evolución y expansión de Internet y del Web, se han abierto una infinidad de posibilidades para un usuario en cuanto al acceso a la información desde casi cualquier sitio. Además con el avance de la tecnología y los requerimientos de los usuarios, cada vez más crecientes, aumenta la demanda de aplicaciones cada vez más rápidas.

Gracias al desarrollo de nuevas herramientas y tecnologías, las Aplicaciones Web son cada vez más populares. La facilidad de su desarrollo provoca, a veces, la ausencia de un análisis y diseño correctos, pero están consiguiendo reemplazar a las aplicaciones software tradicionales.

El CGI añadió interactividad a las páginas Web pero sus deficiencias en el desarrollo de aplicaciones y en la escalabilidad de las mismas, ha conducido al desarrollo de APIs específicos de servidor como *Active Server Pages*, ASP, y PHP, que son más eficientes que su predecesor CGI.

El esquema general de la situación se puede ver en la Figura 2.7, donde se muestran cada tipo de tecnología involucrada en la generación e interacción de documentos Web.

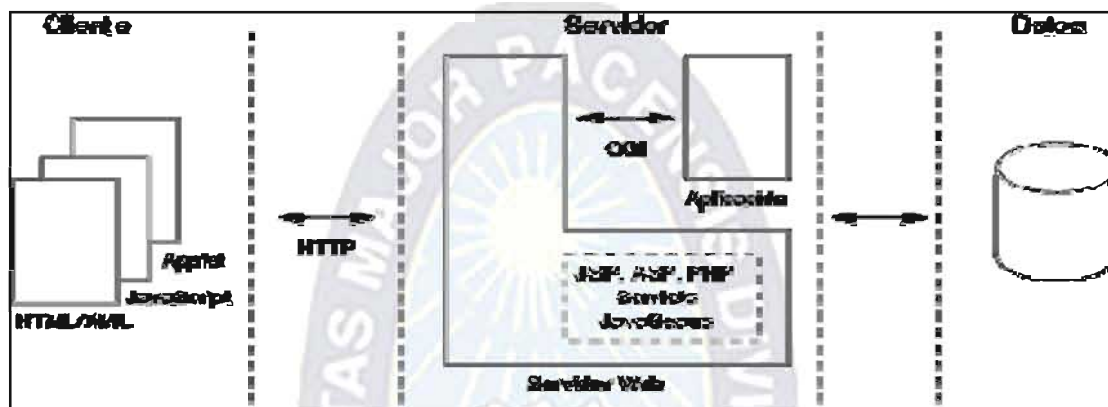


Figura 2.7: Esquema General de las Tecnologías Web [REF8]

Una aplicación Web no es más que una especialización de un proceso cliente/servidor, con lo que se puede aprovechar el modelado de dichas aplicaciones. En particular, los casos de uso son una herramienta fundamental en la captura de requisitos.

2.12.1 Arquitectura de las aplicaciones Web

Una Aplicación Web se sirve de un sitio o página como entrada a una verdadera aplicación. Tenemos las siguientes arquitecturas:

- ✓ Arquitectura de un sitio Web Tradicional:

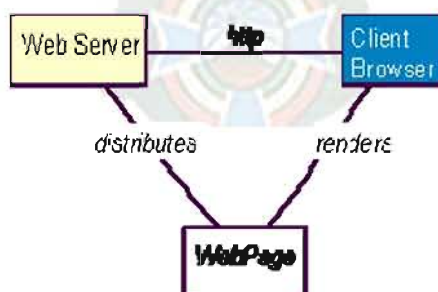


Figura 2.8: Arquitectura - Web Tradicional [REF8]

✓ Arquitectura de un sitio Web Dinámico

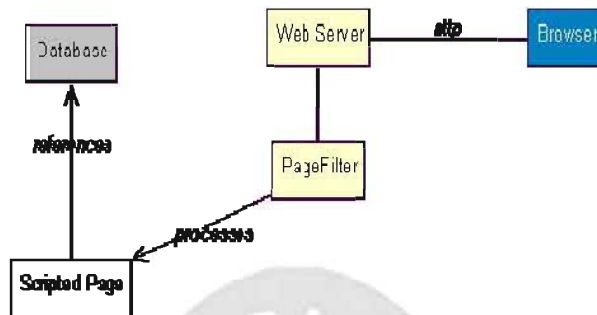


Figura 2.9: Arquitectura - Web Dinámico [REF8]

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles (véase la Figura 2.10):

- El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado.
- El segundo nivel está referido habitualmente a algún tipo de programa o *script*.
- Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

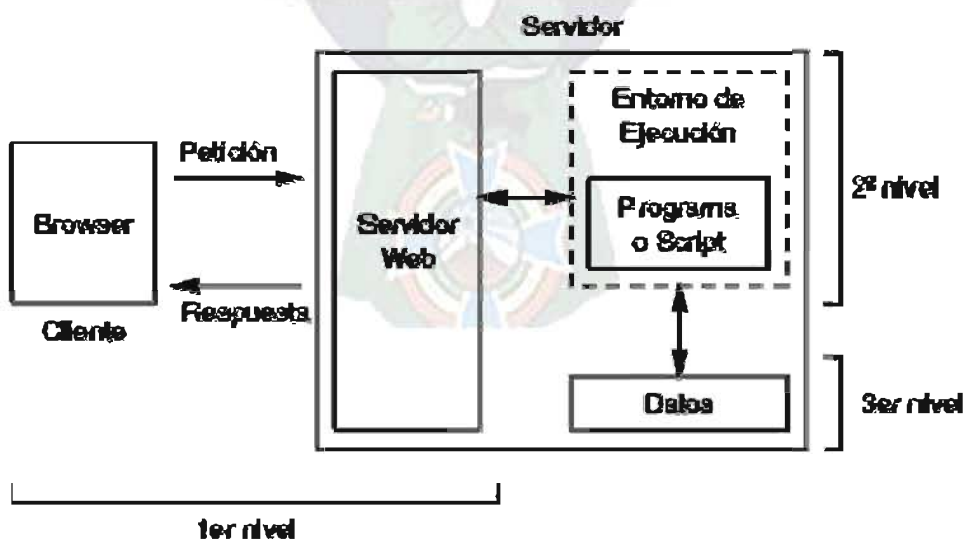


Figura 2.10: Arquitectura Web de tres niveles [REF8]

La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como esta distribuido este código.

2.12.2 Usabilidad en la Web

La definición de usabilidad conforme a la norma ISO 9241, parte 11 dice: La usabilidad es el rango en el cual un producto puede ser usado por unos usuarios específicos para alcanzar ciertas metas especificadas con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

La usabilidad no se limita a sistemas computacionales exclusivamente, sino que es un concepto aplicable a cualquier elemento en el cual se va a producir una interacción entre un humano y un dispositivo.

En la actualidad la usabilidad de los sistemas computacionales tiene los siguientes atributos definidos:

Facilidad de aprendizaje: ¿Cuánto le toma al usuario típico de una comunidad aprender la manera en como se usan los comandos relevantes a un conjunto de tareas?

Velocidad de desempeño: ¿Cuánto le toma a un usuario completar un grupo de tareas específicas (benchmark tasks)?

Tasas de error por parte de los usuarios: ¿Cuántos y qué errores hace la gente al ejecutar un grupo de tareas específicas?

Retención sobre el tiempo: ¿Qué tan bien recuerdan los usuarios la manera en como funciona el sistema después de una hora, un día o una semana?

Satisfacción subjetiva: ¿Qué tanto le gustaron a los usuarios los distintos atributos del sistema? Este atributo se refiere a la impresión subjetiva del usuario respecto al sistema.

John Cato en su publicación User-centered web design el 2001 sugiere además los siguientes atributos:

Control: Los usuarios deben de sentir que tienen el control por sobre la aplicación, y no al revés.

Habilidades: Los usuarios deben de sentir que el sistema apoya, complementa y realza sus habilidades y experiencia - el sistema tiene respeto por el usuario.

Privacidad: El sistema ayuda a los usuarios a proteger su información o la de sus clientes.

Cada uno de los atributos antes mencionados tiene una ponderación acorde a la actividad que se quiera realizar con un sistema. Algunos sistemas darán una mayor importancia a ciertos atributos dependiendo de sus características y de las circunstancias en las cuales se usará la aplicación.

Para hacer Utilizable un sitio web, es necesario saber:

- ¿Cuáles son los objetivos de la compañía con respecto al sitio?
- ¿Quiénes serán los usuarios del sitio?
- ¿Cuáles son sus metas?

2.12.3 Heurísticas de Usabilidad para la Web

Jakob Nielsen, en 1990, enumeró diez heurísticas de usabilidad. En ellas se hacían una serie de recomendaciones para verificar puntos críticos en interfaces de usuario para asegurar que esta tuviera un alto nivel de usabilidad. Estas heurísticas no fueron escritas específicamente para aplicaciones basadas en la Web.

En 1997 Keith Instone retomó las heurísticas de Nielsen y las adaptó específicamente para la Web, estas son:

- Visibilidad del estado del sistema
- Similitud entre el sistema y el mundo real
- Control por parte del usuario y libertad
- Consistencia y cumplimiento de estándares
- Prevención de errores
- Preferencia al reconocimiento frente a la memorización
- Flexibilidad y eficiencia de uso
- Estética y diseño minimalista
- Ayuda para que el usuario reconozca, diagnostique y se recupere de los errores.
- Ayuda y documentación

2.13 Active Server Pages (ASP)

Active Server Pages (ASP), es una tecnología propietaria de Microsoft. Se trata básicamente de un lenguaje de tratamiento de textos (scripts), basado en Basic. Sus características son:

- Se utiliza casi exclusivamente en los servidores Web de Microsoft (*Internet Information Server* y *Personal Web Server*).
- Los *scripts* ASP se ejecutan, en el servidor donde esta alojada la página web.
- Puede utilizarse conjuntamente con HTML y Javascript para realizar tareas interactivas y en tiempo real con el cliente.
- Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos.
- Es totalmente ampliable gracias a que es compatible con la tecnología ActiveX.

2.13.1 Internet Information Server (IIS)

Para ejecutar páginas que contengan código ASP necesitamos un servidor, existen dos que son accesibles: Personal Web Server y el IIS, ambos pertenecen a Microsoft. Servicios de Internet Information Server (IIS) es el servicio Web de Windows (NT, 2000 y XP Professional) que facilita la publicación de información en una intranet.

El IIS 5.1 para Microsoft Windows XP Professional ofrece la eficacia de las páginas Web a Windows. Con ayuda de IIS, se puede crear aplicaciones para publicar de forma segura como información en la Web a fin de mejorar la forma en que una organización comparta información. IIS es una plataforma segura para crear y distribuir soluciones de comercio electrónico y aplicaciones críticas en la Web.

La utilización de Windows XP Professional con IIS instalado proporciona un sistema operativo personal y de desarrollo que permite:

- Configurar un servidor Web personal.
- Compartir información con un equipo.
- Tener acceso a las bases de datos.
- Desarrollar aplicaciones para el Web.

El IIS viene incluido como un componente de Windows en el CD de instalación de Windows XP Professional.

2.13.2 JavaScript

Es un lenguaje de tipo script compacto, *basado en objetos y guiado por eventos* diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de internet. Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como ser: pedir datos, confirmaciones, mostrar mensajes, crear animaciones ó comprobar campos.

Al ser un lenguaje de tipo script significa que no es un lenguaje compilado, es decir, tal cual se va leyendo se ejecuta por el cliente. Al estar basado en objetos, habrá que comentar qué son los objetos, aunque no vamos a tener toda la potencia que estos nos dan en Java, sólo algunas de sus características.

Estar guiado por eventos significa que no se tendrá un programa que se ejecute de principio a fin en cuanto carguemos la página web. Significa que, cuando en el navegador suceda algún evento, entonces, si lo hemos decidido así, pasará algo. Y ese algo será alguna función JavaScript. Al ser guiado por eventos, no tenemos una función principal que se ejecute por delante de las demás, sino que tendremos funciones.

El programa que va a interpretar los programas JavaScript es el propio navegador, lo que significa que si en el que trabajamos no soporta JavaScript, no podremos ejecutar las funciones que programemos. Desde luego, Netscape y Explorer lo soportan, el primero desde la versión 2 y el segundo desde la versión 3 (aunque las primeras versiones de Explorer 3 soportaban una versión propia del lenguaje llamada JScript y con la que eran incompatibles con JavaScript).

2.14 Microsoft® SQL – SERVER 2000

Es uno de los productos de Microsoft, que aprovecha la sólida base establecida por SQL Server 6.5 y SQL 7. El SQL – SERVER 2000 es el RDBMS ideal para un amplio espectro de clientes corporativos y productores independientes de software, inmersa en la creación de aplicaciones empresariales.

Las necesidades y requisitos del cliente han dado lugar a innovaciones significativas en el producto SQL Server versión 2000, entre las que se incluyen la facilidad de uso, escalabilidad, fiabilidad y almacenamiento de datos.

2.14.1 Características del SQL – SERVER 2000

Entre las más sobresalientes cabe citar las siguientes:

- Primera base de datos que soporta la configuración automática y la auto-optimización.
- Primera base de datos con un servidor OLAP integrado.
- Contiene servicios de transformación de datos (Data Transformation Services, DTS) integrados.
- Ofrece administración multiservidor para un gran número de servidores.
- Acceso universal a los datos (Universal Data Access), la estrategia de Microsoft para permitir el acceso de alto rendimiento a una gran cantidad de fuentes de información.

2.15 Calidad de Software

“Calidad de software es el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentado y de las características implícitas que se esperan de todo software desarrollado profesionalmente.”

Esta definición sirve para destacar 3 puntos importantes:

1. Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con estos requisitos es una falta de calidad.
2. Los estándares especificados definen un conjunto de criterios de desarrollo, que guían la ingeniería del software. Si no se siguen los criterios, el resultado será, casi seguramente, la falta de calidad.
3. A menudo se soslaya un conjunto de requisitos implícitos. Si el software cumple con sus requisitos explícitos pero no con los implícitos, la calidad del software estará en duda. **[PRESSMAN, 2006]**

2.15.1 Factores de calidad del estándar ISO 9126

El estándar ISO 9126 se desarrolló como un intento por identificar los atributos ó factores de calidad para el software de computadora, aunque estos factores no necesariamente se prestan a la *medida directa*⁴, sin embargo proporciona una base valiosa para medidas indirectas y una lista de verificación excelente para evaluar la calidad de un sistema.

Este estándar identifica seis atributos de la calidad:

Funcionalidad: Grado en que el software satisface las necesidades que indican los siguientes subatributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

Confiabilidad: Cantidad de tiempo en que el software está disponible para usarlo.

Facilidad de uso: La facilidad con que se usa el software.

Eficiencia: Grado en que el software emplea en forma óptima los recursos del sistema.

Facilidad de mantenimiento: Facilidad con que se repara el software.

Portabilidad: Facilidad con que se lleva el software de un entorno a otro.

De estos factores deben establecerse cuales son determinantes para la medición de la calidad de software de un determinado producto, definiendo así una línea base para las métricas de calidad de software a ser utilizadas.

⁴ Una medida directa indica que sólo es posible contar un valor que proporciona una indicación directa del atributo que se examina. (Ej. Medir contando el número de líneas de código).

3. DISEÑO DEL SISTEMA

En el presente capítulo, se desarrollara las fases correspondientes al análisis, diseño, implementación y pruebas, siguiendo el proceso de Desarrollo RUP detallado en el Marco Teórico.

3.1 Fase de Inicio

En esta fase se determina los casos de uso y se delimita el alcance del proyecto, identificando así las entidades externas con los que el sistema interactuará (actores), se hará un análisis de los requerimientos del usuario realizando una descripción de los casos de uso mas significativos.

En el proceso de recopilación de la información para el desarrollo de este trabajo se tuvo que recurrir a entrevistas personales con los Usuarios del sistema a desarrollar.

3.1.1 Situación actual del sistema

En la siguiente figura observamos el funcionamiento actual de la institución.

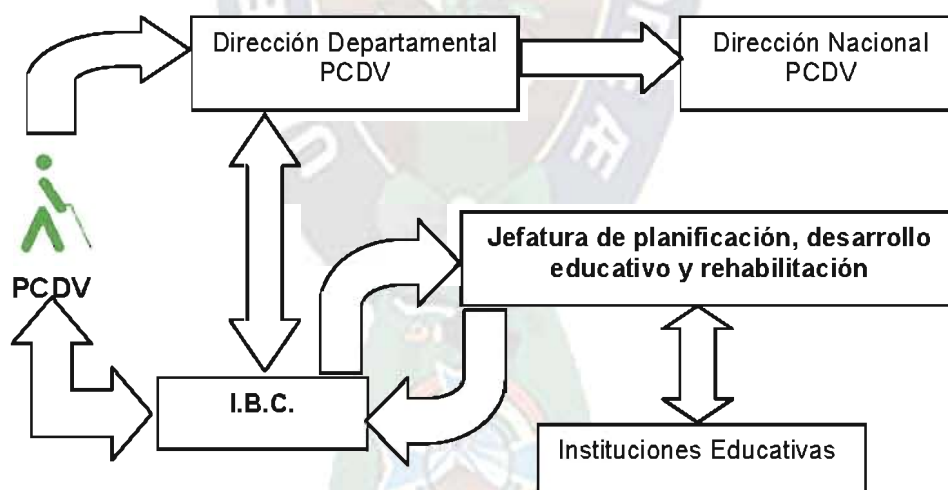


Figura 3.11: Funcionamiento actual de la institución.

En la Figura 3.11 vemos las relaciones que tiene la Jefatura de planificación, desarrollo educativo y rehabilitación con otras entidades, se observa que ésta jefatura es la encargada de realizar la planificación del desarrollo educativo y la rehabilitación de las personas con discapacidad visual. Para lo cual trabaja conjuntamente con las Instituciones Educativas como ser: Escuelas, Centros de Rehabilitación, ONG's, Prefectura de La Paz y diversas Instituciones.

En la figura siguiente se muestra el sistema actual para el registro del afiliado en la Jefatura de planificación, desarrollo educativo y rehabilitación.

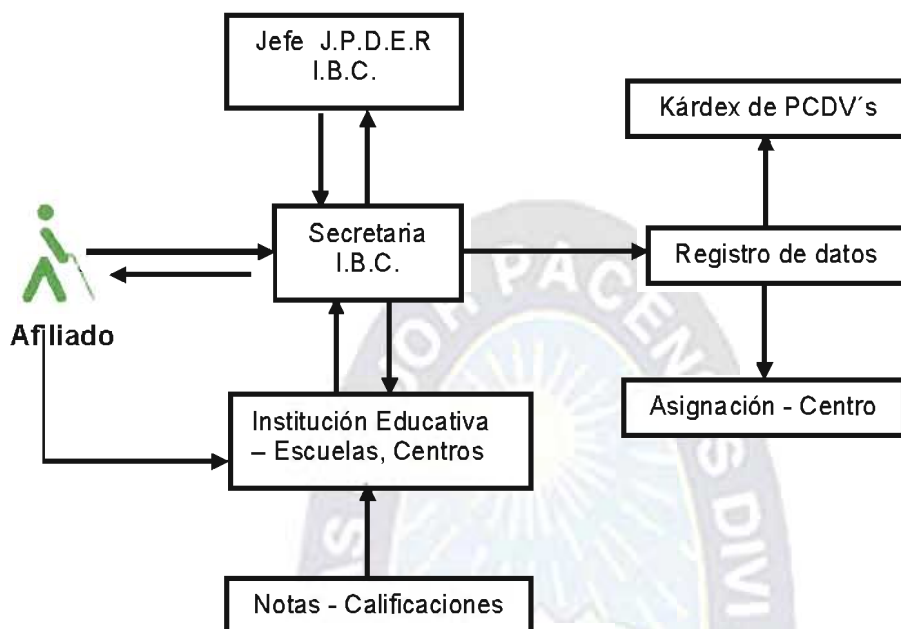


Figura 3.12: Sistema actual para el registro de la PCDV

En el proceso de Registro todos los procedimientos son realizados de forma manual, no se tiene un archivo lógico de almacenamiento de datos, todo el almacenamiento de datos es realizado de forma física en carpetas que se guardan en estantes. Para actualizar los datos de una carpeta se tiene que buscar manualmente y anexarla a su carpeta, lo cual lleva a una demora en el proceso actualización y modificación de datos.

3.1.2 Casos de uso y su descripción

Los casos de uso son un importante artefacto del análisis de requerimientos, pero realmente no están orientados a objetos. Ponen de relieve la vista del dominio a partir de un proceso. [LARMAN, 1999]

Con la aplicación de los casos de uso se puede describir y dibujar de manera superficial el modelo de casos de uso resultante del análisis de requerimientos.

Primeramente identificamos los actores principales que interactúan en el Sistema.

➤ **Persona con Discapacidad Visual (PCDV)**

La PCDV es el principal actor ya que el sistema será alimentado directamente por los datos que este actor le proporcione.

➤ **Administrador**

Es la persona encargada de administrar el sistema, responsable del registro de los datos de la PCDV y la actualización de los mismos.

➤ **Actores Externos**

Son entidades que requieren información del Sistema de Registro y Seguimiento a Nivel Nacional de personas con discapacidad visual, para su correcto funcionamiento. Por ejemplo: Las Escuelas y los centros de Rehabilitación.

3.1.2. a) Definición de los casos de uso de alto nivel

En los casos de uso de alto nivel se describen claramente los principales procesos globales.

La descripción del caso de uso para la solicitud de registro, que es realizado por la PCDV, se muestra en la Tabla 3.1.

Tabla 3.1 Descripción de los casos de uso de solicitud de registro

CASO DE USO:	SOLICITUD DE REGISTRO
Actores:	PCDV, Administrador
Tipo:	Primario
Descripción:	La PCDV presenta su currículum y el documento de afiliación para registrarse en el Sistema y el administrador lo registra, le asigna una escuela para su rehabilitación.

Cuando la PCDV esta registrada, se procede a actualizar su ficha personal proporcionado por el Sistema, como se observa en la Tabla 3.2.

Tabla 3.2: Descripción del caso de uso de generación y actualización de la ficha personal

CASO DE USO:	GENERACIÓN Y ACTUALIZACIÓN DE FICHA PERSONAL
Actores:	Administrador
Tipo:	Primario
Descripción:	El administrador procede a llenar y/o actualizar los datos en la ficha personal de la PCDV.

La descripción del caso de uso para el proceso de Seguimiento Académico, que es realizado por el administrador y donde interviene la PCDV, se muestra en la Tabla 3.3.

Tabla 3.3: Descripción del caso de uso de Seguimiento Académico.

CASO DE USO:	SEGUIMIENTO ACADÉMICO
Actores:	Administrador, PCDV, Escuela de Rehabilitación
Tipo:	Primario
Descripción:	El administrador asigna a un centro de rehabilitación a la PCDV, controla el rendimiento académico y almacena los resultados que la PCDV obtiene de su rehabilitación y de las capacitaciones en diferentes áreas que recibió.

El administrador es el encargado de elaborar reportes solicitados por otras unidades pertenecientes al IBC ó por las escuelas de rehabilitación, como se ven la Tabla 3.4

Tabla 3.4: Descripción del caso de uso de generación de reportes

CASO DE USO:	GENERACIÓN DE REPORTES
Actores:	Administrador
Tipo:	Primario
Descripción:	El administrador elabora reportes solicitados bien por la administración del IBC, por otra unidad ó bien por algún centro de rehabilitación que lo requiera.

3.1.2.b) Diagrama de caso de uso

Este diagrama tiene por objeto ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan. [LARMAN, 1999]

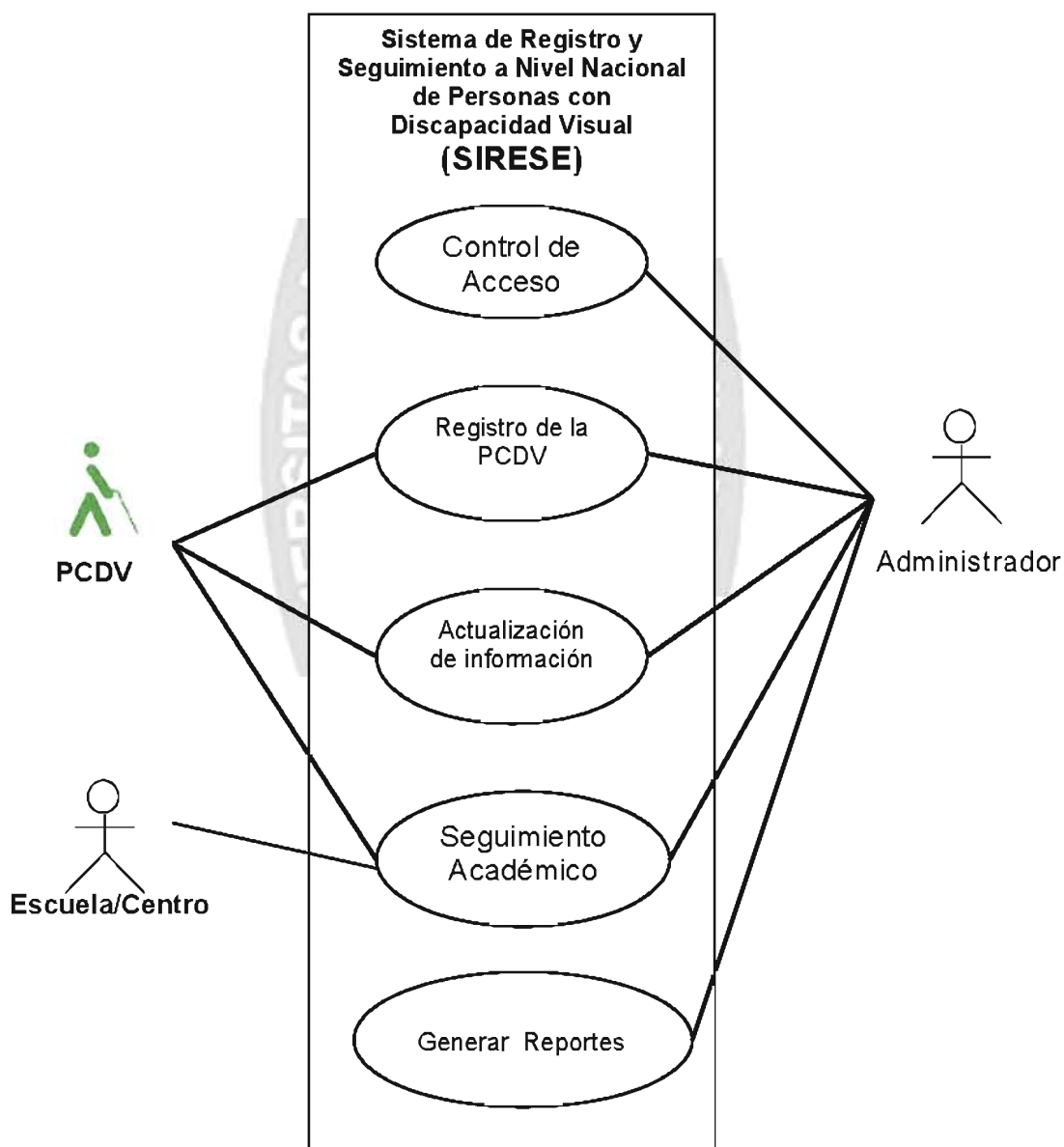


Figura 3.13: Diagrama de caso de uso para el Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual (SIRESE)

3.2 Fase de Elaboración

En esta fase se definen la visión del producto y su arquitectura, especificando los requisitos del sistema para crear una base arquitectónica sólida, para esto se utiliza los diferentes modelos como ser: Modelo de casos de uso, Casos de Uso Expandido, Modelo Dinámico, Modelo Funcional, Modelo de Dominio y el Modelo de Aplicación.

3.2.1 Modelo de casos de uso

En la figura 3.14 se observa el modelo de casos de uso para el proceso de Registro del sistema.

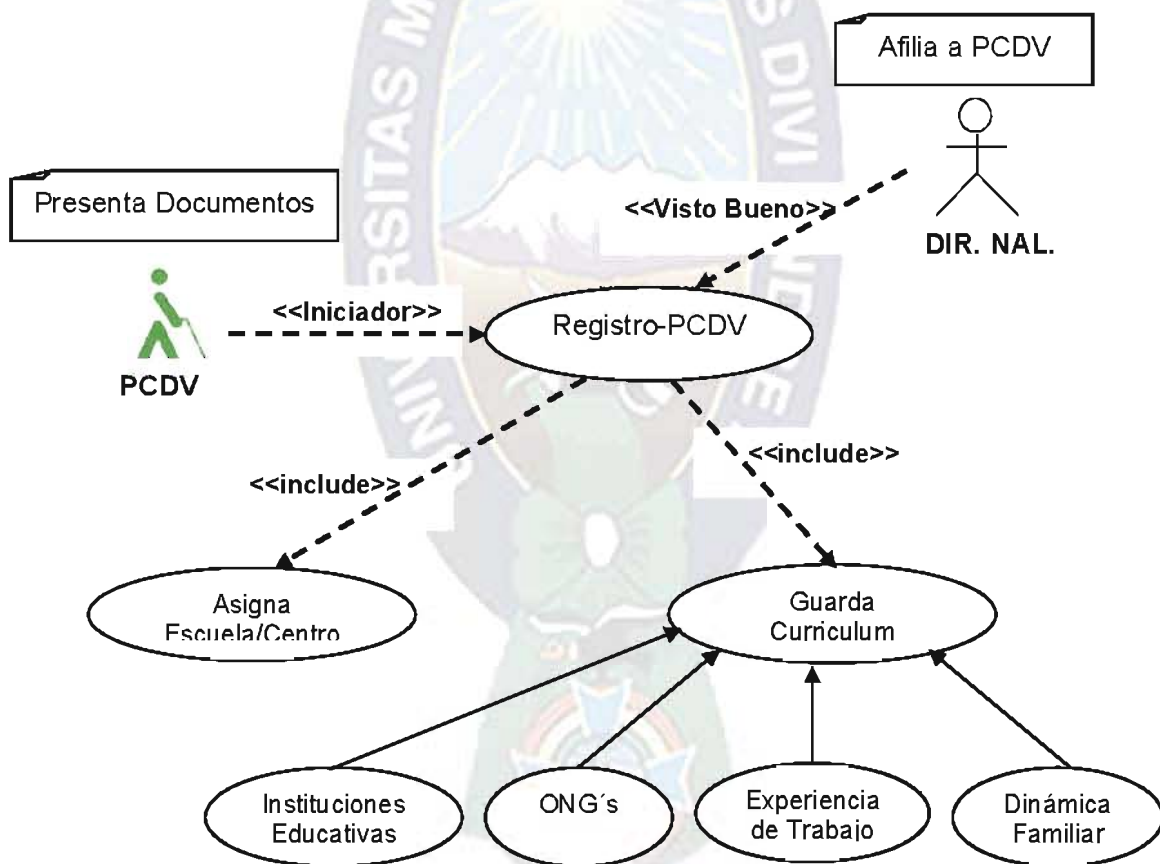


Figura 3.14: Diagrama de caso de uso para el proceso de Registro.

También se considera el modelo de casos de uso para el proceso de Seguimiento Académico en la figura 3.15.

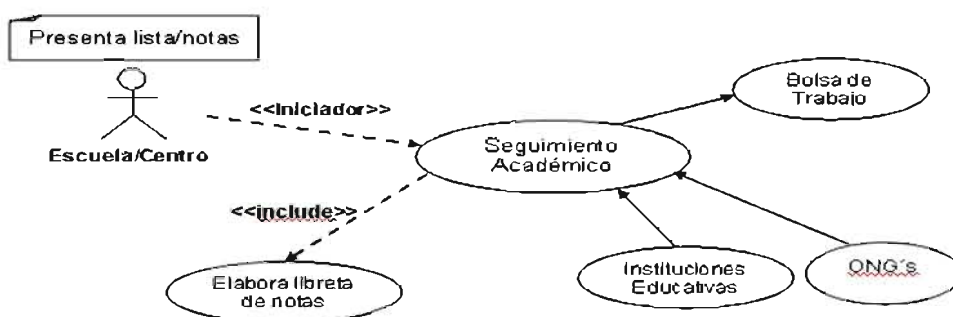


Figura 3.15: Diagrama de caso de uso para el proceso de Seguimiento Académico.

3.2.2 Casos de Uso Expandidos

Los casos de uso que se consideren los más importantes y que se considere que son los que más influyen al resto, se describen a un nivel más detallado, porque suelen ser útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos.

Tabla 3.5: Caso de uso expandido de solicitud de registro

Casos de uso:	Solicitud de Registro	
Actores:	PCDV, Administrador	
Propósito:	Registrar los datos de la PCDV.	
Resumen:	Una vez que la PCDV presente su currículum y su documento de afiliación, se le registrará y asignará a una escuela para que empiece con su rehabilitación.	
Tipo:	Primario y esencial	
Referencia Cruzada:		
Curso normal de acciones		
Acción del Actor	Respuesta del Sistema	
1. Este caso de uso empieza cuando la PCDV solicita su registro en el sistema y presenta su currículum y su documento de afiliación.	2. El sistema almacena los datos proporcionados y crea una ficha personal para el funcionario.	
4. La PCDV está registrado y se le asigna a un centro de rehabilitación.	3. El sistema guarda los datos del administrador para controlar su acceso al sistema.	
Cursos Alternos		
Línea 1: Introducción de datos no válidos, por lo que se puede volver a hacer la solicitud		

Tabla 3.6: Caso de uso expandido de generación y actualización de Ficha Personal

Casos de uso:	Generación y Actualización de Ficha Personal	
Actores:	Administrador	
Propósito:	Generar y actualizar la ficha personal de la PCDV.	
Resumen:	El Administrador debe generar la ficha personal de cada una de las PCDV y mantenerlos actualizados.	
Tipo:	Primario y esencial	
Referencia Cruzada:		
Curso normal de acciones		
Acción del Actor	Respuesta del Sistema	
<p>1. Este caso de uso empieza cuando el Administrador procede a generar ó actualizar la ficha personal de una PCDV.</p> <p>4. El Administrador obtiene confirmación del registro ó actualización de los datos</p>	<p>2. El sistema genera ó actualiza la ficha personal, almacenando la información en la Base de Datos.</p> <p>3. El sistema registra en una bitácora todos los cambios realizados por el Administrador.</p>	
Cursos Alternos		
Línea 1: Introducción de datos no validos, por lo que se puede volver a llenar los datos.		

Tabla 3.7: Caso de uso expandido de Seguimiento Académico.

Casos de uso:	Seguimiento Académico	
Actores:	Administrador, PCDV, Centro de Rehabilitación	
Propósito:	Realizar un seguimiento académico, controlando el desempeño de la PCDV y almacenando los resultados que obtiene de su rehabilitación y/o capacitación en otras áreas.	
Resumen:	El Administrador debe registrar y almacenar los resultados obtenidos por la PCDV durante su rehabilitación en su ficha personal.	
Tipo:	Primario y esencial	
Referencia Cruzada:		
Curso normal de acciones		
Acción del Actor	Respuesta del Sistema	
<p>1. Este caso de uso empieza cuando el Administrador asigna a la PCDV a una Escuela, para su rehabilitación.</p> <p>3. El Administrador registra las calificaciones obtenidas por la PCDV durante su rehabilitación.</p>	<p>2. El sistema almacena la información de la PCDV y el centro al cual se le asigna en la Base de Datos.</p> <p>4. El sistema clasifica la información en la ficha personal de la PCDV durante su rehabilitación.</p>	
Cursos Alternos		
<p>Línea 1: Introducción de datos no validos, por lo que se puede volver a llenar los datos.</p> <p>Línea 2: En caso de Abandono ó Retiro por parte de la PCDV durante su rehabilitación, se considera la re inserción en el mismo ú otro centro de rehabilitación, considerando el tiempo de su rehabilitación.</p>		

Tabla 3.8: Caso de uso expandido de generación de reportes

Casos de uso:	Generación de reportes	
Actores:	Administrador	
Propósito:	Generar reportes estadísticos sobre las PCDV registradas en el sistema.	
Resumen:	La generación de reportes responde a los requerimientos del IBC y de la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación, además de otras instancias pertenecientes al IBC.	
Tipo:	Primario y esencial	
Referencia Cruzada:		
Curso normal de acciones		
Acción del Actor	Respuesta del Sistema	
<p>1. Este caso de uso empieza cuando el Administrador genera reportes.</p> <p>3. El Administrador obtiene el reporte requerido en poco tiempo y con información actualizada.</p>	<p>2. El sistema mediante su Motor de reportes genera los datos requeridos por el usuario.</p>	
Cursos Alternos		
Línea 1: Si los datos no existen, entonces el sistema no generará el reporte solicitado.		

3.2.3 Modelo dinámico

El Modelo Dinámico, modela los aspectos del sistema que tienen que ver con el tiempo y los cambios, modela el Control, aspecto del sistema que describe la secuencia de operaciones que ocurren como respuesta a unos estímulos.

Conceptos principales del Modelo Dinámico:

- Sucesos o Eventos: representan los estímulos externos.
- Estados: representan valores de los objetos.

3.2.3.1 Diagrama de Estado

Este tipo de diagrama relaciona Sucesos y Estados. El siguiente Estado depende del Suceso recibido y del Estado actual. [RUMBAUGH, 1998]

El diagrama de estados para la verificación del Login y Password del usuario, muestra los pasos que debe seguir un usuario para tener acceso al sistema, como se muestra en la Figura 3.16

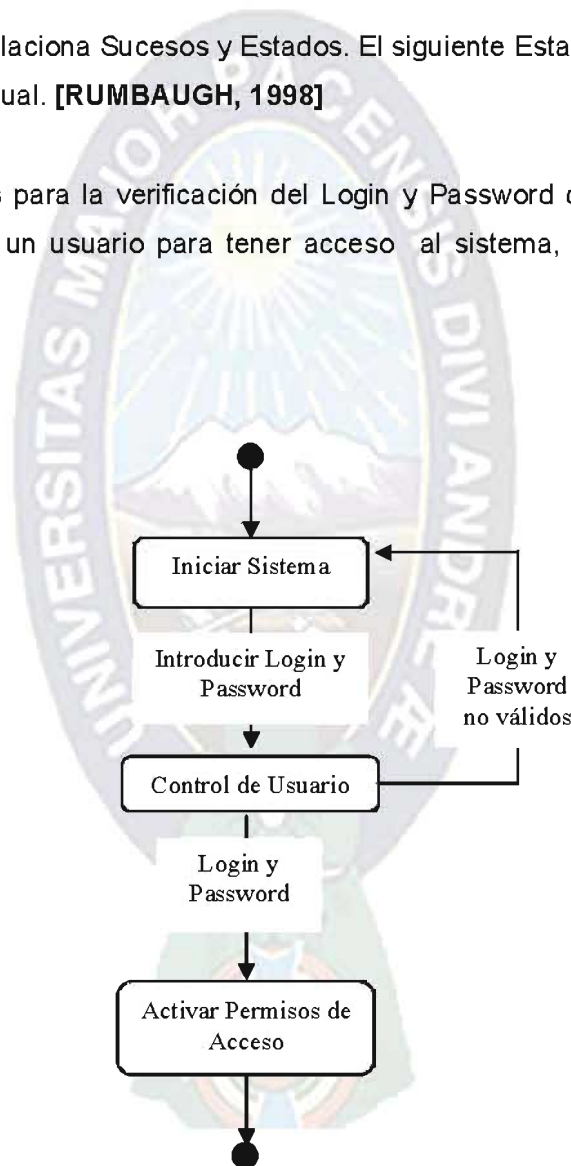


Figura 3.16: Diagrama de estado correspondiente a la verificación de usuario

El diagrama de estado para la generación o actualización de la ficha personal de cada PCDV se muestra en la Figura 3.17.

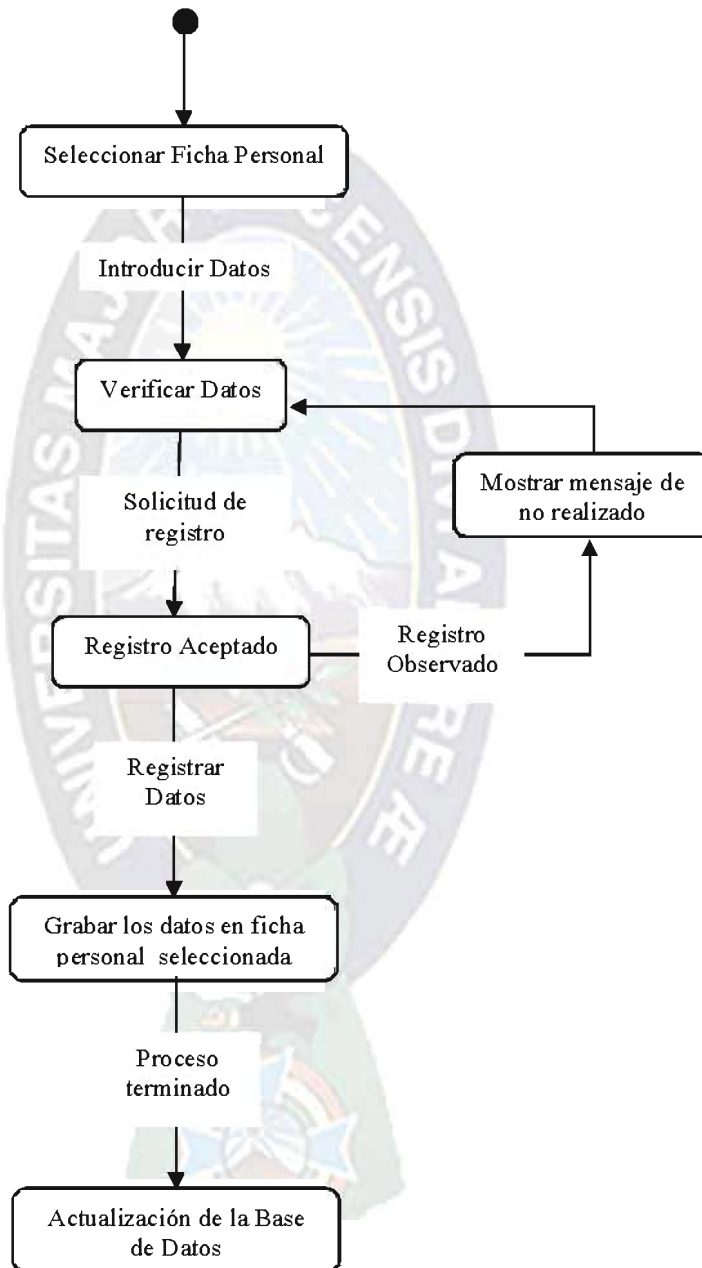


Figura 3.17: Diagrama de estado correspondiente a la generación ó actualización De la Ficha Personal de una PCDV.

Los pasos a seguir en el proceso de Seguimiento Académico, se muestran en el siguiente diagrama de estado, ver Figura 3.18.

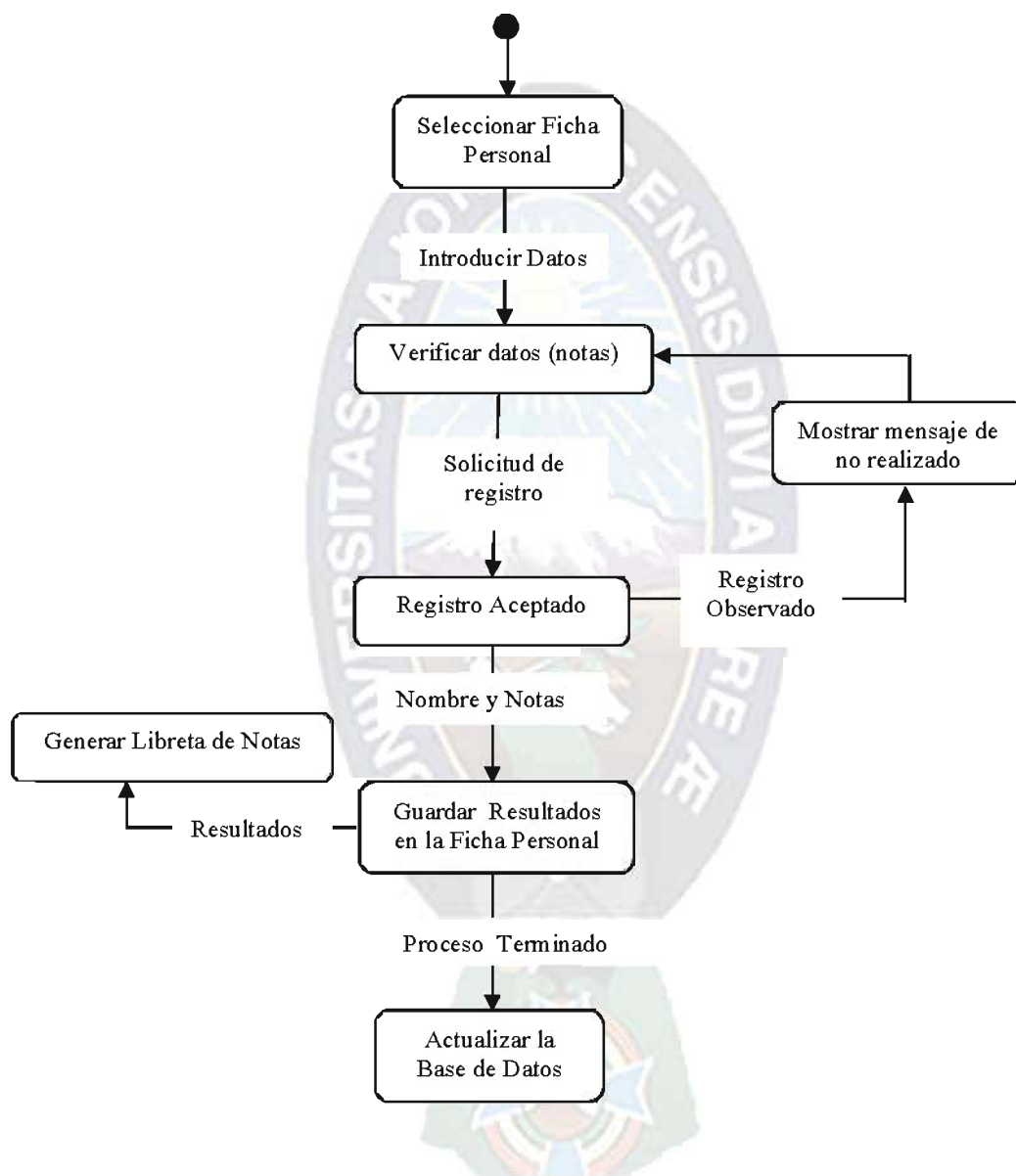


Figura 3.18: Diagrama de estado correspondiente al proceso de Seguimiento Académico.

Los pasos para realizar o generar reportes solicitados por alguna instancia se muestran en la Figura 3.19.

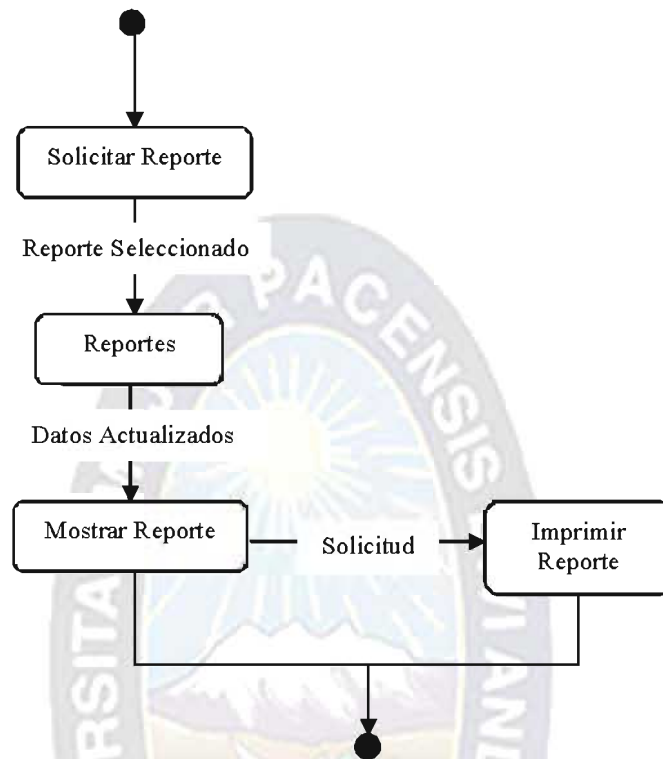


Figura 3.19: Diagrama de estado correspondiente a la solicitud de Reportes

3.2.4 Modelo funcional

El modelo funcional describe los cálculos existentes del sistema sin describir ni como ni cuando se calculan.

3.2.4.1 Diagramas de secuencia

El diagrama de secuencia de un sistema es una representación que muestra, en un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema. A todos los sistemas se les trata como una caja negra; los diagramas se centran en los eventos que trascienden las fronteras del sistema y que fluyen de los actores a los sistemas. [LARMAN, 1999]

Con lo desarrollado hasta ahora se tiene una perspectiva como para poder desarrollar los diagramas de secuencia, a continuación se detallara las siguientes figuras.

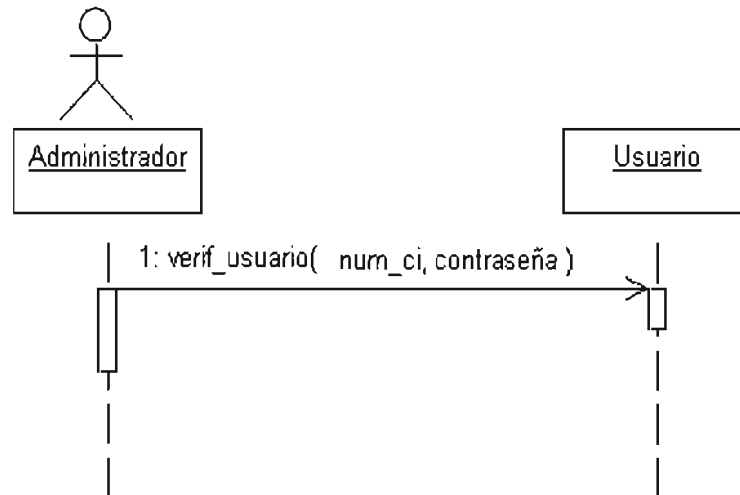


Figura 3.20: Diagrama de secuencia correspondiente al control de usuario

El sistema se inicia, cuando la PCDV solicite generar o actualizar su ficha personal, como se ve en la Figura 3.21.



Figura 3.21: Diagrama de secuencia correspondiente a la generación y actualización de la ficha personal

Para el Seguimiento Académico, es el Administrador el que se encarga de este subsistema como se ve en la Figura 3.22.

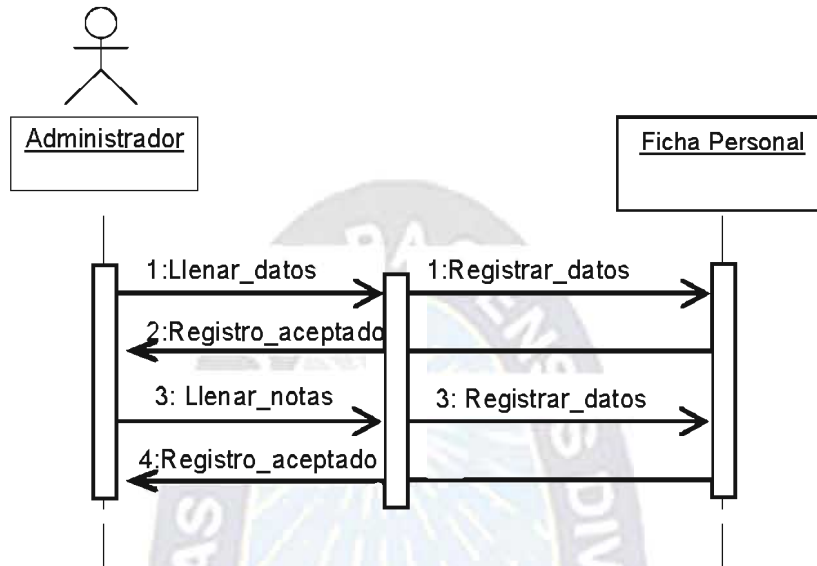


Figura 3.22: Diagrama de secuencia correspondiente al Seguimiento Académico.

La generación de reportes es realizada por el administrador, estos reportes cumplen los requerimientos de algunas unidades pertenecientes al IBC ó de otro solicitante, esta generación se la observa en la Figura 3.23.

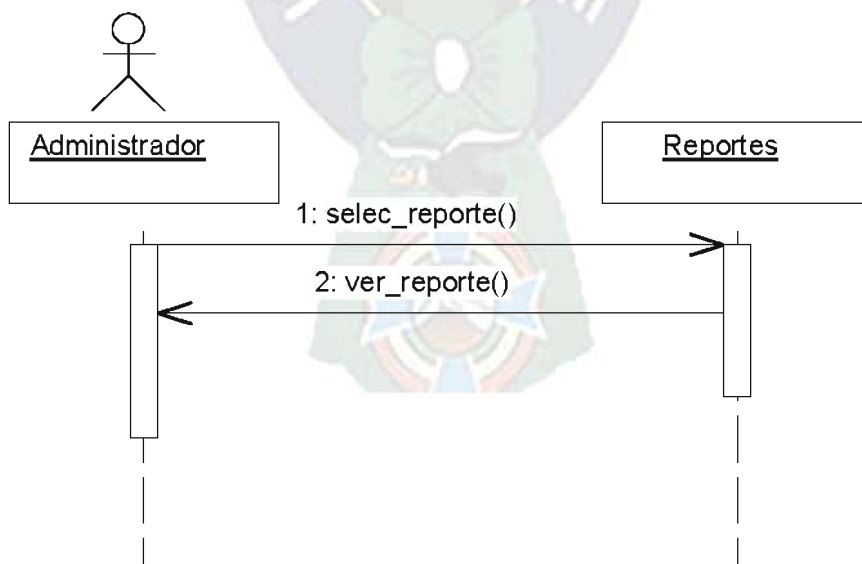


Figura 3.23: Diagrama de secuencia correspondiente a la generación de reportes.

3.2.5 Modelo de dominio

Este modelo es creado explorando el dominio general y adquiriendo conocimiento de las tareas que serán efectuadas. UML nos facilita modelos para entender el dominio del problema, por esta razón utilizamos esta herramienta.

3.2.5.1 Modelo conceptual

La designación de modelo conceptual ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software. Un modelo conceptual es una descripción del dominio de un problema real, no es una descripción del diseño del software. Una regla práctica consiste en que un modelo conceptual no es absolutamente correcto ni erróneo, sino de mayor o menor utilidad; es una herramienta de la comunicación. [LARMAN, 1999]

Primeramente identificamos los conceptos que están sujetos a la restricción de los requerimientos, es mejor exagerar y especificar un modelo conceptual con muchos conceptos refinados que no especificarlo cabalmente. Ver Figura 3.24.

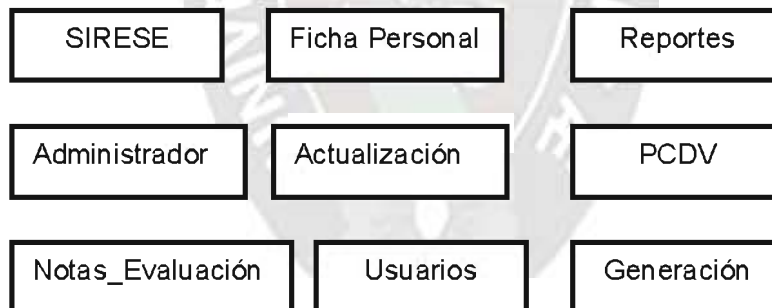


Figura 3.24 Conceptos

Cuando asignamos nombres a los conceptos y a los atributos, utilizamos el vocabulario del dominio, porque el modelo conceptual es una especie de mapa de conceptos o cosas de un dominio. En la Figura 3.25 detallamos el Modelo Conceptual relacionado al dominio del problema tomando en cuenta la multiplicidad de sus instancias y especificando sus atributos.

3.2.5.2 Diccionario Modelo

El diccionario modelo o glosario es semejante a un diccionario de datos, éste incluye y define todos los términos que necesiten explicación para mejorar la comunicación y disminuir el riesgo de malos entendidos. En la tabla 3.9 se ven algunos términos que requieren de una explicación para su mejor comprensión.

Tabla 3.9 Diccionario modelo

TERMINO	CATEGORIA	COMENTARIOS
Usuario.num_ci_usu: Entero	Atributo	Numero único que identifica a un solo usuario del sistema, que es el administrador.
PCDV_Rehab.ingreso: Date	Atributo	Fecha en la que la PCDV es registrada en la escuela para su rehabilitación.
PCDV_Rehab.año_rehab: Entero	Atributo	Especifica la cantidad de años que la PCDV tiene en el proceso de rehabilitación en un determinado centro ó escuela.
Emitir_Reporte	Caso de uso	Es la generación de reportes mediante un módulo, el cual se alimenta de los datos actualizados de las PCDV.
PCDV_Rehab.estado: Texto	Atributo	Especifica el estado de la PCDV en la escuela: Abandono, retiro, traslado, en curso.
Escuelas.tipo: Texto	Atributo	Especifica el tipo de escuela ó centro de rehabilitación: Estatal, Privada, ONG's.
Responsable	Tipo	Se almacenan los datos de la persona que es directamente responsable por la PCDV, en caso de surgir alguna emergencia ó contratiempo con la PCDV, debe comunicarse directamente con su responsable ó encargado.
Responsable.fecha: Date	Atributo	Fecha en la cual la persona acepta la responsabilidad de la PCDV .
PCDV_Rehab.observaciones: Texto	Atributo	Especificación del estado de la PCDV en la escuela ó centro de rehabilitación. Por ejemplo: Traslados departamentales.
Usuario	Tipo	Es la persona encargada de administrar el sistema, responsable del registro de las PCDV y de la actualización de sus datos.

3.3 Fase de Construcción

Esta fase requiere varios ciclos de desarrollo mediante los cuales se va constituyendo el sistema. El objetivo final es obtener un sistema funcional de software que satisfaga y este conforme a los requerimientos de los usuarios.

3.3.1 Modelo de aplicación

Este modelo es construido sobre el modelo de dominio examinando los casos de uso del dominio. Para ello también nos ayudamos de la herramienta UML.

3.3.1.1 Contrato de operaciones

El comportamiento de un sistema es una descripción de lo que hace, sin explicar cómo lo hace. Los contratos son documentos muy útiles que describen el comportamiento de un sistema a partir de cómo cambia el estado e un sistema cuando se llama una operación suya.

Un contrato es un documento que describe lo que una operación se propone lograr. Los contratos suelen expresarse a partir de los cambios de estado de las precondiciones y las poscondiciones. Puede elaborarse un contrato para un método de una clase de software ó para una operación más global del sistema. **[LARMAN, 1999]**

El contrato de operación del sistema describe los cambios del estado del sistema total cuando se llama a una de sus operaciones.

A continuación en las siguientes tablas se desarrollan los contratos más elementales correspondientes al comportamiento del sistema.

Tabla 3.10 Contrato para la verificación de usuario

1. Contrato para la verificación de usuario	
Nombre	Verif_usuario(num_ci, contraseña)
Responsabilidades	Verifica si el usuario es válido o no válido.
Tipo	Proceso
Referencias Cruzadas	
Notas	El usuario que quiera acceder al sistema lo hará solamente si esta registrado como el administrador del sistema.
Excepciones	Solamente el administrador ó usuario autorizado podrá registrar una nueva PCDV ó eliminar una.
Salida	La PCDV estará registrada en el sistema.
Precondiciones	Para que un usuario acceda al sistema, el administrador deberá registrarlo y asignarle permisos de acceso, delegando así sus responsabilidades al nuevo usuario asignado.
Poscondiciones	El administrador realizó las tareas requeridas por la PCDV.

Tabla 3.11: Contrato correspondiente a la generación ó actualización de la ficha personal

1. Contrato para la generación ó actualización de la ficha personal	
Nombre	Llena_actualiza(num_ci)
Responsabilidades	Validar y registrar los datos referentes a la ficha personal de cada PCDV.
Tipo	Sistema
Referencias Cruzadas	
Notas	Los datos se almacenaran en tablas clasificadas.
Excepciones	Si el usuario introduce datos de forma incorrecta, se mostrara mensaje de advertencia.
Salida	Muestra mensaje de registro correcto ó incorrecto en el sistema.
Precondiciones	La información de cada PCDV se tendrá en la base de datos referenciado por su carnet de identidad de manera única.
Poscondiciones	Se registró toda la información del funcionario.

Tabla 3.12: Contrato correspondiente para el seguimiento académico.

1. Contrato para el seguimiento académico	
Nombre	Actualizar_rehabilitación()
Responsabilidades	Registrar información sobre la rehabilitación y capacitación en diferentes áreas, de cada PCDV que se encuentre registrado en el sistema.
Tipo	Proceso
Referencias Cruzadas	
Notas	El administrador es el encargado de asignar a una PCDV a una escuela o centro de rehabilitación, conjuntamente debe realizar el seguimiento de dicha rehabilitación.
Excepciones	Solamente el administrador debe actualizar la información sobre rehabilitación y capacitación de cada PCDV, con los datos que le proporcionen los centros y las escuelas, según sea el caso. Considerando la documentación presentada por cada PCDV que sea respaldada por la institución en la cual recibió la capacitación.
Salida	Muestra mensaje de finalización del proceso si los datos son válidos.
Precondiciones	Los años de rehabilitación no deben exceder los 4 años de cursación como máximo, incluyendo casos especiales, tomando en cuenta: los retiros, abandonos y cambios de institución. En cuyo caso se hará una suma de los años que curso en cada uno de los casos.
Poscondiciones	Se obtuvo información actualizada del rendimiento académico en la rehabilitación y capacitación de cada una de las PCDV registradas en el sistema.

Tabla 3.13: Contrato correspondiente a la generación de reportes

1. Contrato para la generación de reportes	
Nombre	Selec_reporte()
Responsabilidades	Generar reportes al SIRESE
Tipo	Sistema
Referencias Cruzadas	
Notas	El administrador puede elegir entre guardar el reporte o imprimirlo según sus necesidades.
Excepciones	El administrador del sistema es el único que puede realizar reportes con el sistema, previa solicitud de sus superiores inmediatos.
Salida	Muestra los datos actualizados encontrados, si los datos no son válidos el sistema no mostrará ningún resultado.
Precondiciones	El sistema generará los reportes en base a los datos actualizados de las fichas personales de las PCDV que hayan sido registradas en el sistema.
Poscondiciones	Se obtendrá un archivo o una impresión del reporte seleccionado.

3.3.2 Diseño del sistema

El diseño del sistema es la segunda parte de la fase de construcción, su principal objetivo es el diseño lógico del sistema.

3.3.2.1 Casos reales de uso

Un caso real de uso describe el diseño concreto del caso de uso a partir de una tecnología particular de entrada y salida, así como de su implementación global. Por ejemplo, si interviene una interfaz gráfica para el usuario, el caso de uso real incluirá diagramas de ventanas en cuestión y una explicación de la interacción de bajo nivel con los artefactos de la interfaz. [LARMAN, 1999]

3.3.2.2 Secuencias de pantallas

No es necesario realizar los casos reales de uso, una alternativa es generar secuencias de pantallas de la interfaz general para el usuario, incorporando después los detalles en la fase de implementación. Para el sistema tenemos la siguiente secuencia de pantallas básicas:

En la Figura 3.26, se muestra la pantalla de identificación de usuario, en el cual se pide el login y la contraseña de cada usuario, para validar un usuario que entre en nuestra web, la validación se hace mediante una sesión, en caso de que el usuario no este habilitado se muestra un mensaje de notificación, si el usuario es correcto se le habrá una sesión y en cada página que queremos que sea privada se controlara si la sesión esta abierta para permitir el acceso, solamente se permite el acceso al Administrador asignado en la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C., estos accesos son controlados mediante los permisos de acceso que se le son asignados al usuario.

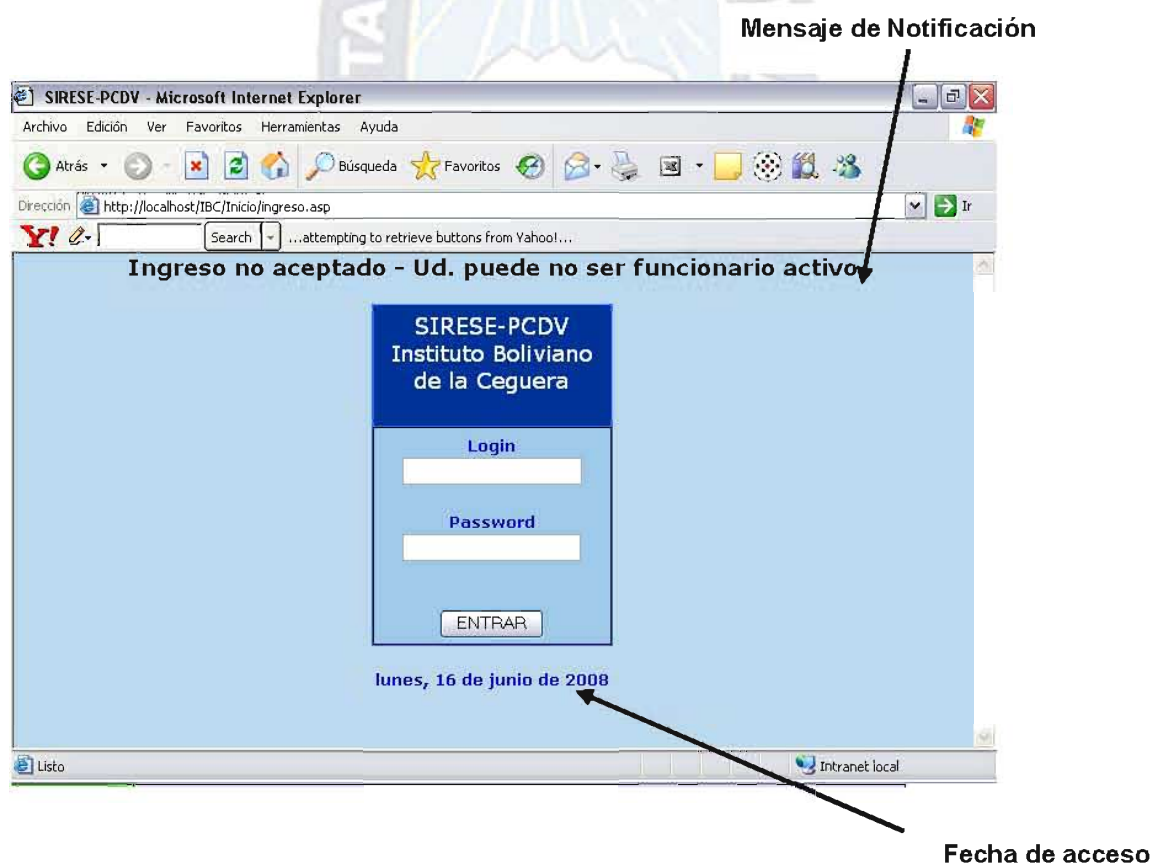


Figura 3.26 Pantalla de identificación de usuario

En la figura 3.27, se tiene la interfaz de usuario, en la cual se presentan las diversas tareas a las cuales el administrador tiene acceso. Se observa en el lado izquierdo, una tabla con el listado de las tareas, las cuales abrirán el proceso requerido en el marco central.

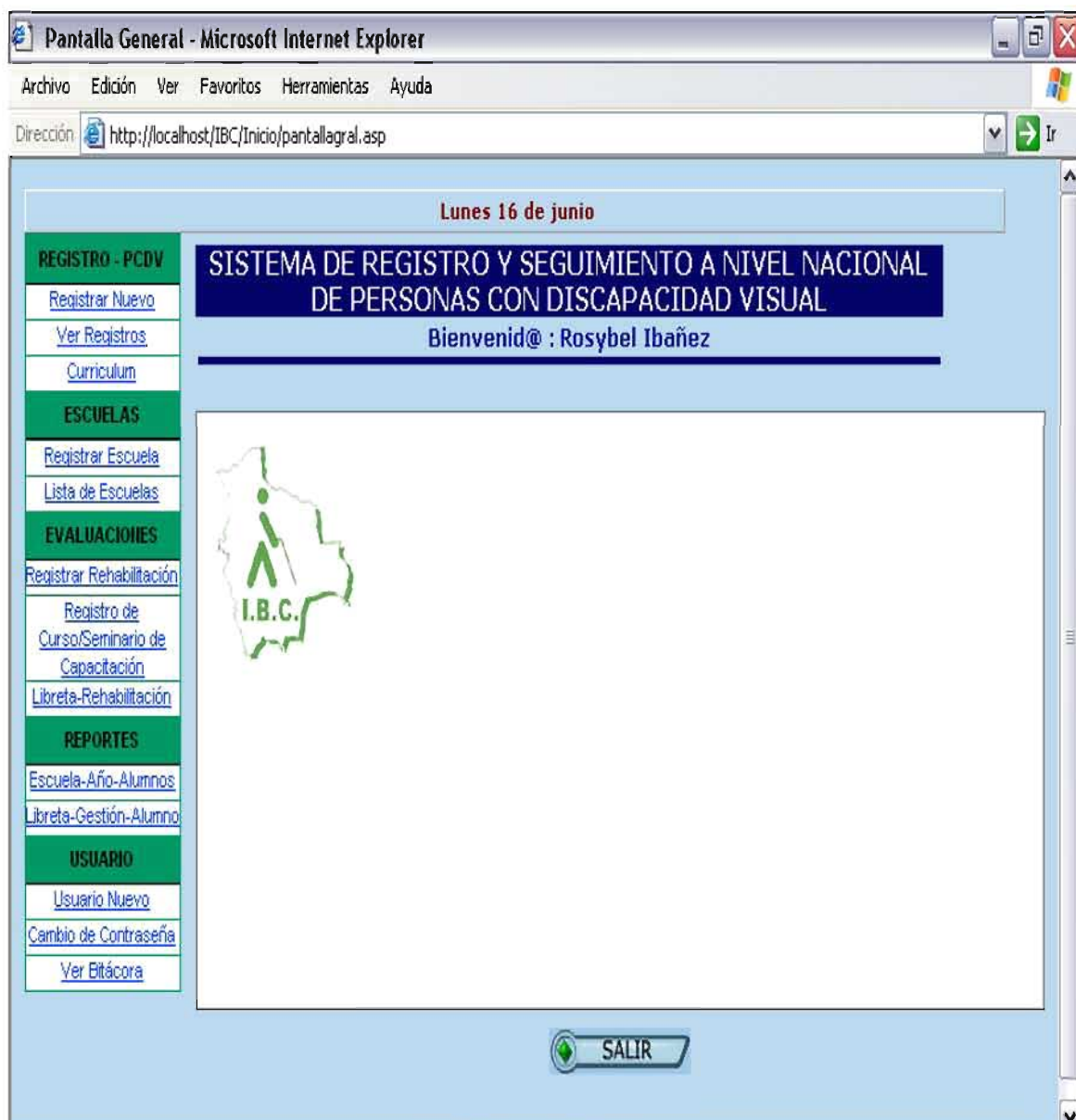


Figura 3.27: Interfaz para el administrador del sistema.

En la figura 3.28, se muestra el Historial del sistema, este historial se actualiza cada vez que el administrador ingrese al sistema, en el momento en que se verifica que el usuario es valido y se le pasa a la interfaz del administrador, el sistema registra hora de ingreso y los datos del administrador, este historial solamente puede ser vista por el administrador pero no así ser modificada. La vista se presenta como un listado en la cual se observa los diferentes accesos por el usuario asignado ó permitido.

REGISTRO - PCDV

- [Registrar Nuevo](#)
- [Ver Registros](#)
- [Curriculum](#)

ESCUELAS

- [Registrar Escuela](#)
- [Lista de Escuelas](#)

EVALUACIONES

- [Registrar Rehabilitación](#)
- [Registro de Curso/Seminario de Capacitación](#)
- [Libreta-Rehabilitación](#)

REPORTES

- [Escuela-Año-Alumnos](#)
- [Libreta-Gestión-Alumno](#)

USUARIO

- [Usuario Nuevo](#)
- [Cambio de Contraseña](#)
- [Ver Bitácora](#)

SISTEMA DE REGISTRO Y SEGUIMIENTO A NIVEL NACIONAL DE PERSONAS CON DISCAPACIDAD VISUAL

Bienvenid@ : Rosybel Ibañez

HISTORIAL DE INGRESO AL SISTEMA

Resultados : 1 a 5 de 50 encontrados.

Fecha	Hora Ingreso	Num. C.I.	Nombre de usuario	Tiem
16/6/2008	1:18:35	1234567	Ibañez Rosybel	0
16/6/2008	0:07:33	4880430	Mújica Torrez Silvia	0
13/6/2008	0:02:10	1234567	Ibañez Rosybel	0
12/6/2008	22:28:01	4880430	Mújica Torrez Silvia	0
12/6/2008	21:40:41	4880430	Mújica Torrez Silvia	0

SALIR

Figura 3.28: Listado del Historial del sistema.

En la Figura 3.29 se muestra el formulario que esta diseñado para recoger los datos personales de cada una de las PCDV, donde se consideran: Domicilio, tipo de vivienda, situación en la que vive la persona y los datos del responsable de la PCDV.

Este formulario se encuentra validado de acuerdo al tipo de datos que se introducen en él, en caso de omitir o fallar un dato, se muestra un mensaje de notificación y se posiciona el cursor en el dato que se omitió ó fallo, este formulario también se carga en el marco central de la ventana, por su tamaño se lo recorre con la barra de desplazamiento, pero se lo muestra en su integridad en la figura.

Datos Personales de la Persona con discapacidad visual - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://localhost/IBC/Inicio/PCDV/datos.asp

DATOS PERSONALES - PCDV

Número de C. I. :	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Primer Nombre:	<input type="text"/>
Segundo Nombre:	<input type="text"/>
Nacionalidad :	<input type="text"/>
Sexo:	<input checked="" type="radio"/> F <input type="radio"/> M
Fecha Nacimiento (dd/mm/aaaa) :	<input type="text" value="9/11/2007"/>
País de Nacimiento:	<input type="text"/>
Ciudad Nacimiento :	<input type="text"/>
Provincia :	<input type="text"/>
Estado Civil:	<input type="text" value="Soltero(a)"/> ▼
Idioma Materno :	<input type="text" value="Español"/> ▼
C. I. Expedido en :	<input type="text"/>
Causa de Ceguera :	<input checked="" type="radio"/> Congénita <input type="radio"/> Adquirida
Grado de Ceguera :	<input type="text" value="Baja Visión"/> ▼
Tiempo de Ceguera:	<input type="text"/> (años)

DIRECCIÓN

Departamento :	<input type="text" value="Beni"/> ▼	Ciudad :	<input type="text"/>
Provincia :	<input type="text"/>	Zona :	<input type="text"/>
Calle ó Avenida:	<input type="text"/>	Número :	<input type="text"/>
Teléfono Domicilio:	<input type="text"/>	Teléfono Celular :	<input type="text"/>
E-Mail (Personal) :	<input type="text"/>	Año - Traslado :	<input type="text"/>
Tipo de Vivienda :	<input type="text" value="Propia"/> ▼	Situación en la que vive :	<input type="text" value="Solo (a)"/> ▼

RESPONSABLE

Apellido Paterno:	<input type="text"/>	Apellido Materno:	<input type="text"/>
Nombre Completo :	<input type="text"/>	Parentesco :	<input type="text" value="Progenitor (e)"/> ▼
Teléfono Domicilio:	<input type="text"/>	Teléfono Celular :	<input type="text"/>
Dirección :	<input type="text"/>		
Fecha - Aceptación (Responsable) :	<input type="text"/>	<input type="button" value="Guardar"/>	

Figura 3.29: Formulario de datos personales de una PCDV.

La figura 3.30 muestra el listado de las personas registradas por el sistema, los muestra en una tabla en la cual hay dos acciones: Ver, en la que puedo ver todo el formulario de registro de la PCDV e incluso modificarlo y la acción Eliminar, que elimina la información de la PCDV.

The screenshot shows a web browser window titled "Pantalla General - Microsoft Internet Explorer". The address bar shows "http://localhost/IBC/Inicio/pantallagral.asp". The page content includes a navigation menu on the left with categories like REGISTRO - PCDV, ESCUELAS, EVALUACIONES, REPORTE, and USUARIO. The main content area displays the date "Lunes 16 de junio" and the title "SISTEMA DE REGISTRO Y SEGUIMIENTO A NIVEL NACIONAL DE PERSONAS CON DISCAPACIDAD VISUAL". Below this, it says "Bienvenid@ : Rosybel Ibañez". The main section is titled "PERSONAS REGISTRADAS" and shows "Resultados : 1 a 5 de 73 encontrados." A table lists the first five records:

Num. C.I.	Nombre Completo	Fecha Nac.	Centro	Ingreso
7890447	AcostaReyesFaustina	4/10/1985	Manuela E. Gandarillas	3/1/2005
464123	AguileraCoaritaRómulo	6/7/1947	Santa Cecilia y Luis Braille	9/2/2007
6789041	Alarcón IllanesJohanDavid	26/4/1997	Manuela E. Gandarillas	25/2/2005
6883190	AlaroQuinoGerman	1/8/1984	Santa Cecilia y Luis Braille	23/2/2007
4924083	AndradeMolinaClaredWara	24/10/1991	Santa Cecilia y Luis Braille	14/3/2007

Below the table, there are pagination links: "01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 [Siguiente >>]". At the bottom of the main content area, there is a "SALIR" button.

Figura 3.30: Listado de PCDV registrados por el sistema.

3.3.2.3 Diagrama de componentes

El diagrama de componentes ayuda a complementar el diseño el comportamiento de los objetos. En la siguiente figura se muestra el diagrama de componentes del sistema en el nivel más general:

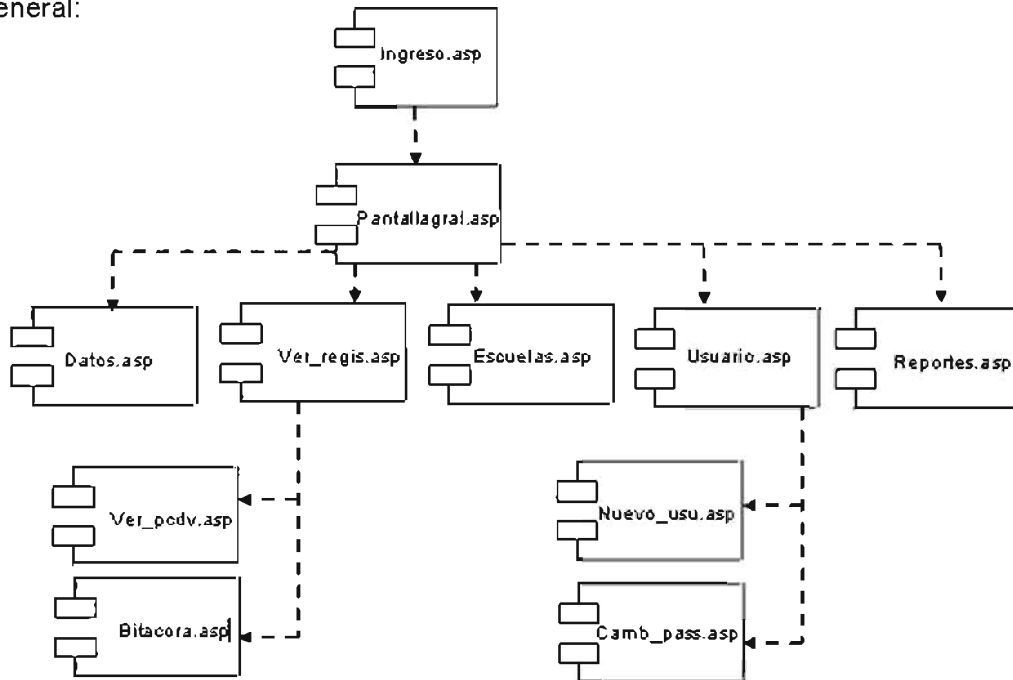


Figura 3.31: Diagrama de componentes general del sistema

3.3.2.4 Diagrama de despliegue

Un aspecto muy importante es ubicar todos los componentes del sistema en objetos físicos para su funcionamiento.

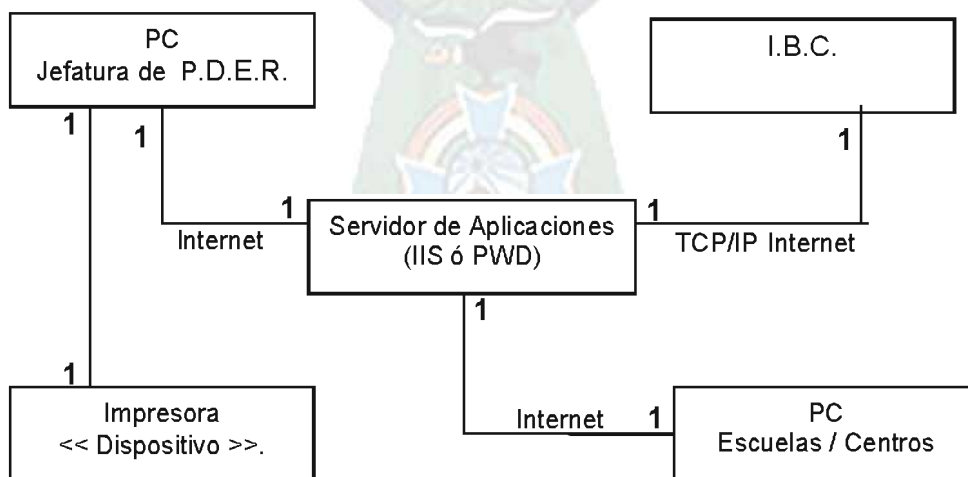


Figura 3.32: Diagrama de despliegue físico del sistema

3.4 Fase de Transferencia

El objetivo es traspasar el software desarrollado a la comunidad de usuarios. Una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).

Esta fase contempla la entrega del software a los usuarios. El proceso de desarrollo de software no termina con esta fase, porque durante esta fase el sistema puede necesitar mantenimiento para eliminar errores de programación.



4. CALIDAD Y COSTO-BENEFICIO DE SOFTWARE

4.1 Calidad de software

La medición de la calidad del software se lo realiza a través de métricas de control de calidad, para medir aspectos del software como ser: Funcionalidad, Instalación/Mantenibilidad y su Portabilidad, los cuales se detallan a continuación:

4.1.1 Funcionalidad

La funcionalidad de un sistema se mide según la complejidad del mismo, cuanto más complejo sea un sistema, es menos funcional y viceversa. Estas consideraciones se toman desde el punto de vista del usuario.

La funcionalidad de un sistema no puede ser medido directamente, entonces corresponde derivar mediante otras medidas directas como es el Punto Función, para esto se tiene la siguiente relación:

$$PF = Cuenta_Total \times ([0,65 + 0,01 \times \sum[F_i]])$$

Donde:

PF : Medida de funcionalidad entregada a la aplicación como valor de normalización.

Cuenta_Total: Es la suma de todas las entradas obtenidas.

$[0,65 + 0,01 \times \sum[F_i]]$: Ajuste a la complejidad según las medidas contables (directas) del dominio de Información del software, en donde $\sum[F_i]$ ($i = 1$ a 14) son los valores de ajuste de la complejidad.

Los valores de ajuste de complejidad se calculan en base a las respuestas del siguiente cuestionario:

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables? (5)
2. ¿Se requieren comunicaciones de datos? (5)
3. ¿Existen funciones de procesamientos distribuidos? (1)
4. ¿Es crítico el rendimiento? (3)
5. ¿Será ajustado el sistema en un entorno operativo existente y fuertemente utilizado? (5)
6. ¿Requiere el sistema la entrada de datos interactiva? (4)
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones? (2)

8. ¿Se actualiza los archivos maestros de forma interactiva? (3)
9. ¿Son complejas las entradas, las salidas, los archivos ó las peticiones? (0)
10. ¿Es complejo el procesamiento interno? (3)
11. ¿Se ha diseñado el código para ser reutilizable? (3)
12. ¿Están incluidas en el diseño la conversión y la instalación? (1)
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? (5)
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario? (5)

Calculando tenemos: $\sum[F_i] = 45$

Entonces:

$$PF = 370 \times ([0,65 + 0,01 \times 45])$$

$$PF = 370 \times 1.1$$

$$PF = 407$$

Si asignamos como valor máximo (5) a cada una de las preguntas, entonces $\sum[F_i]$ varia en el intervalo de $[0 - 70]$ y $[0,65 + 0,01 \times \sum[F_i]]$ varia en el intervalo de $[0.65 - 1.35]$. Relacionando el valor obtenido para el sistema y su valor máximo, que puede alcanzar PF , obtendremos la siguiente relación:

$$Funcionalidad = \frac{407}{463.05}$$

$$Funcionalidad = 0.88 \equiv 88 \% \text{ de funcionalidad}$$

Si el valor de la funcionalidad esta más cerca ó aproximado a **1**, significa que la funcionalidad del sistema es mejor.

4.1.2 Instalación / Mantenibilidad

El Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual, es una aplicación orientado para entorno a red y web, bajo la arquitectura cliente - servidor. Habiendo realizado un estudio previo, se tienen en cuenta los siguientes requerimientos:

- ✓ Un servidor de Base de Datos SQL-SERVER 2000 con una capacidad mínima de 533 Mhz de velocidad, 10 Gb de disco duro, 512 Mb de memoria RAM.

- ✓ Un servidor IIS (Internet Information Server) ó PWS (Personal Web Server), con una capacidad mínima de 533 Mhz de velocidad, 10 Gb de disco duro y 512 Mb de memoria RAM.
- ✓ Navegador para Internet (Internet Explorer ó Netscape Navigator)

La mantenibilidad del sistema esta asociado a la detección y corrección de fallas, como también a los cambios que son requeridos por el usuario. El tiempo de vida es indefinido, pero está sujeto a las necesidades del usuario, renovación, expansión, que desee realizar posteriormente.

La métrica que usaremos para medir la mantenibilidad es el Índice de madurez del software (IMS) el cual proporciona un indicador de la estabilidad del sistema. Se considera lo siguiente:

M_T = Número de módulos en la versión actual

F_c = Número de módulos en la versión actual que se han cambiado.

F_a = Número de módulos en la versión actual que se han añadido.

F_d = Número de módulos de la versión anterior que se han borrado en la versión actual.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = \frac{M_T - (F_a + F_c + F_d)}{M_T}$$

A medida que el IMS se aproxime al valor de 1, el producto se empieza a estabilizar.

Para el Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual, se han realizado unas pruebas a los prototipos, que fueron sujetos a cambios y modificaciones en sus diferentes versiones, esta relación se puede observar de la siguiente manera:

Tabla 3.14: Relación de cambios y modificaciones del SIRESE versión 1.0

Versión del sistema	M_T	F_c	F_a	F_d	IMS
VER. 1.0	4	2	0	0	$IMS = \frac{4 - (0+2+0)}{4}$ $IMS = 0.5$

Se observa que a medida que el software esta siendo probado, se identifican las diferencias y sus correcciones, alcanzando así una estabilidad deseable.

4.1.3 Portabilidad

Un sistema se considera portable, si el costo de transportar y adaptar un ambiente es menor al costo de rediseñar el sistema para el mismo ambiente.

La portabilidad en el caso para el sistema del presente proyecto presenta las siguientes características:

Hardware: Se considera:

CPU Pentium III de 1 Ghz de velocidad con 512 Mb de memoria RAM, como mínimo, puesto que la base de datos tanto como el servidor WEB se instalaran en una sola CPU. Para el entorno de Intranet, se necesitaría una red con cualquier topología y cableado estructurado.

Software: El código del sistema es portable a otros lenguajes de programación, como es el caso de PHP, esto se debe a que las características de funcionamiento de ASP y PHP son las mismas.

Datos: El motor de base de datos del sistema es portable y se puede transferir a otros sistemas de base de datos, como ser: Access y versiones superiores de Sql Server.

4.2 Análisis Costo – Beneficio del Sistema.

Para realizar el cálculo del costo relacionado al Sistema De Registro y Seguimiento a nivel Nacional de Personas con Discapacidad visual, se tomaran en cuenta los siguientes aspectos:

- ✓ Costo de análisis de programación
- ✓ Costos de capacitación del usuario
- ✓ Costo de instalación

4.2.1 Costo de análisis de programación

Utilizaremos el modelo con el nombre COCOMO, en la cual se tiene la siguiente ecuación:

$$E = a_b KLDC^{b_b}$$

Donde:

- ✓ E es el esfuerzo aplicado en personas – mes.
- ✓ KLDC es el número estimado de líneas de código distribuidas para el proyecto.
- ✓ El coeficiente a_b y el exponente b_b se observan a continuación:

Tabla 3.15: Relación de valores empleados en el modelo COCOMO.

Proyecto de Software	a_b	b_b	c_b	d_b
Orgánico	2,40	1,05	2,50	0,38
Semiacoplado	3,00	1,12	2,50	0,35
Empotrado	3,60	1,20	2,50	0,32

Reemplazando los datos en la ecuación se tiene:

$$E = 3.0(KLDC)^{1.12}$$

Las líneas de código en su totalidad se estima 3500 del cual el 35% es código reutilizable, entonces el total de LDC (líneas de código) es:

$$KLDC = (\text{total LDC} - \text{LDC Reutilizable})/1000$$

$$KLDC = 2,3$$

Reemplazando en la ecuación de determinación de esfuerzo se tiene:

$$E = 3.0 (2.3)^{1.12}$$

$$E = 7,6 \text{ personas} - \text{mes}$$

La estimación de duración esta dado por: $D = c_b E^{d_b}$

Utilizando la ecuación:

$$D = 2.5 (7,6)^{0.35}$$

$$D = 5,08 \text{ meses}$$

El costo de análisis y programación es de 400 \$us por analista – programador, dando un total de costo de 15443 \$us.

4.2.2 Costos de capacitación del usuario

La capacitación del usuario se realizará durante un tiempo estimado de 2 horas diarias con un costo de 2,5 \$us por hora, durante un tiempo de 10 días hábiles, lo que resulta un costo total de 50 \$us por la capacitación a los usuarios.

4.2.3 Costo de instalación

La instalación de los sistemas se realizará conjuntamente con el personal de la Jefatura de Planificación, desarrollo educativo y Rehabilitación del IBC, por lo cual no existe ningún costo.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Terminado el sistema se llegaron a las siguientes conclusiones:

- ✓ La implementación del Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual, logra integrar los módulos de registro y seguimiento académico, que mejora el control de la información de las PCDV.
- ✓ Durante el análisis de requerimientos se lograron especificar varios elementos que son importantes para la administración de la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C., que hasta ese entonces eran considerados relevantes, tales como: La dinámica Familiar, la capacitación en diversas áreas para las PCDV para mejorar su forma de vida y un seguimiento a la rehabilitación de las PCDV conjuntamente con las escuelas ó centros de rehabilitación.
- ✓ Se reduce los errores que se producían al llenar los formularios manualmente y los errores de transcripción de datos, gracias a la validación de datos del nuevo sistema.
- ✓ Se logro diseñar una base de datos que contiene toda la información de las PCDV durante y después de su rehabilitación.
- ✓ Se desarrollo y especificó el formato uniforme del formulario de registro para PCDV a nivel nacional, para estandarizar los datos personales que serán almacenados.
- ✓ Se aplico toda la metodología descrita en el desarrollo del presente sistema.
- ✓ Se estableció la funcionalidad del sistema en un 87%, lo que indica la utilidad del sistema.
- ✓ Los reportes son un apoyo para la toma de decisiones que colabora a las instancias superiores del I.B.C., a las Escuelas y Centros de Rehabilitación a Nivel Nacional.

5.2 Recomendaciones

El presente Sistema de Registro y Seguimiento a Nivel Nacional de Personas con Discapacidad Visual, para la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación, solamente fue instalado en dependencias de la Jefatura de planificación, desarrollo educativo y rehabilitación del IBC, e implementado solamente a nivel departamental en su primera etapa y que registra los datos a nivel nacional mediante transcripción de datos, enviados vía correo ó fax, se recomienda en un trabajo ulterior desarrollar un Sistema en Red a nivel nacional y mecanismos de seguridad para implementarlo así en todos los demás departamentos de nuestro país, en los que se cuente con Escuelas ó Centros de Rehabilitación para Personas con Discapacidad Visual.

Contemplar el desarrollo de un módulo de alerta temprana mediante estadísticas para el estudio y consideración de los factores que afectan el crecimiento de la población no vidente en nuestro país.

También se recomienda capacitar al personal que manipulara el sistema para que conozca y dé un funcionamiento correcto al sistema.

BIBLIOGRAFÍA

[DS, 26973] Decreto Supremo N° 26973, del 27 de marzo de 2003.

[KENDALL, 1997] KENDALL, 1997: *Análisis y Diseño de Sistemas*. 3ra Edición, Prentice-Hall Hispanoamérica S.A., México

[LARMAN, 1999] LARMAN C., 1999: *UML y Patrones - Introducción al análisis y diseño orientado a objetos*. Primera Edición, Prentice-Hall, México

[LOPE, 2003] Ley de Organización del Poder Ejecutivo N° 2446 del 19 de marzo de 2003.

[PRESSMAN, 2002] PRESSMAN, R., 2002: *Ingeniería de Software - Un enfoque práctico*. 5ta Edición, McGraw-Hill, España

[PRESSMAN, 2006] PRESSMAN, R., 2006: *Ingeniería de Software - Un enfoque práctico*. 6ta Edición, McGraw-Hill / Interamericana Editores, S.A. de C.V., México.

[RUMBAUGH, 1998] RUMBAUGH J., 1998: *Modelado y diseño orientado a Objetos*. Prentice Hall, México

[SENN, 1990] SENN J., 1990: *Análisis y Diseño de Sistemas de Información*. 2da Edición Prentice-Hall Hispanoamérica S.A., México

Proyectos de grado:

[VILA, 2004] VILA M. 2004: *Sistema de Registro y Control de Afiliación – Seguro Social Universitario*.

[CAÑIZARE, 2005] CAÑIZARE V., 2006: *Sistema de Seguimiento y Control de rehabilitación para el Régimen Penitenciario*.

Referencias Web:

[REF1] www.ibcbolivia.org – Instituto Boliviano de la Ceguera. [visitado: 04/04/2008]

[REF2] www.congreso.gov.bo/5biblioteca/Boletines/Fuentes%20N%C2%BA%2012A.pdf
Ley del 22 de Enero de 1957. [visitado: 22/10/2007]

[REF3] www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.htm. [visitado: 02/08/2007]

[REF4] http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado, Wikipedia:
Lenguaje Unificad de Modelado (UML). [visitado: 11/09/2007]

[REF5] http://www.pcm.gob.pe/porta_ongei/publicaciones/cultura/Lib5098/c02.HTM
Publicaciones Ongei - Perú: *Modelo Cliente – Servidor*. [visitado: 19/08/2007]

[REF6] http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion_EVirtual – Tutorial: *Esquema Cliente Servidor*. [visitado: 19/08/2007]

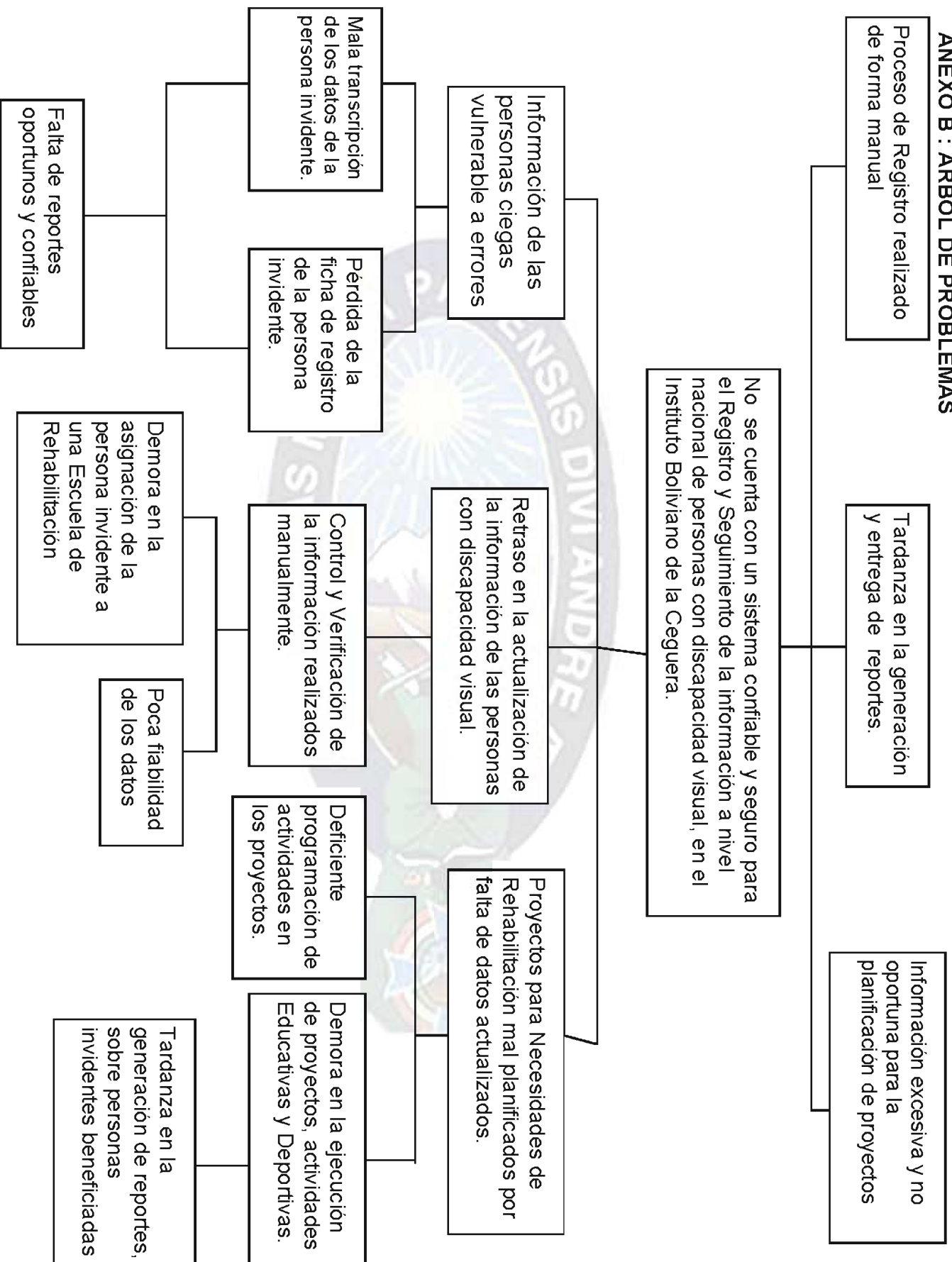
[REF7] http://web.frm.utn.edu.ar/comunicaciones/redes.html#2__UTN – Argentina:
Estructura Cliente – Servidor. [visitado: 19/08/2007]

[REF8] <http://www.lania.mx/biblioteca/newsletters/1993-invierno/ar2-ne6.html> - Boletín
Tecnológico #244: *Aplicaciones WEB*. [visitado: 24/11/2007]

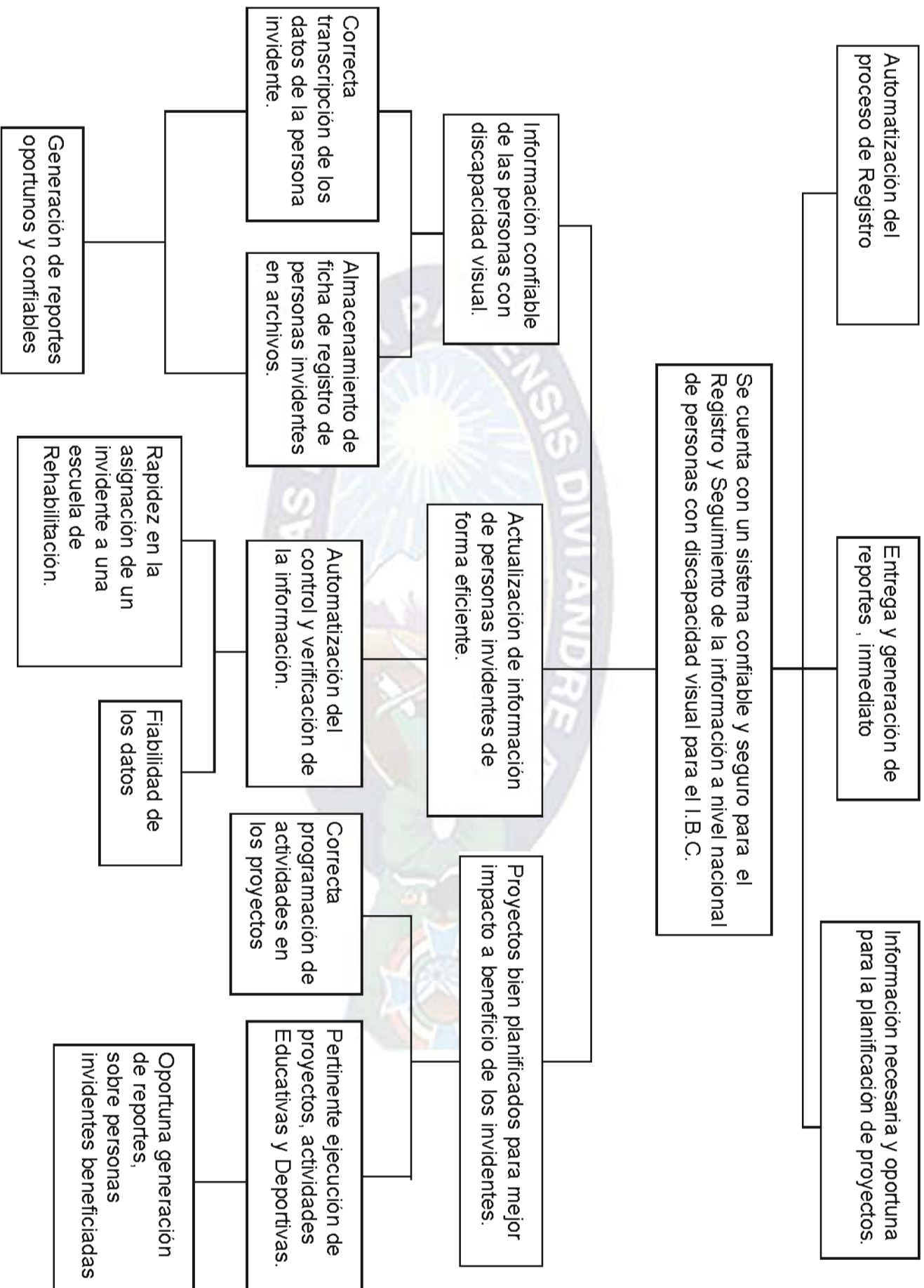
Resumen Narrativo	I.O.V.	Medio de Verificación	Supuestos
<p>Contribuir en el desarrollo del Sistema de Registro y Seguimiento de personas con discapacidad visual.</p>	<p>85 % menos de errores en la información de los afiliados y alumnos.</p>	<p>- Generación de reportes almacenados en archivo físico. - Generación de información sobre discapacitados visuales en archivo físico.</p>	
<p>Sistema de Registro y Seguimiento a Nivel Nacional de personas con discapacidad visual.</p>	<p>- Modelo del sistema de información de Registro concluido. - Modelo del sistema de seguimiento de la información de personas con discapacidad visual concluido.</p>	<p>- Sistema automatizado para el Registro y Seguimiento a nivel nacional de personas con discapacidad visual almacenado en la jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C. - Copia del código fuente en CD en la Jefatura.</p>	<p>La Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C. cuenta con el software y hardware necesarios.</p>
<p>Producto</p> <p>- Módulo de Registro de la información de personas con discapacidad visual. - Módulo de Seguimiento académico. - Módulo de generación de reportes.</p>	<p>- Prueba de caja blanca al módulo de registro de información. - Prueba de caja blanca al módulo de seguimiento académico. - Prueba de caja blanca al módulo de generación de reportes.</p>	<p>- Listado con resultados de las pruebas de caja blanca realizado al modulo de registro de la información. - Listado con resultados de las pruebas de caja blanca realizado al módulo de seguimiento académico. - Informe de verificación y documentación del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual, a la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C. - Copia de documentación del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual, en la biblioteca de la carrera de Informática de la Universidad Mayor de San Andrés.</p>	<p>Los equipos de computación con los que cuenta el I.B.C. cuentan con los soportes necesarios, para el buen funcionamiento del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual.</p>

Actividades				
<p>Analizar la situación actual de la Jefatura de Planificación, Desarrollo Educativo y Rehabilitación del I.B.C.</p>	<p>Observación de actividades y entrevistas con los encargados de la Jefatura.</p>	<p>Cuestionarios y Diagramas de Flujo de Datos anexados a la documentación del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual.</p>		
<p>Diseñar una propuesta del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual, para la institución.</p>	<p>Un analista de sistemas / 1 mes total 10000 Bs.</p>	<p>Recibo, por pago del trabajo realizado por el analista de sistemas.</p>	<p>Los gastos estimados por la elaboración del sistema no superan los 25000 Bs.</p>	
<p>Desarrollar e implementar el Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual, para la institución. Pruebas al sistema.</p>	<p>Dos programadores de sistemas cada uno 6000 Bs. En total 12000 Bs.</p>	<p>Recibo por el pago del trabajo realizado por los programadores.</p>	<p>La institución presta colaboración de manera constante para el desarrollo del sistema.</p>	
<p>Elaborar documentación con copias, del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual.</p>	<p>Compra de material de escritorio con un costo de 500 Bs.</p>	<p>Facturas por compra de material de escritorio.</p>		
<p>Elaborar el Manual de Usuario del Sistema.</p>	<p>Un instructor, 2 sesiones cada una de 3 horas, dependiendo del horario asignado por los responsables, con un total de 500 Bs.</p>	<p>Recibo, por pago del trabajo realizado por el instructor.</p>		
<p>Capacitar al Personal para el uso correcto del Sistema de Registro y Seguimiento a nivel nacional de personas con discapacidad visual.</p>				

ANEXO B : ÁRBOL DE PROBLEMAS



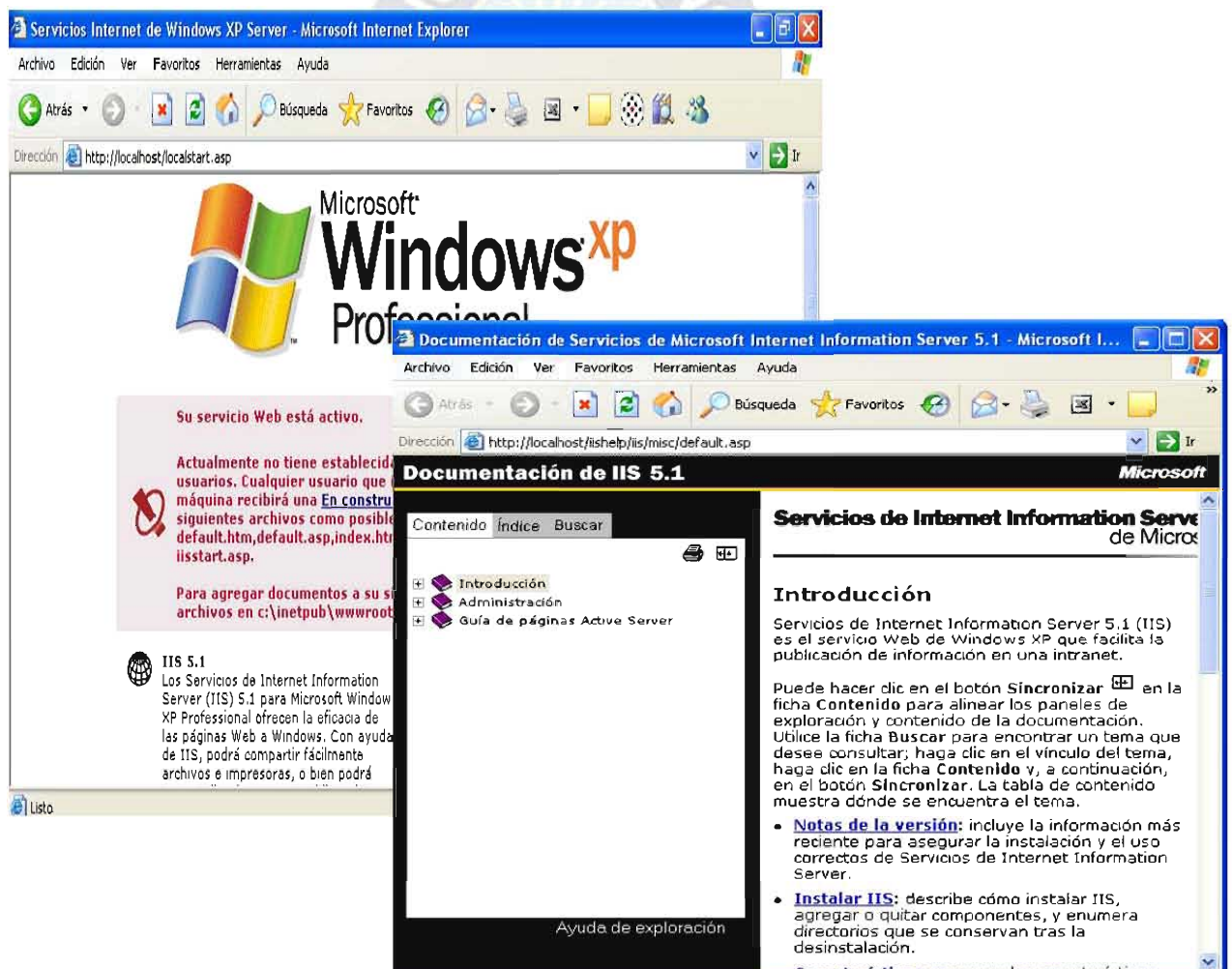
ANEXO C : ÁRBOL DE OBJETIVOS



ANEXO D: MANUAL DE INSTALACIÓN

Introducción.- El lenguaje ASP es un código de tratamiento de textos (scripts) que se ejecutan directamente en el servidor cuando son llamados, por ende no puede construirse un ejecutable con la extensión (.exe). Es por esta razón que la instalación del sistema debe realizarse con los siguientes pasos:

1ro. Tener instalado el IIS en el servidor, para verificar la conexión debe abrir la página del localhost: Abrir el Explorer y escribir en la barra de direcciones: ***http://localhost/***

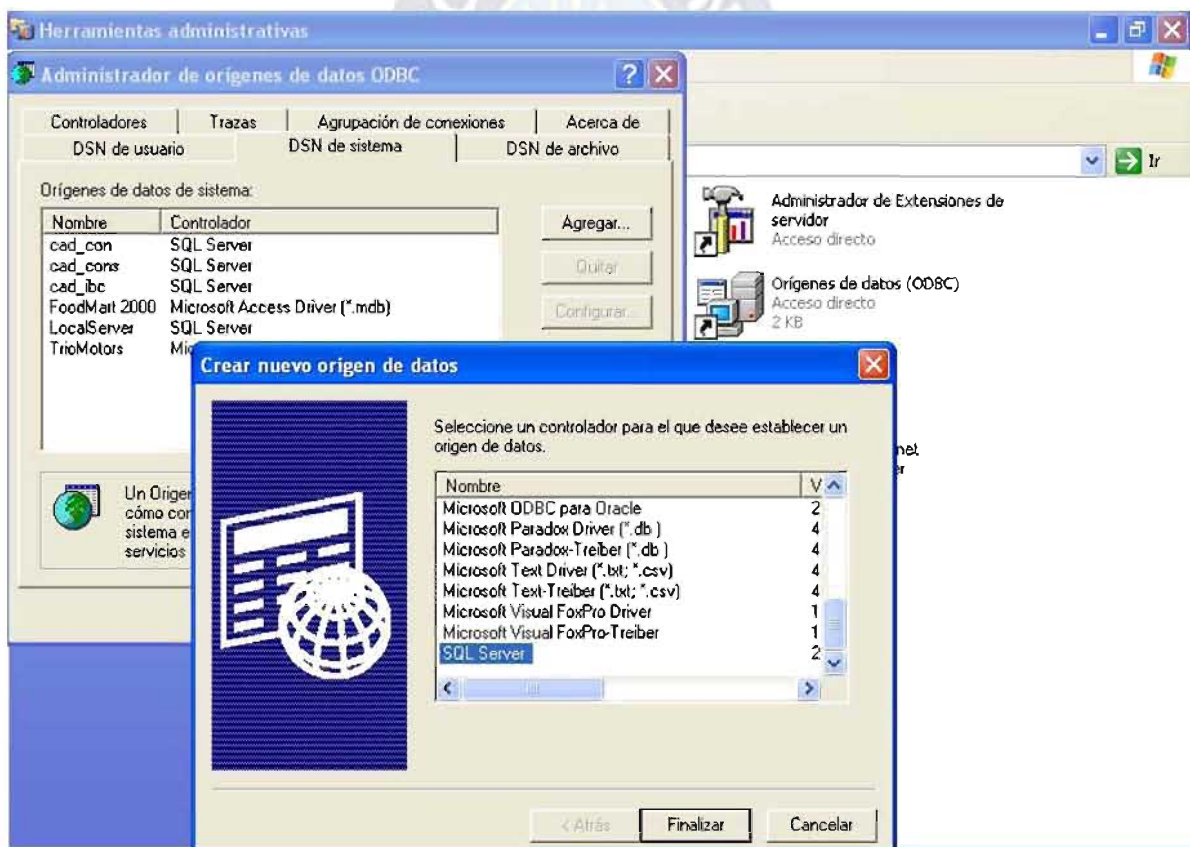


2do. Instalar el SQL-Server 2000 y crear una nueva base de datos llamada “registro” y copiar registro.Log y registro.Data dentro de la siguiente ubicación:

C:\Archivos de programa\Microsoft SQL Server\MSSQLData

3ro. Generar la cadena de conexión a la base de datos, haciendo:

- a) Panel de Control -> Herramientas administrativas -> Orígenes de Datos (ODBC).
- b) En la ventana de Administrador de Orígenes de datos ODBC, elegir la pestaña DSN de Sistema, escoger **Agregar**.
- c) En la ventana de Crear nuevo origen de datos, escoger **SQL Server**->Finalizar.
- d) Colocar Nombre: **cad_con** , Servidor: (Nombre del servidor de SQL) -> Siguiente
- e) Elegir con Autenticación de Windows -> Siguiente.
- f) Establecer la base de datos “registro” como predeterminada -> Siguiente.
- g) Escoger idioma Spanish -> Finalizar. Posteriormente Probar origen de datos.



4to. Posteriormente se debe copiar la carpeta “IBC” en la ubicación:

C:\inetpub\wwwroot

5to. Para acceder al sistema simplemente colocar en la barra de direcciones del Explorer la siguiente dirección:

http://localhost/IBC/Inicio/ingreso.asp