



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Sistema para el Centro Nacional de
Información y Documentación
DGADyR – CONADE

TESIS

para obtener el título de
INGENIERO EN COMPUTACIÓN

PRESENTAN:

ALTAMIRANO GARCÍA AARÓN IVÁN

LEÓN PALOMINO HUMBERTO

MEDINA CAMPOS RAÚL

DIRECTOR DE TESIS

Ing. Sebastián Poblano Ordoñez

Ciudad Universitaria, México, Octubre 2011





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Introducción	5
Capítulo 1 Antecedentes.....	9
1.1 Visión general.....	10
1.2 Objetivos.....	13
1.3 Propósito del sistema.....	13
1.4 El CEID en la Universidad Nacional Autónoma de México.....	15
Capítulo 2 Especificaciones y requerimientos del sistema	17
2.1 Identificación de los requerimientos.....	18
2.2 Análisis y especificación de los requerimientos.....	24
Módulo general.....	25
Módulos de usuarios de acceso controlado	26
Módulo de documentación.....	26
Módulo de informes.....	27
Módulo de comprobaciones	28
Especificaciones generales del sistema.....	30
A futuro.....	31
Capítulo 3 Fundamentos teóricos	33
3.1 Proceso de desarrollo de un sistema informático.....	34
Sistema.....	34
Ingeniería de software.....	35
3.2 Metodología de desarrollo.....	36
3.2.1 Ciclo de vida del software	36
3.2.2 Ciclo de vida en cascada	38
3.2.3 Ciclo de vida en V	40
3.2.4 Ciclo de vida por prototipos	41
3.2.5 Modelo DRA.....	42
3.2.6 Ciclo de vida en espiral	43
3.2.7 Ciclos de vida orientados a objetos.....	45
3.2.8 Herramientas CASE.....	46
3.3 Técnicas de implementación de sistemas	47
3.3.1 Metodología estructurada	48
3.3.2 Metodología orientada a objetos.....	49

3.3.3 Metodología OMT	50
3.3.4 Metodología UML.....	50
3.4 Metodología utilizada.....	52
Capítulo 4 Análisis y diseño	53
4.1 Análisis del sistema.....	54
4.2 El Diagrama de Flujo de Datos (DFD).....	54
4.2.1 Subproceso de Acceso al Sistema	58
4.2.2 Subproceso de Desbloqueo de Cuenta de Usuario	61
4.2.3 Subproceso de Registro de Actividades Realizadas por el Usuario	64
4.2.4 Subproceso de Envío de Mensajes	64
4.2.5 Subproceso de Evaluación Previa	65
4.2.6 Subproceso de Plan Anual	69
4.2.7 Subproceso de Recibo Fiscal	72
4.2.8 Subproceso de Comprobaciones	74
4.3 Diagrama Entidad-Relación (DER)	76
4.3.1 Sección de Control de Usuario	78
4.3.2 Sección de Documentación.....	79
4.3.3 Sección de Informes.....	80
4.3.4 Sección de Comprobación.....	82
4.4 Diccionario de Datos.....	83
4.5 Diseño de Módulos	92
4.5.1 Módulo de Control de Acceso	92
4.5.2 Módulo de Mensajería.....	92
4.5.3 Módulo de Registro de Actividades	92
4.5.4 Módulo de Informes.....	93
4.5.5 Módulo de Comprobaciones	93
Capítulo 5 Análisis de herramientas para la implementación del sistema.....	95
5.1 Leguajes de Desarrollo Web.....	96
5.1.1 Lenguaje HTML	97
5.1.2 Lenguaje Javascript	97
5.1.3 Lenguaje PHP	98
5.1.4 Lenguaje ASP	99

5.1.5 Lenguaje ASP.NET	100
5.1.6 JSP.....	101
5.1.7 Lenguaje Python	102
5.1.8 Lenguaje Ruby	103
5.2 Manejadores de Bases de Datos	103
5.2.1 DB2	107
5.2.2 MySQL.....	107
5.2.3 ORACLE.....	108
5.2.4 PostgreSQL	109
5.2.5 SQL Server.....	110
5.3 Propuestas de Soluciones	110
5.3.1 Comparativa de Bases de Datos.....	111
5.3.2 Comparativa de Lenguajes Web	115
5.3.3 Propuestas.....	116
5.4 Análisis	117
5.4.1 Comparativa de las propuestas.....	117
Capítulo 6 Implementación e implantación del sistema	119
6.1 Arquitectura cliente / servidor.....	120
6.1.1 Arquitectura multicapa	121
6.2 Modelos OSI y TCP/IP	122
6.3 Herramientas de Desarrollo.....	125
6.3.1 Microsoft SQL Server	126
6.3.2 Servidor IIS.....	126
6.3.3 Plataforma .NET	128
6.4 Arquitectura del sistema	129
6.5 Interfaz de usuario	130
6.6 Aplicación web en .NET	131
6.7 Base de Datos.....	132
6.8 Seguridad	133
Capítulo 7 Pruebas y mantenimiento del sistema	135
7.1 Pruebas y depuración del sistema.....	136
7.1.1 Detección y modificación de programación lógica	137

7.1.2 Realización de pruebas de funcionalidad de todos sus componentes, estimar tiempos de captura, transmisión y procesamiento de los datos	137
7.1.3 Pruebas de usabilidad	138
7.1.4 Identificar posibles mejoras de funcionalidad	138
7.2 Tipos de Mantenimiento	139
7.3 Revisiones periódicas	140
7.4 Respaldos de la base de datos del sistema	140
7.5 Cambios en la lógica del sistema	141
Conclusiones	143
Anexo. Manual de instalación del sistema	145
Requerimientos del sistema	145
Software	145
Instalaciones requeridas	145
Instalación los servicios de Microsoft Internet Information Server (IIS)	146
Para instalar IIS 7.0 en Windows server 2008	150
Hardware	150
Requisitos de espacio en disco duro para la instalación de SQL Server (32 y 64 bits)	151
Requerimientos de sistemas operativos de 32 bits	152
Requerimientos para sistemas operativos de 64 bits	152
Instalación del sistema	153
Requerimientos mínimos de información en la base	158
Glosario	163
Bibliografía	173
Referencias Web	174

Introducción

Los sistemas de información se han convertido en una parte muy importante de la vida moderna. La capacidad de clasificar enormes cantidades de datos y de producir rápidamente información para cualquier clase de usuario, desde el empleado que hace la nómina hasta el presidente de una nación los hace indispensables en una sociedad como la nuestra. Los sistemas de información siguen paso a paso las entradas y salidas de los recursos de todas las empresas, controlan los sistemas de reservaciones masivas de los medios de transporte, ejecutan millones de tareas indispensables para llevar un mejor y rápido control de información. En resumen, simplifican nuestra vida, automatizando tareas cotidianas y permitiendo que nos enfoquemos en asuntos que tendrán un mayor impacto en los procesos que realizamos.

La importancia y la necesidad de implementar sistemas de información en las empresas privadas y el sector público han ido en aumento debido a que éstos permiten administrar de manera eficiente las diferentes actividades realizadas dentro de una organización, propiciando así el crecimiento y consolidación exitosa de las mismas.

Las instituciones públicas son organizaciones en las cuales día a día se realizan un sinnúmero de operaciones de asignación de recursos, así como el pago y comprobación correcta, a lo largo del año fiscal. Estas constantes actividades son algunas de las muchas razones que exigen a los administradores de dichas dependencias a adquirir e implementar sistemas de información que permitan controlar la entrada y salida de recursos.

Dentro del gran abanico de instituciones públicas que hay en nuestro país podemos encontrar a la Comisión Nacional de Cultura Física y Deporte (CONADE), organismo que vela por la salud y el bienestar de millones de mexicanos ofreciendo servicios de la más alta calidad a todo tipo de agrupaciones deportivas a lo largo y ancho del país.

Los compromisos que la CONADE tiene en todo momento son poner al alcance del público servicios deportivos de excelente calidad y de resultados palpables, así como lograr una relación duradera a través de la excelencia en cuanto al servicio, la apertura hacia los deportistas y una constante comunicación con ellos.

Con el objetivo de lograr cada una de las metas anteriores, la CONADE examinó a fondo la necesidad de adquirir un sistema de información para elevar la calidad de sus servicios a través de una mejora continua en la gestión sistemática de su información.

Para cubrir y satisfacer sus necesidades es determinante llevar a cabo el proceso de desarrollo de *software* que requiere, actividad que conjuga un grupo de conceptos, una metodología y un lenguaje propio.

A este proceso también se le llama ciclo de vida del *software* que comprende cuatro grandes fases: concepción, elaboración, construcción y transición.

Los objetivos principales del sistema son:

- a) Homogeneizar los formatos que emplean las diferentes dependencias.
- b) Mantener la información disponible para aquellos que la requieran aprovechando la posibilidad de un repositorio virtual.
- c) Reducción de recursos físicos (papel, carpetas, cajas, etc.) para los procesos administrativos.
- d) Agilización de los procesos administrativos.
- e) Propiciar una mayor transparencia en lo referente a manejo de recursos.

Y para alcanzarlos los abordaremos en los siguientes capítulos de esta tesis:

En el capítulo 1 “Antecedentes” se menciona la relación que hay entre la CONADE y la UNAM y los propósitos generales del Sistema de Gestión Documental (SIGED).

En el capítulo 2 “Especificaciones y requerimientos del sistema” se describen las necesidades de los usuarios, los diferentes módulos del sistema, así como una proyección a futuro del mismo.

En el capítulo 3 “Fundamentos teóricos” se describen los conceptos básicos que son necesarios para el desarrollo del sistema. Se explica en forma general los lenguajes de programación, bases de datos, redes de computadoras, modelos de desarrollo de software y metodologías de Ingeniería de Software.

En el capítulo 4 “Análisis del sistema” describimos los diferentes módulos y subprocesos a través de los diagramas de flujo de datos y los diagramas de caso de uso que describen el funcionamiento del sistema y por último el diseño de la base de datos mediante su diagrama de entidad relación y el diccionario de datos.

En el capítulo 5 “Análisis de la tecnología para la implementación del sistema” trata sobre la arquitectura tecnológica que utilizamos para el desarrollo del sistema.

En el capítulo 6 “Desarrollo e implementación del sistema” trata de la construcción de la interfaz de usuario, del desarrollo y puntos importantes en cuanto a la base de datos, así como de la explicación breve de las herramientas de desarrollo utilizadas para su implementación.

En el capítulo 7 “Pruebas y mantenimiento del sistema” explica de manera general la forma en la que se llevaron a cabo pruebas al sistema para comprobar el buen funcionamiento del mismo y también se detallan algunos aspectos fundamentales para su mantenimiento.

Y las conclusiones a las que se llegaron después de la realización del sistema y algunas propuestas para su mejora, cerrando este volumen con los anexos citados en diferentes secciones de la tesis.

Capítulo 1

Antecedentes

Desde el 2001 la Comisión Nacional de Cultura Física y el Deporte (CONADE) se ha preocupado por integrar las nuevas tecnologías de comunicación y manejo de la información a los temas deportivos generados en el país por lo que, en colaboración con los Institutos Estatales del Deporte, ha fomentado la creación de los Centros Estatales de Información y Documentación de Cultura Física y el Deporte (CEID). Estos centros reciben recursos económicos federales, los cuales son canalizados a través del Centro Nacional de Información y Documentación de Cultura Física y el Deporte (CNID) residente dentro de la estructura interna de la CONADE y son ejercidos en la adquisición de equipo de cómputo, mantenimiento y administración.

Con todas esas herramientas resulta necesario contar con definiciones de las políticas gubernamentales en materia de infraestructura deportiva. Las cuales buscan llevar a la construcción de una base de datos donde se detallan cada uno de los elementos de la infraestructura pública y privada en todas las localidades del país mayores a 5,000 habitantes, quedando enmarcadas en la filosofía de la administración pública que privilegia la transparencia y la rendición de cuentas.

1.1 Visión general

El CEID es el lugar dentro del Instituto Estatal de Cultura Física y Deporte donde se localiza la infraestructura necesaria para hacer uso de las tecnologías de información. El CEID tiene la visión de proporcionar información oportuna y eficiente sobre las metodologías deportivas, así como dotar de una oportuna difusión de las actividades orquestadas por dicho instituto a través de los medios electrónicos. Paralelamente, el CEID realiza las funciones necesarias para facilitar la recopilación y sistematización de la información haciendo uso de las tecnologías informáticas y de comunicación, así como de los proyectos estratégicos de Gobierno Electrónico de la CONADE, entre los que destacan el Registro Nacional de Cultura Física y Deporte (RENADE) y el Censo Nacional de Infraestructura Deportiva. En resumen, el CEID es el lugar

donde se cubren las necesidades de información de los usuarios interesados en la cultura física y el deporte, así como de los programas de planeación, estadística, informática y de los registros necesarios para el desarrollo de actividades deportivas.

El CEID enfoca sus esfuerzos en 5 rubros:

- a) Apoyo a los programas de Gobierno Electrónico de la CONADE.
- b) Funciones operativas centradas en la realización de los siguientes informes:
 - 1. Evaluación previa.
 - 2. Plan anual.
 - 3. Convenio de colaboración.
 - 4. Comprobación de recursos asignados al programa.
 - 5. Informe técnico y de metas.
- c) Publicaciones electrónicas (diseño y mantenimiento del sitio web, así como la digitalización de documentos).
- d) Información del estado del deporte regional.
- e) Atención y servicios al público.

Para llevar a cabo estos fines el CEID debe contar con el espacio suficiente en un punto estratégico dentro del instituto del deporte estatal o equivalente, así como contar con elementos mínimos de recursos humanos e infraestructura.

La CONADE propone el siguiente esquema de personal dentro del CEID:

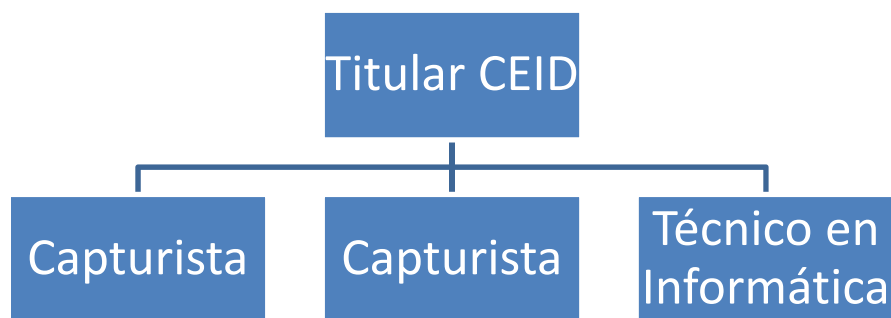


Figura 1.1 Personal mínimo sugerido para el CEID

En cuanto a la infraestructura se recomienda contar al menos con el siguiente equipamiento:

- a) Seis equipos de cómputo:
 - 1. Procesador a 2 GHz o superior.
 - 2. Memoria RAM de 1 GB.
 - 3. Disco duro de 80 GB.
 - 4. Quemador de CD y DVD.
- b) Impresora láser B/N con conexión de red.
- c) Impresora de color con conexión de red.
- d) Escáner tamaño carta.
- e) Cámara digital de 8 Mpx.
- f) Conexión a Internet de banda ancha.
- g) Línea telefónica propia.
- h) *Switch* con capacidad para distribuir la conexión de Internet a todos los equipos.
- i) Consumibles.

Por otro lado, el CNID, como enlace de los CEID con la CONADE, debe ocuparse en atender las siguientes funciones:

- a) Concentrar y proporcionar información sobre cultura física y deporte que se encuentre disponible a los deportistas y población en general.
- b) Integrar la información proveniente de los CEID a través de un sistema basado en Internet.
- c) Llevar el registro de la adecuada aplicación del presupuesto asignado a los CEID.
- d) Validar las compras de equipo de cómputo y/o consumibles de acuerdo a la normatividad federal.
- e) Informar del estado operativo que guarda el CEID.

De esta forma la mancuerna CEID - CNID trabaja usando las herramientas de cómputo que considera pertinentes para hacer difusión de las actividades realizadas en los estados y entidades participantes en

el programa, proveyendo de los recursos económicos necesarios para tales fines y llevando el control del ejercicio de estos recursos y actividades.

1.2 Objetivos

Uno de los propósitos del CEID es mantener actualizado el Registro Nacional de Cultura Física y Deporte (RENADE), que es el encargado de llevar el acervo de información integrado por la CONADE que proporciona el soporte en la toma de decisiones y financiamiento de aquellos programas necesarios para la coordinación, fomento, apoyo, difusión y desarrollo de la cultura física y deporte, manteniendo una estandarización de los procedimientos de registro a nivel nacional.

Los CEID deben mantener al día el RENADE en sus modalidades de trámite, autorización y registro para los organismos, asociaciones deportivas nacionales, sociedades deportivas, entes deportivos, convenios, eventos deportivos e infraestructura deportiva, constituyendo el acervo documental e histórico del desarrollo del deporte en la entidad.

El CEID facilitará el acceso de la información a la comunidad deportiva y población en general a través de los medios electrónicos y multimedia.

1.3 Propósito del sistema

Como se detalló anteriormente, el CEID mantiene con el CNID una constante comunicación mediante la entrega y cumplimiento de sus funciones operativas a través de los siguientes instrumentos:

- a) Evaluación previa.
- b) Plan anual.
- c) Convenio de colaboración.
- d) Comprobación de recursos asignados al programa.
- e) Informe técnico y de metas.

Los cuales son manejados a través de un documento en papel elaborado por el titular del CEID o, en su caso, por el personal encargado de las finanzas del Instituto del Deporte para ser firmado por el director del mismo antes de ser enviado al CNID para su evaluación y aprobación.

Este procedimiento aunque necesario para cumplir con las normas de operación y transparencia a las que se somete cualquier órgano de gobierno genera las siguientes problemáticas:

- a) Documentos con diferencia en los formatos de entrega.
- b) Retraso en la entrega de documentos.
- c) Entrega de informes y documentos en orden inadecuado.
- d) Incoherencias entre lo proyectado y lo reportado en determinados lapsos de tiempo.
- e) Incertidumbre de datos entre los actores involucrados del CEID y del CNID en la asignación y comprobación de recursos.
- f) Pérdida de información al existir cambios de administración en los institutos del deporte estatales.
- g) Generación de un exceso de documentación impresa que demanda infraestructura de almacenamiento voluminosa.
- h) Tiempo excesivo destinado al archivo de documentación.

El Sistema de Gestión Documental (SIGED) se plantea como una solución a toda la problemática descrita ofreciendo las siguientes características:

- a) Estandarización de formatos de informes.
- b) Repositorio de documentación en la web accesible a las personas autorizadas.
- c) Transparencia total para las autoridades del CNID y las del CEID en cuanto a la asignación presupuestal y la comprobación de recursos se refiere.
- d) Entrega de informes con estricta seriación.
- e) Digitalización de todos los procesos y funciones operativas.

La plataforma SIGED es un sistema que busca ordenar, estandarizar y mantener accesible toda la información relacionada con las funciones del CEID de manera digital.

1.4 El CEID en la Universidad Nacional Autónoma de México

La CONADE se ha preocupado por dotar a cada una de las 32 entidades federativas del país, en coordinación con los correspondientes institutos estatales del deporte, con un CEID que realice las funciones descritas con anterioridad a favor del deporte en la entidad. Adicionalmente a las entidades, en el país hay otros organismos con una fuerte presencia en diversas disciplinas deportivas y con una infraestructura deportiva considerable que merecen un trato distintivo por parte de la CONADE dotándolos de una serie de apoyos específicos y, por ende, de un CEID que favorezca su operatividad.

El Instituto Politécnico Nacional, el Instituto Mexicano del Seguro Social, el Consejo Nacional del Deporte de la Educación y la Universidad Nacional Autónoma de México son las entidades particulares en el país que cuentan con identidad propia frente a la CONADE.

A través de la Dirección General de Actividades Deportivas y Recreativas (DGADyR) la UNAM cuenta con un CEID que en representación de la entidad universitaria participa activamente en las actividades que marca el CNID como obligación.

Entre estas actividades se encuentra la participación en proyectos de mejora financiados por el CNID. Como parte de éstos se le solicitó al CEID de la UNAM su apoyo para la realización de un sistema de gestión documental a mediados de 2009.

A partir de ese año, el titular del CEID buscó conformar un equipo de desarrollo integrado por miembros de la comunidad universitaria, concretándolo en el año 2010. Fue entonces que comenzaron las reuniones de trabajo entre el personal del CNID y los desarrolladores

universitarios, comenzando con las labores de planeación y diseño que se describirán en los próximos capítulos.

Capítulo 2

Especificaciones y requerimientos del sistema

Como primer paso en el desarrollo de todo sistema, es primordial tener una aproximación con el usuario final con el propósito de definir claramente las expectativas que se tienen del sistema y el alcance que se le dará al proyecto.

Se debe tener un amplio acercamiento con el usuario para conocer la naturaleza de los procesos que se desean automatizar y virtualizar buscando las generalidades y particularidades del manejo físico de la documentación llevada por el CNID. La labor incluye seguir la generación y flujo de la documentación utilizada, contemplando tiempos de entrega y formatos de la información identificando a los responsables de darle cabida a cada paso hasta el punto en que los documentos han sido validados y almacenados de forma permanente.

Este capítulo habla de la interacción con el personal de CONADE donde explican su concepción del sistema acompañado de un diseño preliminar efectuado por ellos de la que sería la interface de usuario del CEID para después procesar la información y delimitar claramente los requerimientos.

2.1 Identificación de los requerimientos

En abril de 2010 el Departamento de Sistematización Documental del CNID encomendó al CEID de la UNAM, dependiente de la Dirección General de Actividades Deportivas y Recreativas (DGADyR), el diseño y construcción de una herramienta web destinada a digitalizar todo el proceso de seguimiento a los CEID. A partir de esa fecha se llevaron a cabo reuniones de manera frecuente y en ellas, el personal del CNID proporcionó bocetos de la interface de usuario deseable para el sistema.

La figura 2.1 corresponde a la pantalla inicial del sistema conceptualizado por el CNID, la cual debe respetar los lineamientos de una publicación del gobierno federal y adaptarse a los cambios que se pudieran dar. El menú contará con espacios abiertos al público en general y otros habilitados únicamente a usuarios registrados y

autenticados en el sistema, los cuales serán detallados en el siguiente apartado de este capítulo.

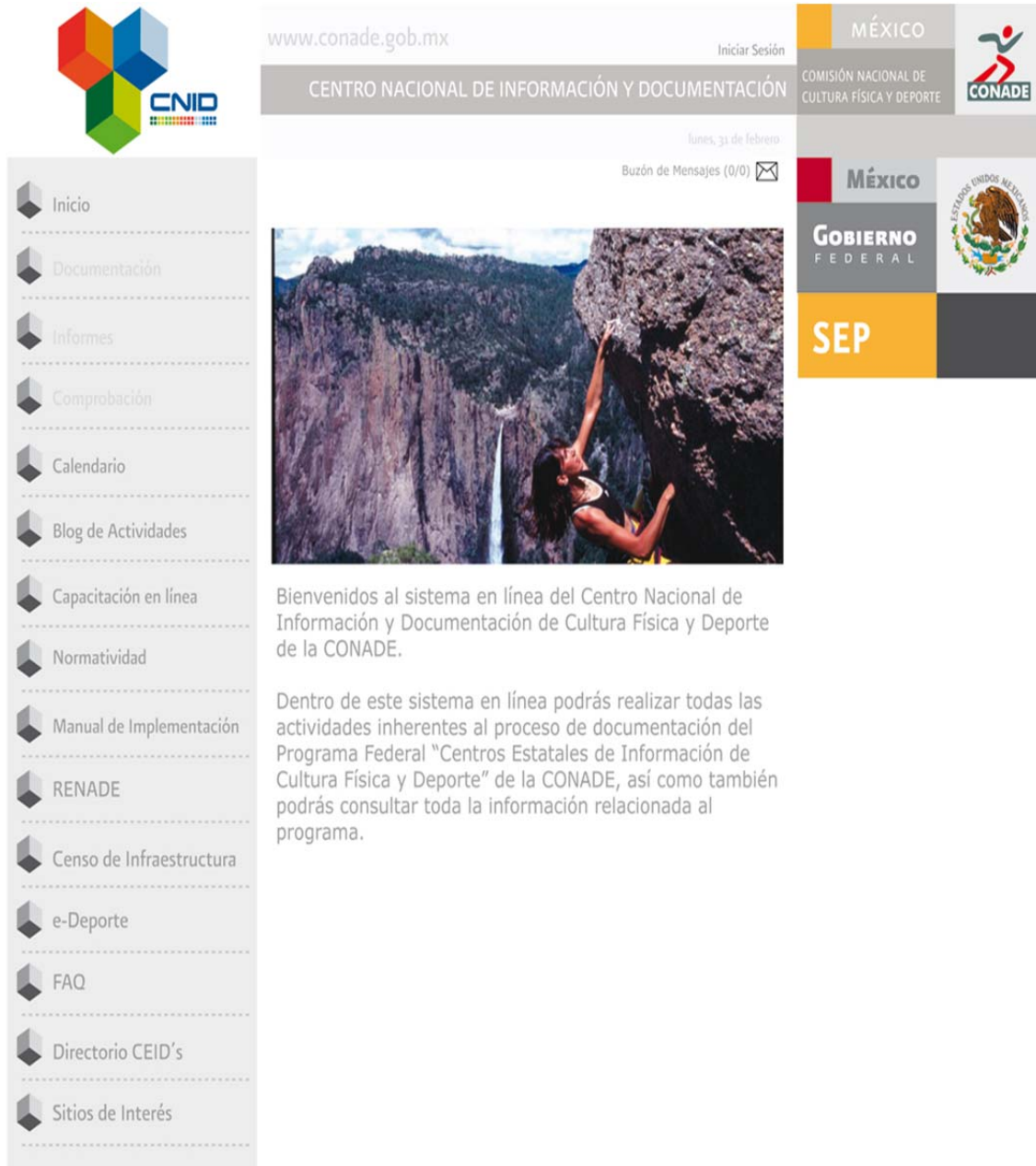


Figura 2.1 Interface propuesta por el personal del CNID

El sistema contará con diversas secciones, una de las cuales permitirá la carga y visualización de documentos electrónicos en formato PDF, como se observa en la figura 2.2, en la que se ejemplifica un módulo

repositorio de documentos legales que otorgan identidad jurídica a la relación entre los institutos estatales del deporte y la CONADE bajo la construcción de las figuras del CEID y del CNID, respectivamente.



Figura 2.2 Liga de visualización de documento PDF

Dentro de las posibilidades contempladas en el funcionamiento del sistema se considera la elaboración de informes específicos que deben cumplir con los tiempos establecidos en la norma oficial. Dichos informes están conformados por diversos elementos de entrada de datos, los cuales se pueden editar y evaluar de manera independiente.

En la figura 2.3 se muestra la estructura propuesta para el Plan Anual, uno de los diferentes informes que se contemplan para la operación y relación entre el CNID y los CEID.

www.conade.gob.mx Cerrar Sesión

MÉXICO COMISIÓN NACIONAL DE CULTURA FÍSICA Y DEPORTE CONADE

lunes, 31 de febrero Buzón de Mensajes (0/0)

Plan Anual

Año

*Favor de rellenar los espacios correspondientes

Para acceder a los apoyos del Programa "Centros Estatales de Información y Documentación de Cultura Física y Deporte" de acuerdo al inciso 4.4.1.1 de las Reglas de Operación del Programa Deporte los Institutos Estatales de Deporte a través de su CEID tendrán que presentar un programa anual de actividades que una vez aprobado por la CONADE, servirá como elemento de planeación, evaluación y seguimiento. (No aplica a Infraestructura Deportiva y proyectos especiales).

1.- Objetivos del CEID.

2.- Funciones y Servicios del CEID.

3.- Adquisiciones de equipo informático y de telecomunicaciones.

Equipo	Descripción	Costo

Siguiente

Centro Nacional de Información y Documentación de Cultura Física y Deporte, México
Algunos Derechos Reservados, CONADE 2008

Figura 2.3 Ejemplo de Plan Anual.

Al final de cada año los CEID tienen la obligación de comprobar el ejercicio de los recursos federales que les fueron asignados y la figura 2.4 muestra cómo, de manera categorizada, los centros deben otorgar los

2.2 Análisis y especificación de los requerimientos

El Sistema de Gestión Documental (SIGED) es una herramienta dinámica que tiene como finalidad el automatizar los procesos de evaluación, documentación, generación de informes y comprobación que ocurren entre el CNID y los CEID. Este sistema funcionará sobre una plataforma web y atenderá a los siguientes tipos de usuario:

- a) Usuario general: aquel que acceda al portal del CNID sin la necesidad de autenticación previa.
- b) Usuario controlado: se entiende por usuario de acceso controlado a aquel que requiere autenticación de acceso al sistema. Las cuentas de usuario serán gestionadas únicamente por el Administrador y ningún otro usuario podrá generar alguna cuenta. Se contará con los siguientes tipos de usuarios controlados:
 1. CNID/Administrador. Usuario administrador del sistema, tiene la obligación de autorizar los informes que generen los CEID, actualizar la documentación disponible en línea, así como otras funciones que se detallarán más adelante.
 2. CEID. Cada centro tendrá acceso al sistema a través de 3 diferentes cuentas con roles diferentes en función de su cargo:
 - i. Finanzas. Cuenta encargada de registrar las comprobaciones financieras, el recibo fiscal y ver el Plan Anual.
 - ii. Administrativo. Cuenta encargada de cargar el Plan Anual, la Evaluación Previa, así como los diversos informes a entregar a lo largo del año.
 - iii. Dirección. Esta cuenta es la encargada de dar el visto bueno a los informes y comprobaciones.

Los usuarios navegarán a través de módulos definidos como se describe a continuación:

Módulo general

Es aquel módulo que funcionará como entrada al sistema y que podrán usar los usuarios generales, con las siguientes posibilidades dentro del sistema:

- a) Página de inicio. Espacio donde se colocará un mensaje de bienvenida, una breve descripción del sitio y algún otro contenido estático que no tendrá ninguna relación con la base de datos.
- b) Capacitación en línea. Espacio destinado a publicar un catálogo de materiales generados por el CNID o los CEID, se construirá bajo un formato web estático con navegación por hipervínculos a los materiales (PDF, animaciones Flash, presentaciones Power Point, etc.).
- c) Normatividad. Liga a los documentos digitales que determinan los lineamientos de los centros.
- d) Manual de implementación. Liga al documento electrónico correspondiente.
- e) RENADE. Liga al sitio del RENADE.
- f) Censo de infraestructura. Liga al sitio correspondiente.
- g) e-Deporte. Liga a e-Deporte.
- h) Preguntas frecuentes. Espacio dedicado a las dudas más comunes relacionadas con el sistema.
- i) Directorio CEID. Espacio donde se desplegará una lista de los diversos centros, así como sus respectivos responsables e información de contacto.
- j) Sitios de interés. Catálogo de ligas a sitios relevantes a consideración de la CONADE.
- k) Web 2.0. Catálogo de las redes sociales particulares de cada centro o de cada entidad que sea de interés para el sistema.

Las opciones antes descritas serán de acceso a los usuarios generales a través de un menú colocado a la izquierda de la interface de usuario, acorde al formato proporcionado por el personal de diseño de la CONADE.

Desde este módulo existirá una liga de inicio de sesión, localizada en el sector superior derecho de la pantalla, que dará acceso a un formulario de autenticación que requerirá de un usuario y contraseña, datos que deberá cotejar con los respectivos registros en la base de datos, tras lo cual se determinará si concede el acceso, así como el rol que tendrá el usuario autenticado. El restablecimiento de contraseña lo deberá realizar el administrador del sistema, previa solicitud por correo electrónico u otra vía de comunicación por parte del afectado. Cabe señalar que no habrá ninguna función de registro de usuarios libre, ya que la administración de usuarios será responsabilidad y privilegio del administrador del sistema.

Módulos de usuarios de acceso controlado

Estos módulos representan la parte estructural del sistema y son aquellos en los que el carácter dinámico del sistema se manifiesta al responder a las necesidades de los usuarios controlados.

Este grupo de módulos se compone de 3 elementos:

- a) Documentación.
- b) Informes.
- c) Comprobación.

Los usuarios controlados pueden realizar diferentes operaciones en cada módulo en función del rol que desempeñan. A continuación se detalla cada uno de estos:

Módulo de documentación

Este módulo busca ser un repositorio de los siguientes documentos electrónicos para cada CEID:

- a) Anexo al Convenio.
- b) Anexo Técnico.

- c) Acta de Instalación.
- d) Informe Técnico.
- e) Recibo Fiscal.

Con excepción del Recibo Fiscal, el subir todos los documentos es responsabilidad del usuario CNID/Administrador y se destacan las siguientes particularidades:

- a) El Recibo Fiscal debe ser subido por cada uno de los CEID, este es requisito para liberar los recursos económicos asignados al ejercicio presupuestal, así como para habilitar el módulo de comprobaciones.
- b) El Anexo al Convenio y el Anexo Técnico son requisitos para hacer la asignación de recursos y habilitar el módulo de comprobaciones.
- c) Tras haber sido cargado el Anexo Técnico correspondiente a cada CEID se hará la asignación del apoyo al mismo, especificando monto asignado, la fecha de liberación del recurso y la fecha límite para la comprobación.

Todos los documentos cargados se encontrarán en formato PDF, siendo visibles para el usuario CNID/Administrador y para el CEID, permitiendo su consulta en cualquier momento. Se verificará un tamaño de archivo acorde al tamaño máximo esperado y estos archivos se cargarán en una carpeta del servidor, de tal forma que en la base de datos se almacenará solo la ruta.

Módulo de informes

Este módulo ofrecerá pantallas en función del informe a consultar, en las cuales los CEID tendrán la obligación de capturar sus documentos en los tiempos y fechas previamente establecidos en la normatividad del CNID. Los informes deberán ser cargados por la persona responsable dentro del CEID, y aprobados posteriormente por el CNID. A continuación se especifican los informes:

- a) Evaluación Previa. Este documento evalúa el estado actual del CEID en cuanto a infraestructura y personal disponible, así como el estado de los programas CNID - CEID. Este documento debe generarse una vez al año.
- b) Plan Anual. Para poder documentar este informe es requisito indispensable contar con la Evaluación Previa cargada en el sistema y aprobada por el CNID. Este informe es un requerimiento imprescindible para la liberación de recursos y la habilitación del módulo de comprobaciones, además describe los objetivos que busca alcanzar el CEID en el año y los nuevos recursos necesarios para lograrlos (humanos, electrónicos y de difusión). Este documento se entrega de manera anual.
- c) Informes Técnicos. Reporte solicitado al CEID en el cual se tendrá que enlistar todo el equipo de cómputo asignado al programa junto con sus especificaciones y uso dado en el Centro. De igual forma se tendrá que incluir la información relacionada al RENADE y al Censo Nacional de Infraestructura. El documento se entrega de manera trimestral.
- d) Informe de Metas y Registros. Reporte que contendrá los avances en el RENADE y el Censo Nacional de Infraestructura así como las rutas de los censadores y los viáticos devengados para este concepto. El documento se entrega de manera trimestral.

Para conocer los detalles en cuanto a estructura y formato específicos de cada informe se deben consultar los formatos establecidos por parte del CNID, así como el Manual de Administración de Centros Estatales de Información y Documentación de Cultura Física y Deporte.

Módulo de comprobaciones

Es el encargado de controlar el ejercicio de los recursos asignados por parte del CNID a los CEID, de acuerdo a la siguiente dinámica:

- a) El encargado de Finanzas del CEID accederá al sistema y tras ingresar al área de comprobaciones encontrará un listado de

categorías (a describir más adelante) en las cuales deberá enmarcar cada uno de los egresos del CEID, proporcionando, a su vez, factura digitalizada (en formato PDF).

Mientras carga los egresos el sistema hará una relación con los recursos asignados de la carga previa del presupuesto, permitiendo ver la cantidad de recursos pendientes a ejercer. Es factible hacer la comprobación de manera parcial, guardando los datos cargados a través de diversas sesiones.

- b) Una vez cargada la comprobación de gastos, ésta debe ser aprobada por el Director del CEID.
- c) La comprobación cargada y firmada ya es visible para el CNID, que se encarga de decidir entre aprobar o rechazar cada comprobación en función de los criterios establecidos por la institución. Se contará con un espacio de comentarios relacionados a las causas de rechazo de los rubros concernientes.
- d) Una vez aprobadas las comprobaciones, las facturas pasan a estar disponibles en un repositorio digital de consulta para el respectivo CEID y el CNID.

Los rubros de comprobación para los CEID son los siguientes:

- a) Computadoras.
- b) *Hardware*.
- c) Consumibles.
- d) Internet/Teléfono.
- e) Periféricos.
- f) Honorarios.
- g) *Software*.

Estos rubros serán cargados desde un catálogo dentro de la base de datos que permitirá su administración. Cada rubro solicitará cantidad, concepto, descripción y costo de cada elemento a comprobar y bajo esa misma estructura será el almacenamiento en la base de datos.

Es muy importante señalar que si la comprobación del año en curso no es realizada correctamente y justificando la totalidad de los recursos asignados, el sistema inhabilita a la entidad que presente irregularidades para la asignación de recursos en el próximo año, sin que esto la exima de la obligación de cumplir con el resto de los informes. En el caso que la entidad haya cubierto toda la documentación que debe, el sistema la rehabilitará.

Especificaciones generales del sistema

El sistema, sin importar el módulo al que se refiera, debe cumplir con los siguientes requerimientos:

- a) El sistema debe desarrollarse sobre el manejador de base de datos SQL Server 2008 y sobre ASP.NET, utilizando el lenguaje C# en virtud de que los sistemas de la CONADE están en un proceso de unificación hacia estas tecnologías.
- b) En cada carga o actualización de datos es necesario que el sistema registre el usuario que generó el evento, así como la fecha, dirección IP y navegador web utilizado.
- c) El sistema deberá contar con validación de cadenas de texto cargadas en los diferentes campos que eliminen la posibilidad de insertar sentencias ejecutables (*scripts* y código SQL).
- d) Los documentos PDF que se soliciten en los diversos módulos deben cargarse en una carpeta del servidor y en la base de datos solo se almacenará la ubicación de cada uno de estos archivos.
- e) El menú de navegación y las opciones disponibles para cada usuario son personalizables de acuerdo a los permisos que tengan asignados, entendiendo esto como la capacidad del sistema para habilitar o inhabilitar a cada usuario, en función de su rol en el sistema en las diversas opciones disponibles.

A futuro

El sistema tiene expectativas de crecimiento que implican la entrada de otras áreas de la CONADE que, aunque operan bajo un marco de procedimientos similares, cuentan con particularidades. En consideración a eso, el diseño del sistema debe contar con la flexibilidad de cargar nuevos rubros en cada módulo, así como algunas características de los mismos (conceptos que cuenten con la misma estructura). Para garantizar la escalabilidad se proporcionará la documentación y el código fuente del sistema.

Capítulo 3

Fundamentos teóricos

3.1 Proceso de desarrollo de un sistema informático

Sistema

Un sistema se define como un conjunto de elementos relacionados entre sí de alguna u otra manera para llegar a un fin común. Enfocándonos al área de sistemas informáticos, dichos elementos son el *hardware*, el *software* y los usuarios.

Aunque es muy común usar “sistema informático” y “sistema de información” como sinónimos son términos diferentes. Un sistema de información comprende elementos como personas, datos, técnicas y recursos materiales para la administración de los datos y su posterior aprovechamiento. En el sistema informático se ven involucrados equipos de cómputo que permitirán el almacenamiento, procesamiento y acceso a dicha información.

Los sistemas, aunque tengan diferentes objetivos, comparten elementos de los que dependerá su buen funcionamiento.

Uno de los agentes más importantes que forma parte de un sistema informático son las personas, ya que estas son las que ingresan los datos por medio de periféricos de entrada, dando pie a una serie de procesos y generando la información que puede ser consultada.

Otro de los elementos que integran los sistemas son los procedimientos, los cuales atienden a las actividades del usuario, procesando los datos y generando la información que posteriormente será consultada, limitando este acceso solo a las personas autorizadas.

Un tercer componente que podemos encontrar es el equipo, que hace referencia a cuestiones técnicas como los dispositivos de cómputo y en general, la infraestructura necesaria para que el sistema funcione de manera correcta.

Ingeniería de software

El *software* se puede definir como el conjunto de componentes lógicos (procedimientos, reglas, datos, etc.) que hacen funcionar y realizar tareas específicas a una computadora, pueden ir desde lo que es el sistema operativo hasta aplicaciones específicas como editores de texto, videojuegos, editores de imágenes y más.

En un principio, el software era desarrollado por quien tenía una necesidad específica y lo producía conforme a su experiencia o mejor dicho, de acuerdo a lo que su intuición le dictara, más que nada, la programación era un arte.

Para generar un producto de *software* se requieren de ciertos métodos y técnicas para que el desarrollo sea de calidad, de ahí que haya surgido la Ingeniería de software, rama de la ingeniería que cuida los aspectos mediante la aplicación de procesos previamente corroborados.

La Ingeniería de software se puede definir como el estudio de principios y metodologías para generar el conocimiento necesario para el buen diseño, desarrollo, operación y mantenimiento del *software*.

Algunos problemas que se llegaron a detectar en la producción de *software* incluyen:

- a) Retrasos considerables en la entrega.
- b) Poca productividad.
- c) Elevados costos de producción y cargas de mantenimiento.
- d) Baja calidad y fiabilidad del producto.
- e) Gran dificultad en el mantenimiento.

A lo anterior se le conoce como crisis del *software* y es mediante el desarrollo de ciertas metodologías y la implementación de nuevas herramientas con las que se pretende evitar problemas.

Lo que se busca con estos cambios es solucionar problemas de administración, calidad, productividad y sobre todo, facilidad de

mantenimiento, actividad que demanda una gran cantidad de recursos de toda índole (humanos, materiales y económicos).

Dentro de la Ingeniería de software se han desarrollado diferentes metodologías que ayudan a mejorar la producción, este proceso se denomina ciclo de desarrollo o ciclo de vida e incluye varias fases que van desde el diseño, pasando por la codificación y pruebas, hasta el mantenimiento del producto.

3.2 Metodología de desarrollo

3.2.1 Ciclo de vida del software

El ciclo de vida del software se utiliza para estructurar las actividades que se llevarán a cabo en el desarrollo de un producto.

En todo sistema se deben considerar tiempo y asignación de recursos para el desarrollo y posteriormente, para el mantenimiento y que el producto siga teniendo una vida útil.

Existen varios modelos de ciclo de vida, dependiendo de esto serán las etapas que lo componen y la forma en cómo se llevará a cabo la realización del proyecto.

Independientemente del modelo que se trate, hay etapas en común:

- a) Análisis. Estudio de los requerimientos para determinar su viabilidad.
- b) Diseño. Abstracción del sistema y elaboración de la interfaz de usuario.
- c) Codificación. Propiamente la elaboración del sistema como tal.
- d) Pruebas. Verificación del correcto funcionamiento del producto.
- e) Mantenimiento. Correcciones y/o modificaciones al sistema.

Análisis

Tanto el desarrollador como el usuario tienen un papel activo en la Ingeniería de software, es aquí, en esta interacción usuario - desarrollador donde nacen un conjunto de actividades llamadas análisis. El usuario intentará planear un sistema confuso a nivel de descripción de datos, funciones y comportamiento. El desarrollador debe actuar como interrogador, como un consultor que resuelve problemas y como negociador.

El análisis de requisitos permite al ingeniero de sistemas especificar las características operacionales del *software* (función, datos y rendimientos), indica la relación de la interfaz con otros elementos del sistema y establece las restricciones que debe cumplir el producto.

El análisis de requisitos del *software* se enfoca en cinco áreas de trabajo:

- a) Reconocimiento del problema.
- b) Evaluación del problema.
- c) Modelado.
- d) Especificación.
- e) Revisión.

Diseño

El diseño del *software* es realmente un proceso *multipasos* que se enfoca sobre cuatro atributos distintos del problema:

- a) Estructura de datos.
- b) Arquitectura del *software*.
- c) Representaciones de interfaz.
- d) Detalle procedimental (algoritmos).

El proceso de diseño traduce requisitos de una representación del *software* donde se pueda evaluar su calidad antes de que comience la codificación. Si el diseño se ejecuta de una manera detallada, la codificación puede realizarse mecánicamente.

Codificación

El diseño debe traducirse en una forma legible para la máquina, para este fin se utilizan lenguajes de programación que pueden ser procesados a diferentes niveles por los equipos de cómputo. La etapa de codificación puede variar en función de la tecnología en la que se va a implementar la solución de *software*.

Pruebas

Una vez que se ha generado el código, comienzan las pruebas del programa. Las cuales se enfocan en la lógica interna del *software*, asegurando que todas las sentencias se han probado, y sobre las funciones externas. Es decir, se realizan pruebas integrales para asegurar que la entrada definida producirá los resultados que realmente se requieren.

Mantenimiento

El *software*, indudablemente sufrirá cambios después de que se entregue al usuario final. Esto se deberá a diversas razones, por ello, el mantenimiento del *software* se aplica a cada uno de los pasos precedentes.

A continuación se describen algunos modelos de ciclo de vida:

3.2.2 Ciclo de vida en cascada

También se le conoce como modelo lineal secuencial. Es una técnica que se derivó de otras ramas de la ingeniería, tenía como objetivo el mejorar la calidad del *software* y reducir sus costos.

Es el modelo que más ha sido utilizado. La forma en cómo opera consiste en concluir cada etapa con un alto grado de exactitud y verificar si es posible continuar con la siguiente. Una vez que se ha llegado al periodo de mantenimiento es posible regresar a etapas anteriores para hacer cambios, haciendo nuevamente el recorrido hasta llegar al final.

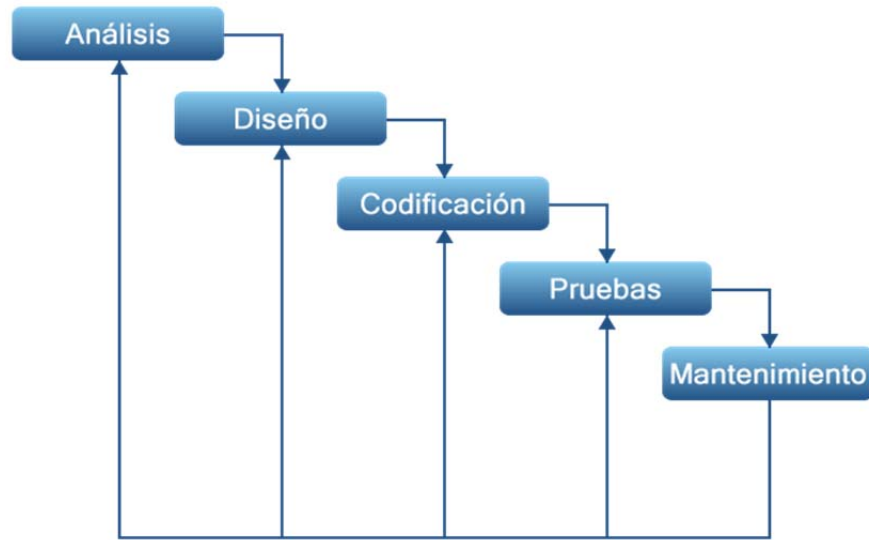


Figura 3.1 Ciclo de vida en cascada

Ventajas:

- a) Facilidad en la planificación.
- b) Producto resultante de alta calidad.

Desventajas:

- a) Disponibilidad de todos los requisitos del sistema.
- b) Resultados perceptibles hasta que el proyecto ya se encuentre en una etapa avanzada.
- c) Gran impacto de errores cuando el proyecto va avanzado.
- d) Repercusión en los tiempos de entrega cuando hay retrasos en etapas tempranas.
- e) Mayor costo del desarrollo.

Un ejemplo de aplicación de este modelo lo podemos encontrar en un proyecto de reingeniería, donde ya están establecidas las especificaciones.

3.2.3 Ciclo de vida en V

Está basado en el modelo en cascada y consta de las mismas fases, con la ventaja de que una fase también nos sirve para verificar o validar otras etapas.

En la figura 3.2 podemos ver que el análisis y diseño se encuentran del lado izquierdo y del derecho tenemos las pruebas y el mantenimiento y el vértice lo hace la fase de codificación.

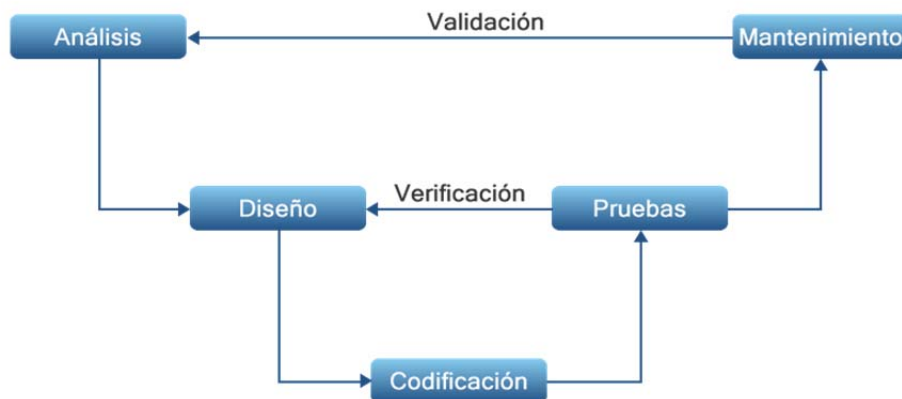


Figura 3.2 Ciclo de vida en "V"

Ventajas

- Facilidad en la detección de fallos gracias a la relación entre desarrollo y pruebas.
- El modelo se puede ajustar completamente a las necesidades del proyecto.
- Ofrece información detallada (instrucciones, recomendaciones, explicaciones, etc.) sobre cómo implementar cada etapa y el camino a seguir.

Desventajas

- Necesidad de todos los requerimientos por parte del usuario.
- Producto disponible hasta el final del ciclo de vida.
- Costo elevado en el periodo de pruebas.

3.2.4 Ciclo de vida por prototipos

Forma parte de los modelos de desarrollo evolutivo y es de gran ayuda cuando los requerimientos no están claramente establecidos, cuando no se sabe cómo implementar la interfaz de usuario o en algún otro caso que implique una gran cantidad de modificaciones y/o interacción con el usuario.

El prototipo nos ofrece un mejor medio para comunicarnos con el usuario y así poder identificar los requerimientos. Gracias a la retroalimentación se afinan detalles para empezar con el desarrollo formal del proyecto.

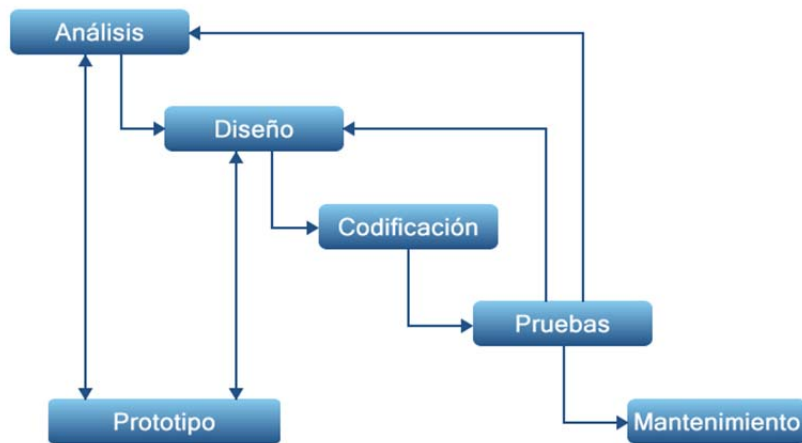


Figura 3.3 Ciclo de vida por prototipos

Ventajas

- No es necesaria la descripción completa de los requisitos.
- Se pueden realizar modificaciones en etapas tempranas del desarrollo.
- Entendimiento más claro por parte del usuario ya que el planteamiento no se encuentra solamente en la documentación.

Desventajas

- a) Se puede llegar a extender excesivamente en el tiempo de desarrollo del proyecto.
- b) Difícil administración si no se tiene un objetivo claro del *software* a desarrollar.
- c) El usuario e incluso los desarrolladores pueden considerar al prototipo como un sistema terminado.

3.2.5 Modelo DRA

El Desarrollo Rápido de Aplicaciones (DRA) es un modelo de proceso del desarrollo del *software* lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto, como se muestra en la siguiente figura:

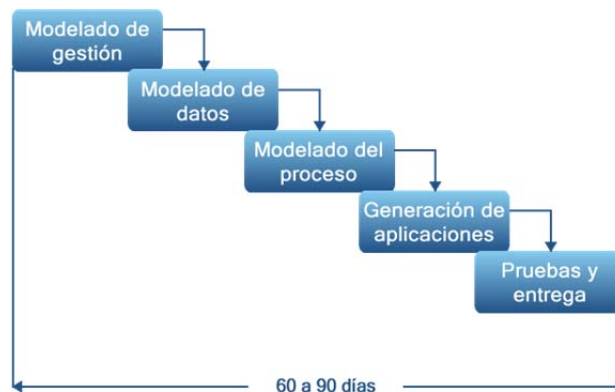


Figura 3.4 Modelo DRA

En el DRA se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el alcance del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo. Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

Modelado de gestión

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el

proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesó?

Modelado de datos

El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características, llamadas atributos, de cada uno de los objetos y las relaciones entre ellos.

Generación de aplicaciones

El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear *software* con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del *software*.

Pruebas de entrega

Como el proceso DRA enfatiza la reutilización ya se han comprobado muchos de los componentes de los programas, esto reduce tiempo de pruebas. Sin embargo, se deben probar los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

3.2.6 Ciclo de vida en espiral

El modelo en espiral consta de una serie de ciclos que se repiten dando como resultado un producto con mejoras respecto del anterior.

Este modelo es similar al de prototipo pero añade un componente denominado análisis de riesgos. Éstos abarcan aspectos como la mala interpretación de requerimientos, errores de diseño, problemas de implementación, etc.

El primer paso es definir los objetivos y posteriormente se realiza el análisis de los riesgos. Si después de haber hecho dicho análisis se

observa que hay incertidumbre sobre el problema, entonces se hace uso de un prototipo con el fin de entenderlo mejor y depurar los requerimientos proporcionados por el usuario.

Una vez terminada esta parte, el usuario evalúa los productos obtenidos y hace la retroalimentación para hacer las modificaciones necesarias y mejorar el desarrollo.

En cada vuelta de la espiral, la actividad de ingeniería se realiza mediante el ciclo de cascada o por medio del modelo de prototipos.

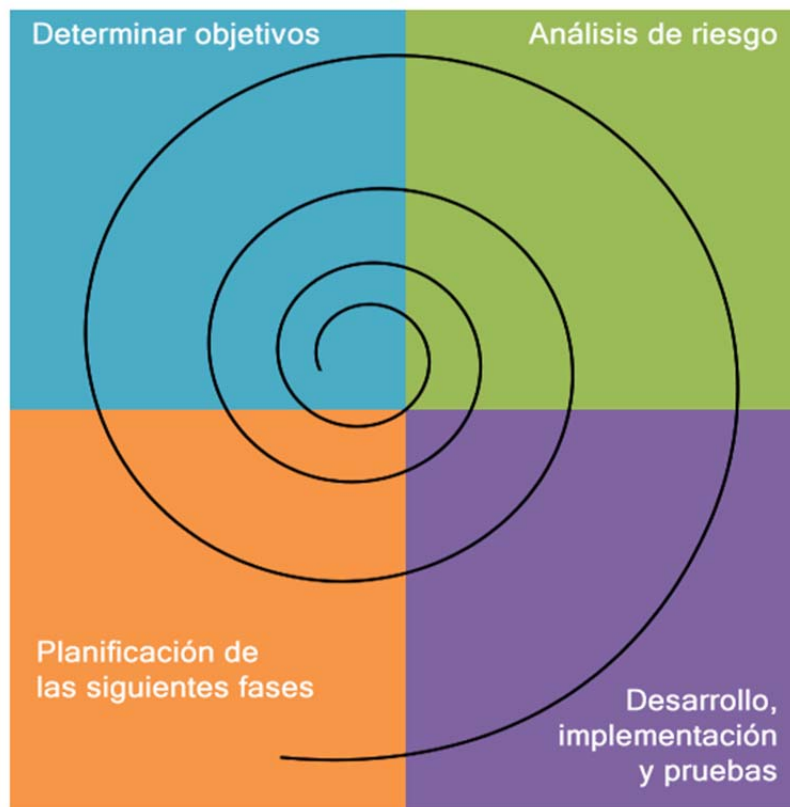


Figura 3.5 Ciclo de vida en espiral

Ventajas

- No se necesita una definición completa de los requerimientos para iniciar el proyecto.
- Validación de requerimientos desde etapas tempranas.

- c) Facilidad para detectar errores de análisis y diseño.

Desventajas

- a) Participación constante por parte del usuario.
- b) Mucho tiempo de desarrollo y alto costo.
- c) Requiere experiencia en la identificación de riesgos.

3.2.7 Ciclos de vida orientados a objetos

Los objetos tienen como característica que están basados en componentes, permitiendo tanto un alto grado de modularidad como la segmentación en proyectos que pueden llegar a trabajarse como independientes. Al aplicar un ciclo de vida orientado a objetos tenemos la ventaja de que el modelo es iterativo e incremental.

Modelo fuente

El modelo fuente es la forma más utilizada del ciclo de vida orientado a objetos, una de las características principales es que permite que cada clase se encuentre en diferente etapa, teniendo un ciclo de vida para cada una de ellas.

Contempla dos periodos para el desarrollo del proyecto: la etapa de crecimiento donde se construye el sistema y la etapa de madurez, donde se lleva a cabo el mantenimiento y cada mejora tiene un ciclo propio.



Figura 3.6 Ciclo de vida orientado a objetos

Ventajas

- a) Mayor productividad debido a que cada clase tiene un ciclo de vida propio.
- b) Reducción en el tiempo de desarrollo.

Desventajas

- a) Dificultad para aprender y emplear la metodología.
- b) Mantenimiento difícil.

3.2.8 Herramientas CASE

Las herramientas CASE (*Computer Aided Software Engineering* - Ingeniería de Software Asistida por Computadora) son un conjunto de utilidades, métodos y técnicas empleadas por los profesionales del desarrollo de *software* para facilitar y agilizar el ciclo de vida en un sistema, ya sea completamente o solo enfocándose en alguna de las etapas que lo componen. Es útil en tareas como el diseño del proyecto,

cálculo de costos, generación de código a partir del diseño, generación automática de documentación, etc.

Actualmente son varias las instituciones públicas que hacen uso de estas herramientas puesto que facilitan las tareas del ciclo de vida, como las etapas de codificación e implementación de pruebas, permitiendo un desarrollo más rápido de los sistemas, a su vez reducen los costos de producción, haciéndolas más productivas y ofreciendo una mayor calidad de servicio al público.

Algunas de las características y ventajas de las herramientas CASE sobre el desarrollo del *software* son:

- a) Permiten la comprobación de errores en etapas tempranas del desarrollo.
- b) Tienen soporte de reusabilidad.
- c) Mejoran la productividad en el desarrollo y mantenimiento.
- d) Aumentan la calidad del producto.
- e) Reducen el tiempo y el costo de desarrollo mediante la generación de código, documentación, automatización de pruebas, etc.
- f) Ayudan a tener una mejor planificación de un proyecto.
- g) Incrementan la biblioteca de conocimiento informático de una organización ayudando a la búsqueda de soluciones para los futuros desarrollos.
- h) Facilitan el uso de las distintas metodologías propias de la Ingeniería del *software*.

3.3 Técnicas de implementación de sistemas

Para el desarrollo de sistemas hacemos uso de alguna de las metodologías que se han tratado. Una metodología se puede definir como el conjunto de procedimientos necesarios para llegar a un objetivo. Enfocándonos en la parte de desarrollo de sistemas, la metodología es un modo sistemático de producir *software*, donde el proyecto se divide en

etapas y cada una de estas tiene tareas a realizar, además de tener bien definidas las entradas que recibe y las salidas que produce.

Lo que se busca mediante el uso de una metodología son cuatro puntos principales:

- a) Satisfacer las necesidades del usuario.
- b) Realizar cambios fácilmente una vez que el sistema esté en funcionamiento.
- c) Proporcionar una interfaz intuitiva y amigable para usar el sistema.
- d) Producir un sistema capaz de funcionar sin fallos y con la mínima cantidad de correcciones posibles.

Al igual que en el proceso de producción de un carro, o en la elaboración de un refresco, en las metodologías podemos encontrar distintos caminos dependiendo de la forma como se quieran satisfacer los requerimientos dados: la metodología estructurada y la metodología orientada a objetos.

3.3.1 Metodología estructurada

La metodología estructurada busca crear modelos de forma descendente, es decir, descompone el sistema de forma que se puedan tener partes lo más pequeñas posibles. La metodología estructurada se divide a su vez en orientada a procesos y orientada a datos.

Existen varias herramientas que se utilizan en la metodología orientada a procesos, entre las que encontramos:

- a) Diagramas de Flujo de Datos (DFD): representan la forma en la que los datos se mueven y se transforman e incluye:
 - 1. Procesos
 - 2. Flujos de datos
 - 3. Almacenamiento de datos

Los procesos individuales se pueden a su vez descomponer en otros DFD de nivel superior.

- b) Especificaciones de procesos: es una herramienta que nos define las acciones que lleva a cabo una función en el sistema, es decir, lo que hay en medio de las entradas y las salidas.
- c) Diccionario de datos: es el conjunto de datos que incluye las características lógicas (si es una cadena, entero, decimal, la longitud, rango de valores, si debe o no ser tener un valor, etc.) para ser utilizados en un sistema.
- d) Diagrama de Transición de Estados (DTE): son herramientas que modelan procesos que dependen del tiempo, se basan en el estado que tiene el sistema y en los cambios de estado.
- e) Diagramas Entidad - Relación (DER): nos ayuda al modelado de datos mostrando las entidades principales, sus propiedades y la interrelación que hay entre ellas.

3.3.2 Metodología orientada a objetos

Es una metodología de más reciente creación que está enfocada al uso de componentes, teniendo la ventaja de reutilizar código y de que su mantenimiento sea más fácil debido a la mejor localización de los cambios.

El paradigma de orientación a objetos se centra en la entidad principal (objeto), la cual contiene los datos y la forma en cómo estos datos pueden ser manipulados.

El objeto es una abstracción de los entes del mundo real y se puede utilizar por medio de la instanciación de una clase, que no es otra cosa más que la esencia del objeto. En otras palabras, la clase nos define las características del objeto.

Algunos de los rasgos más característicos de la orientación a objetos son:

- a) Herencia: como si se tratase de una familia, las clases también pueden heredar métodos y atributos de una clase padre.

- b) Polimorfismo: es la capacidad de nombrar a un método (que puede hacer cosas diferentes) de la misma forma para objetos de diferente clase.
- c) Encapsulamiento: se le llama así a la propiedad de los objetos que hace que aquellas características no necesarias para interactuar con otros no sean visibles.

3.3.3 Metodología OMT

La metodología OMT (*Object Modeling Technique* - Técnica de Modelado de Objetos) es una práctica que nos sirve para el diseño y modelado de un sistema de *software*.

Algunas de sus características principales es que considera la comunicación con el usuario, muestra la información de una forma diferente y disminuye la complejidad del problema mediante la segmentación de procesos.

La etapa de diseño se descompone en:

- a) Modelo de objeto. Representa los artefactos del sistema, es decir las clases y asociaciones con atributos y operaciones.
- b) Modelo dinámico. Constituye las interacciones entre los componentes ya mencionados, estas interacciones son eventos, estados y transiciones.
- c) Modelo funcional. Hace referencia a los métodos del sistema desde el punto de vista de flujo de datos.

Esta metodología pone especial interés en las etapas de análisis y diseño, siendo la precursora de UML.

3.3.4 Metodología UML

UML (*Unified Modeling Language* - Lenguaje de Modelado Unificado) es un lenguaje de modelado de sistemas de software muy utilizado actualmente y aprobado en el 2005 por la ISO como un estándar.

Este lenguaje busca hacer una descripción completa del modelo y los artefactos del sistema, contempla formas de representación para procesos de negocios, funciones del sistema, expresiones del lenguaje de programación, esquemas de bases de datos, etc.

Existen 2 versiones de UML, la que podemos encontrar actualmente es la 2.3 y cuenta con 13 tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas y se agrupan en diagramas de estructura, comportamiento e interacción.

Los diagramas de estructura enfatizan en los elementos que deben existir en el sistema modelado:

- a) Diagrama de clases.
- b) Diagrama de componentes.
- c) Diagrama de objetos.
- d) Diagrama de estructura compuesta.
- e) Diagrama de despliegue.
- f) Diagrama de paquetes.

Los diagramas de comportamiento hacen hincapié en lo que debe suceder en el sistema modelado:

- a) Diagrama de actividades.
- b) Diagrama de casos de uso.
- c) Diagrama de estados.

Los diagramas de interacción son un subtipo de diagramas de comportamiento, que prestan especial importancia sobre el flujo de control y de datos entre los elementos del sistema modelado:

- a) Diagrama de secuencia.
- b) Diagrama de comunicación.
- c) Diagrama de tiempos.
- d) Diagrama global de interacciones.

3.4 Metodología utilizada

Después de haber estudiado y analizado los diferentes ciclos de vida que hay y considerando la forma en cómo se nos presenta la interactividad con el usuario creemos que el modelo que mejor se adapta al desarrollo del sistema SIGED es el ciclo de vida en espiral dado que contempla que el sistema siga en funcionamiento por un largo plazo. En cada iteración, como lo define el modelo, se buscará mejorar la calidad, el funcionamiento y el rendimiento del producto.

Aunado a esto, el ciclo del vida nos ayudará a integrar los módulos necesarios para dotar de una mayor funcionalidad al sistema y así poder responder a las necesidades de otras áreas de la CONADE.

Capítulo 4

Análisis y diseño

El sistema SIGED está dividido en diferentes módulos, cada uno de estos cuenta con diferentes procesos que describen el flujo de datos.

En este capítulo se describe el análisis y el diseño que conforman el proyecto SIGED. Con base en este diseño se programaron los módulos funcionales del sistema.

4.1 Análisis del sistema

Para analizar adecuadamente los requerimientos del sistema es necesario tener una visión global de su funcionamiento.

Primero se enumeran de manera general los requerimientos:

- a) El sistema debe funcionar a través de Internet.
- b) Control de acceso mediante usuario y contraseña.
- c) Los usuarios del sistema deben tener diferentes roles.
- d) Los informes y comprobaciones deben seguir un flujo de aprobación.
- e) Llevar un registro de las actividades de los usuarios.
- f) Enviar mensajes informativos a los usuarios.

Una vez analizados e identificados los requerimientos del sistema, se continuó con la etapa de diseño.

4.2 El Diagrama de Flujo de Datos (DFD)

Los datos fluyen a través del sistema en diferentes subprocesos que describen el comportamiento del mismo. A continuación se detallan los subprocesos y los diagramas de flujo de datos que forman el sistema.

Para llevar a cabo los diagramas que reflejan el funcionamiento del sistema fue necesario, primero, realizar el análisis e identificación de los requerimientos del mismo. Ambas partes, dentro de todo el proceso, funcionan como un pilar principal del cual se parte para realizar las

etapas posteriores de forma óptima, por ello fue primordial hacer hincapié en los detalles de cada una de ellas.

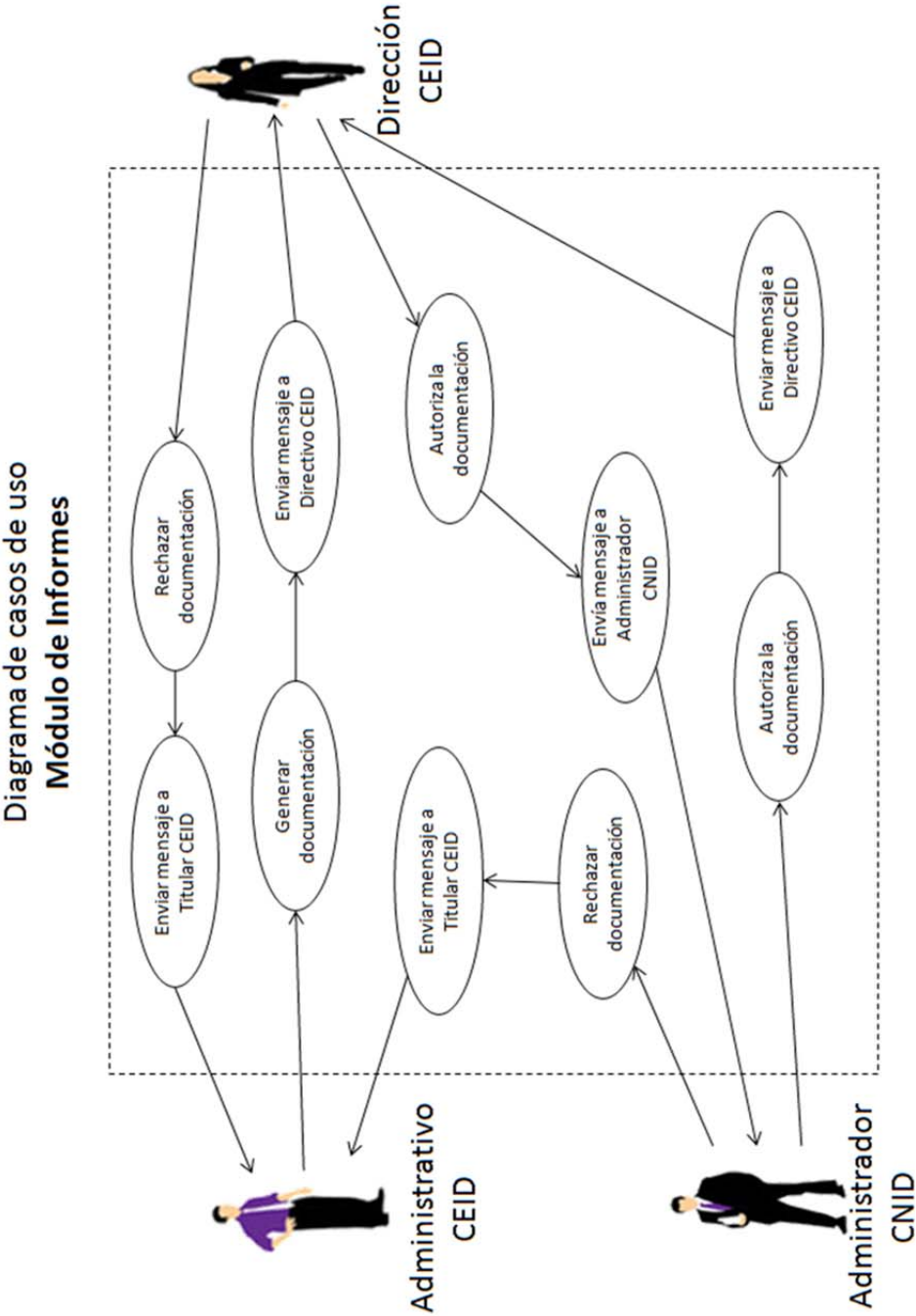


Figura 4.1 Diagrama de caso de uso para el Módulo de Informes

Diagrama de casos de uso Módulo de Documentación

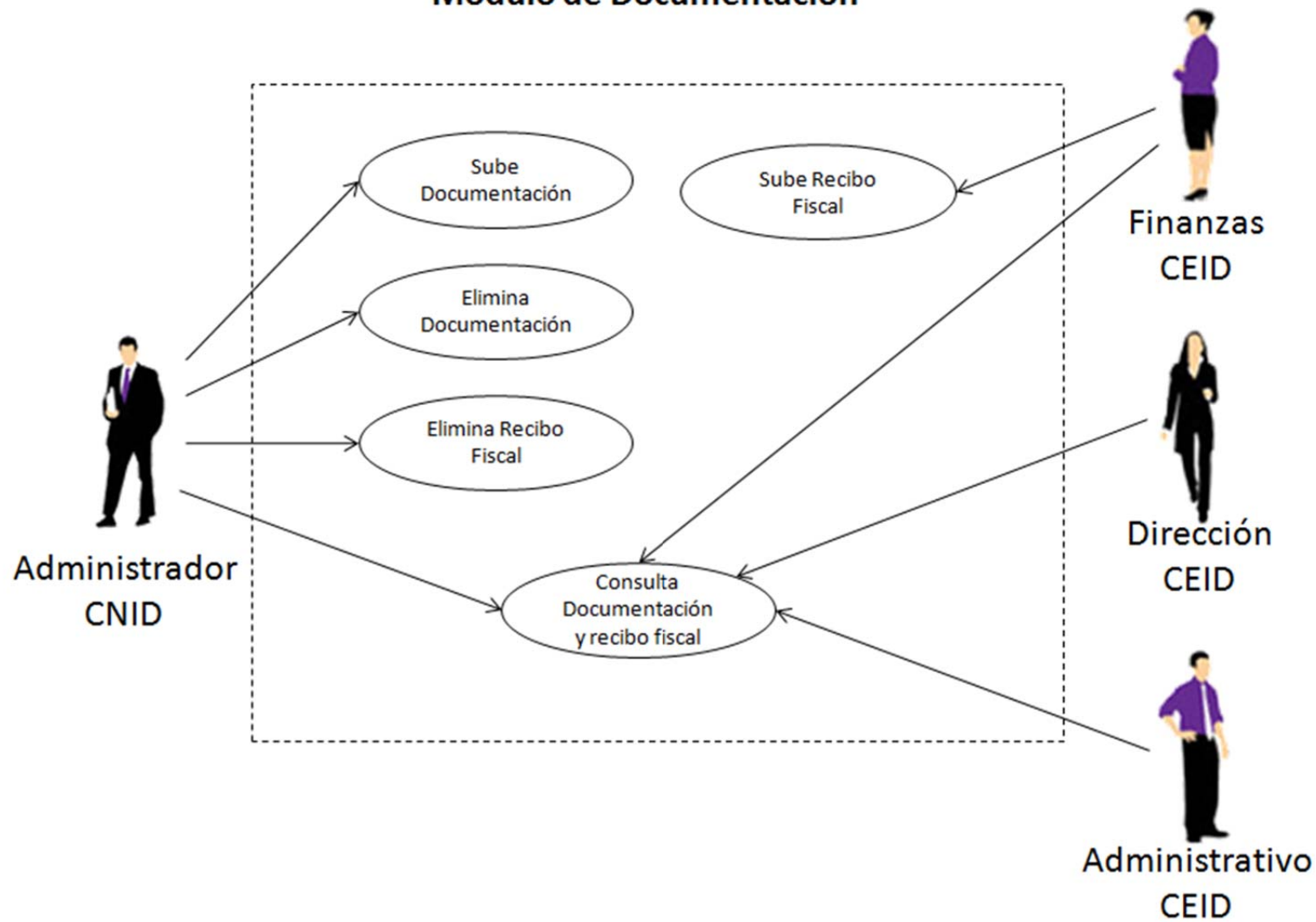


Figura 4.2 Diagrama de caso de uso para el Módulo de Documentación.

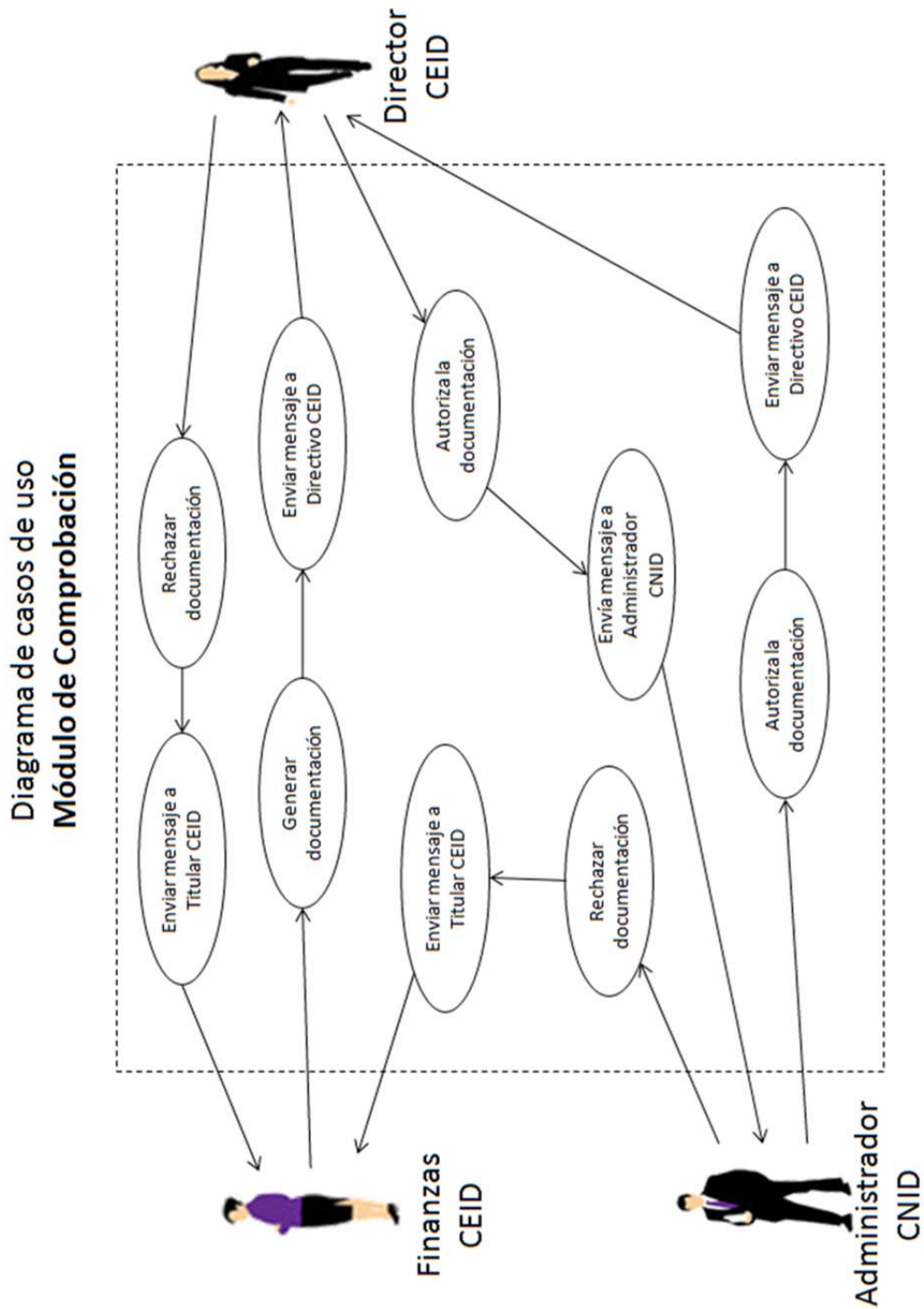


Figura 4.3 Diagrama de casos de uso para el Módulo de Comprobación.

A partir de los correspondientes diagramas de flujo se describieron cada uno de los escenarios que indican a detalle las características del caso de uso en cuestión. Básicamente, se consigue describir con cada diagrama,

no solamente el funcionamiento, sino también los roles que intervienen en ellos.

4.2.1 Subproceso de Acceso al Sistema

En este proceso se describen los pasos para que un usuario registrado tenga acceso al sistema. Los roles que intervienen son los siguientes:

- a) Usuario registrado.- Es el personal que cuenta con un usuario y contraseña para ingresar al sistema.
- b) Sistema.- Módulo de Administración de Usuarios dentro del sistema SIGED.

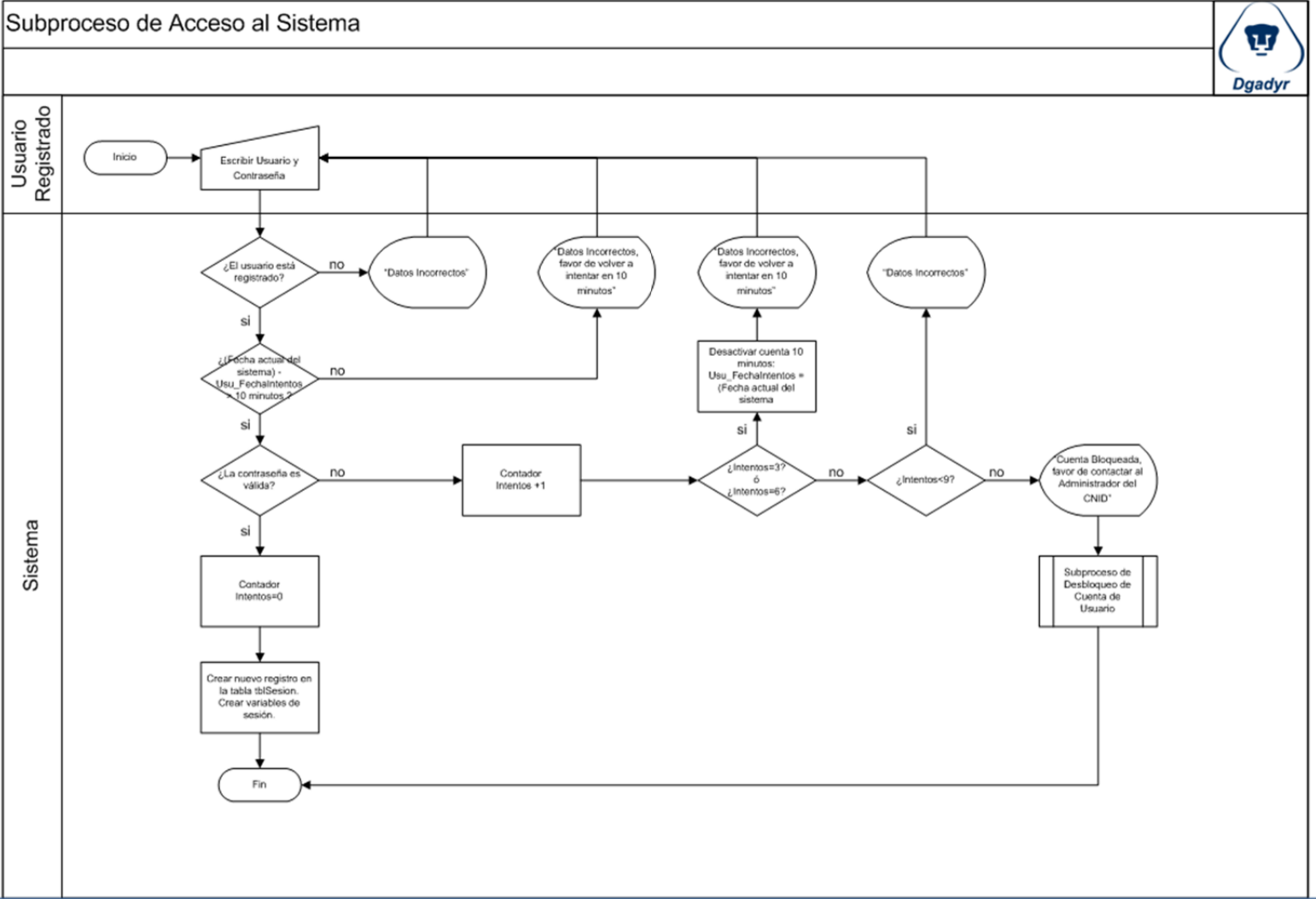


Figura 4.4 Diagrama del Subproceso de Acceso al Sistema

La tabla 4.1 nos describe, paso a paso, el flujo ilustrado en la figura 4.4 que determina la forma de acceder al sistema y lo que ocurrirá en casos donde no se cumplan las condiciones necesarias.

Paso	Rol	Descripción
1	Usuario registrado	Escribir usuario y contraseña. Ir al paso 2.
2	Sistema	¿El usuario está registrado? Sí: Ir al paso 4. No: Ir al paso 3.
3	Sistema	Mostrar mensaje: "Datos incorrectos. Usuario no registrado." Ir al paso 1.
4	Sistema	¿Han pasado menos de 10 minutos desde el último intento de acceso? Sí: Ir al paso 6. No: Ir al paso 5.
5	Sistema	Mostrar Mensaje: "Datos incorrectos, favor de volver a intentar en 10 minutos". Ir al paso 1.
6	Sistema	¿La contraseña es válida? Sí: Ir al paso 14. No: Ir al paso 7.
7	Sistema	Contador Intentos=Intentos+1 Ir al paso 8.
8	Sistema	¿Intentos = 3? o ¿Intentos = 6? Sí: Ir al paso 9. No: Ir al paso 11.
9	Sistema	Desactivar cuenta 10 minutos. Ir al paso 10.
10	Sistema	Mostrar Mensaje: "Datos incorrectos, favor de volver a intentar en 10 minutos". Ir al paso 1.
11	Sistema	¿Intentos < 9? Sí: Ir al paso 12.

		No: Ir al paso 13.
12	Sistema	Mostrar Mensaje: "Datos incorrectos". Ir al paso 1.
13	Sistema	Mostrar Mensaje: "Cuenta bloqueada, favor de contactar al Administrador del CNID". Ir al <i>Subproceso de Desbloqueo de Cuenta de Usuario</i> Fin del proceso.
14	Sistema	Intentos = 0 Ir al paso 15.
15	Sistema	Crear nuevo registro y variables de sesión. Conceder acceso al usuario registrado. Fin del proceso.

Tabla 4.1. Flujo de Datos de Acceso al Sistema

4.2.2 Subproceso de Desbloqueo de Cuenta de Usuario

En este proceso se describen los pasos para desbloquear la cuenta de un usuario. Como medida de seguridad, además de desbloquear al usuario, se cambiará su contraseña de acceso. En este proceso intervienen los siguientes roles:

- a) Usuario registrado. Es el personal que cuenta con un usuario y contraseña para ingresar al sistema.
- b) Sistema. Módulo de Administración de Usuarios dentro del Sistema SIGED.
- c) Administrador CNID. Es el personal del CNID responsable de desbloquear las contraseñas de usuario.

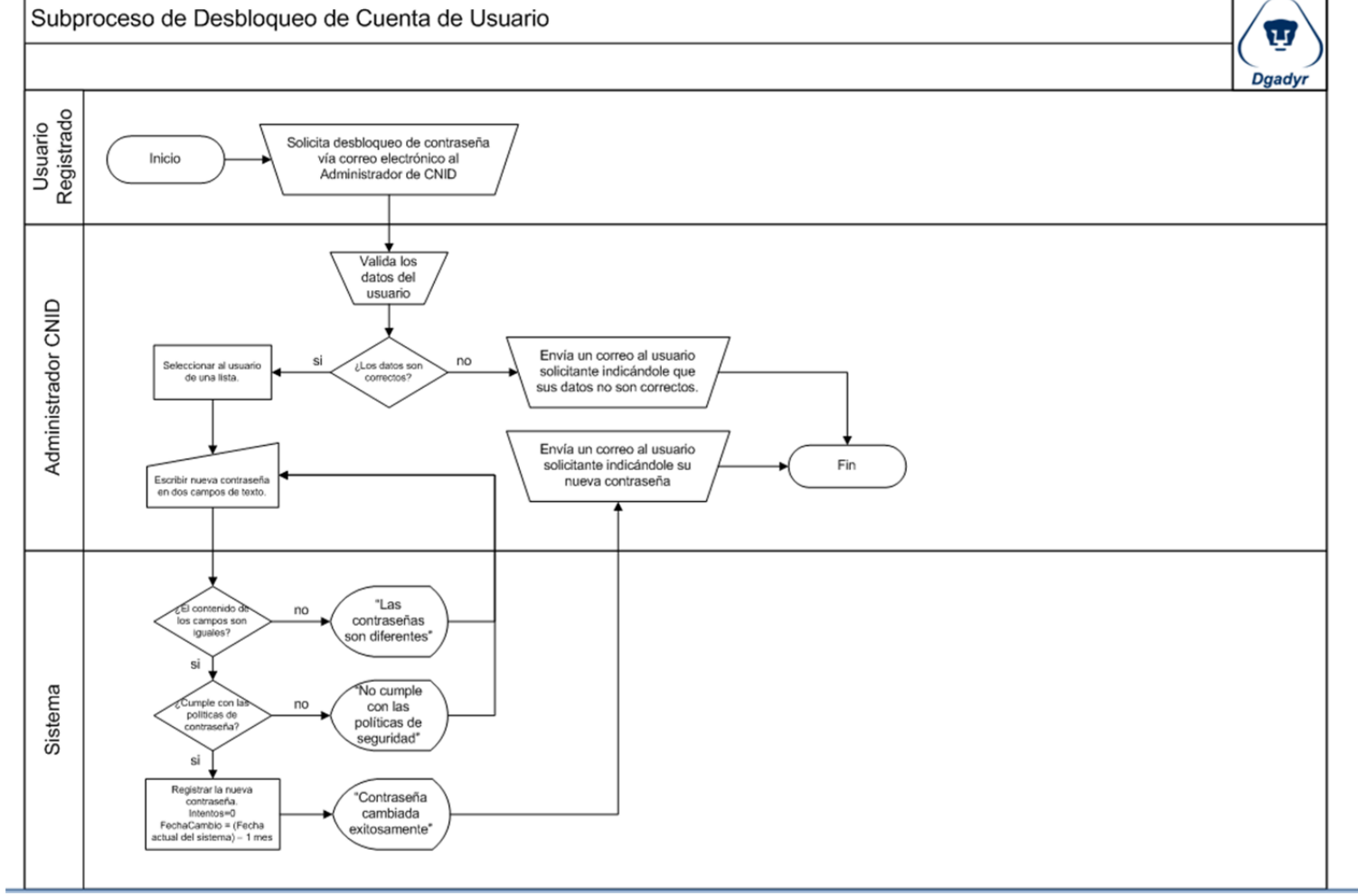


Figura 4.5 Diagrama del Subproceso de Desbloqueo de Cuenta de Usuario

El proceso que a continuación se describe (tabla 4.2) corresponde a los pasos definidos en el diagrama de flujo previo (figura 4.5).

Paso	Rol	Descripción
1	Usuario registrado	Solicita desbloqueo de contraseña vía correo electrónico al Administrador de CNID. Ir al paso 2.
2	Administrador CNID	Valida los datos del usuario. Ir al paso 3.
3	Administrador CNID	¿Los datos son correctos? Sí: Ir al paso 5. No: Ir al paso 4.
4	Administrador CNID	Envía un correo al usuario solicitante indicándole que sus datos no son correctos. Fin del proceso.
5	Administrador CNID	Seleccionar al usuario de una lista. Ir al paso 6.
6	Administrador CNID	Escribir nueva contraseña en dos campos de texto. Ir al paso 7.
7	Sistema	¿Los contenidos de los campos son iguales? Sí: Ir al paso 9. No: Ir al paso 8.
8	Sistema	Mostrar mensaje: “Las contraseñas son diferentes”. Ir al paso 6.
9	Sistema	¿Cumple con las políticas de contraseña? Sí: Ir al paso 11. No: Ir al paso 10.
10	Sistema	Mostrar mensaje: “No cumple con las políticas de seguridad”. Ir al paso 6.
11	Sistema	Registrar la nueva contraseña. Desbloquea al usuario. Regresa a cero el contador de intentos fallidos. Ir al paso 12.
12	Sistema	Mostrar mensaje: “Contraseña cambiada exitosamente”. Ir al paso 13.
13	Administrador CNID	Envía un correo al usuario solicitante indicándole su nueva contraseña. Fin del proceso.

Tabla 4.2 Flujo de Datos de Desbloqueo de Cuenta de Usuario

4.2.3 Subproceso de Registro de Actividades Realizadas por el Usuario

En este proceso se describen los pasos necesarios para registrar un evento en la bitácora. El único rol que interviene en este proceso es el sistema, particularmente el módulo de Bitácora del Sistema.

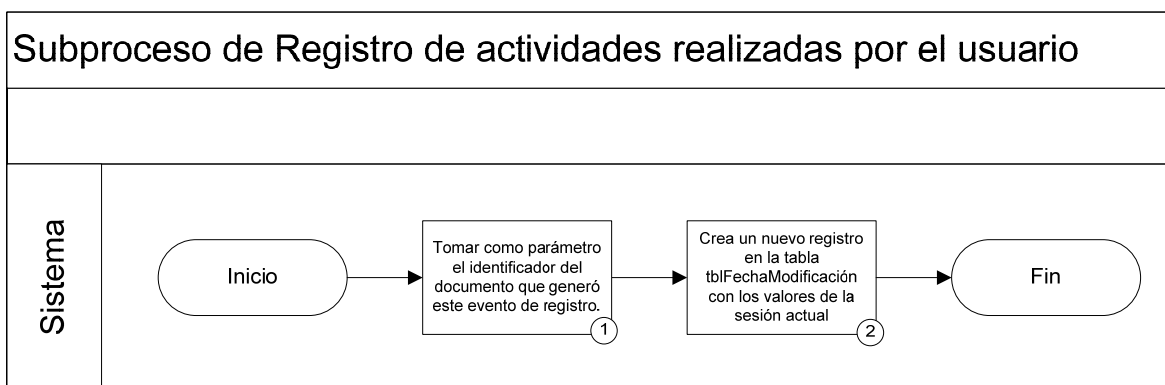


Figura 4.6 Diagrama del Subproceso de Registro de Actividades Realizadas por el Usuario.

Paso	Rol	Descripción
1	Sistema	Tomar como parámetro el identificador del documento que generó este evento de registro. Ir al paso 2.
2	Sistema	Crea un nuevo registro en la Bitácora del Sistema con los valores de la sesión actual y el identificador del documento. Fin del proceso.

Tabla 4.3 Flujo de Datos del Registro de Actividades Realizadas por el Usuario.

4.2.4 Subproceso de Envío de Mensajes

En este proceso se describen los pasos necesarios para el envío de mensajes a los usuarios generados por algún evento en otros subprocesos. El único rol que interviene en este proceso es el sistema, particularmente el módulo de Envío de Mensajes.

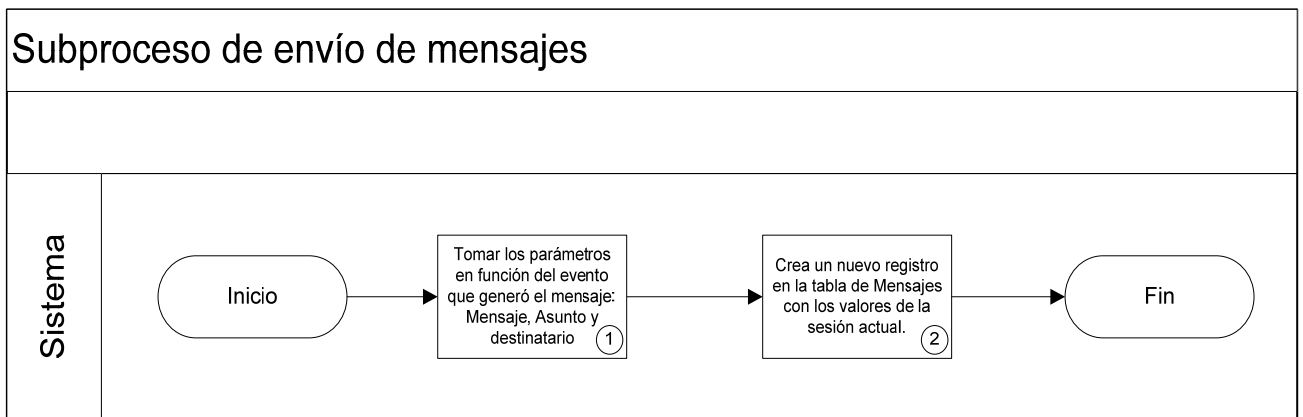


Figura 4.7 Diagrama del Subproceso de Envío de Mensajes

Paso	Rol	Descripción
1	Sistema	Tomar los parámetros en función del evento que generó el mensaje: mensaje, asunto y destinatario. Ir al paso 2.
2	Sistema	Crea un nuevo registro en la tabla de mensajes con los valores de la sesión actual. Fin del proceso.

Tabla 4.4 Flujo de Datos del Envío de Mensajes.

4.2.5 Subproceso de Evaluación Previa

En este proceso se describen los pasos necesarios para realizar la Evaluación Previa. Los usuarios que intervienen en este proceso se encuentran registrados, cuentan con los permisos adecuados y pueden acceder al sistema y a sus respectivos módulos. En el proceso intervienen los siguientes roles:

- a) Titular CEID. Este rol es el responsable de documentar este informe.
- b) Director CEID. El Director CEID es el primero en revisar la información, da su visto bueno y aprobación para este informe. En caso de ser necesario, el director puede regresar la información al titular para su corrección.
- c) Administrador CNID. Tiene la responsabilidad de dar el visto bueno final al informe. Además de revisar la información, puede regresar el documento al titular para su corrección en caso de ser necesario.

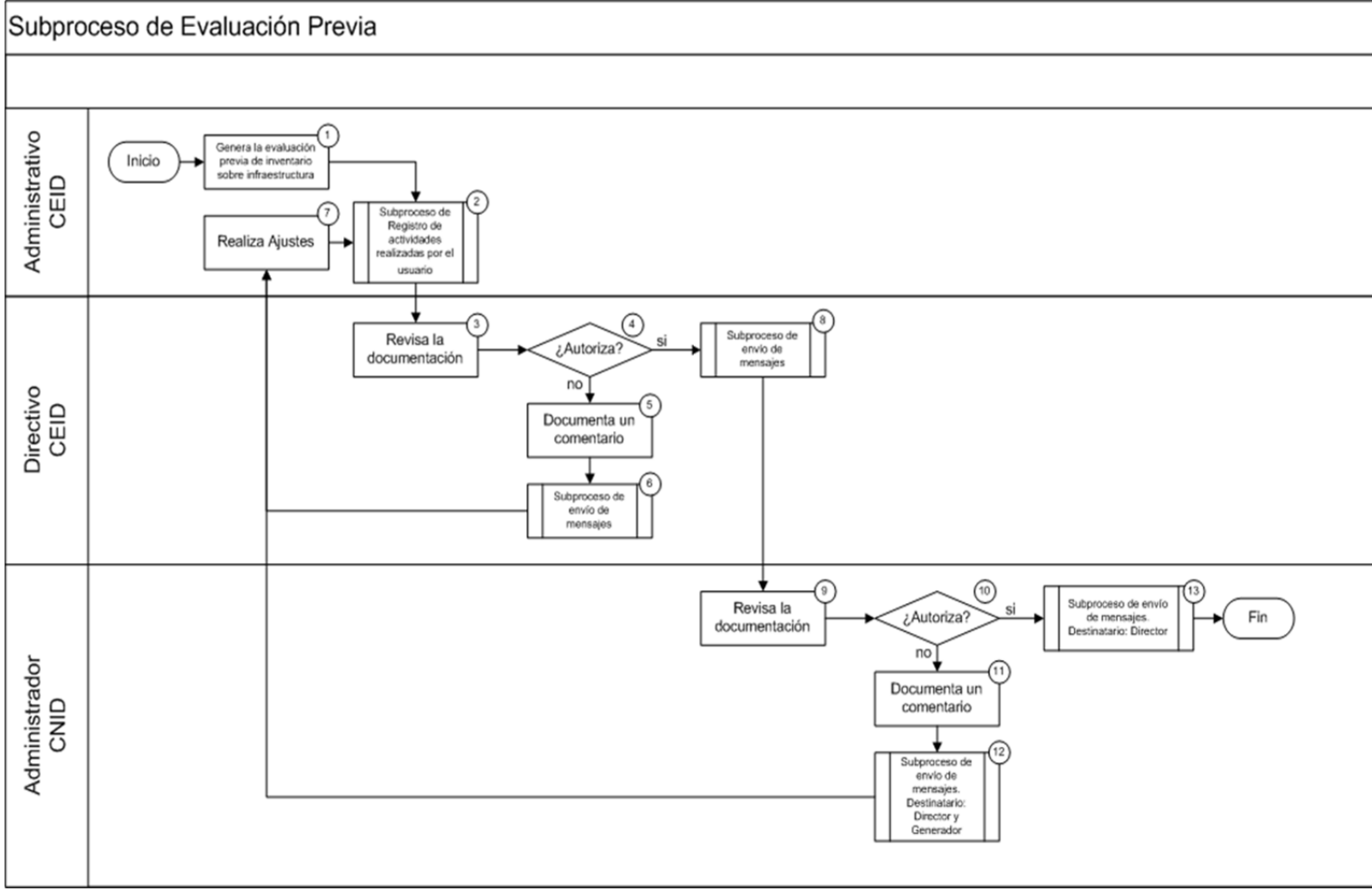


Figura 4.8 Diagrama del Subproceso de Evaluación Previa.

Paso	Rol	Descripción
1	Titular CEID	Genera la evaluación previa de inventario sobre infraestructura. Ir al paso 2.
2	Titular CEID	Subproceso de registro de actividades realizadas por el usuario. Ir al paso 3.
3	Titular CEID	Revisa la documentación Ir al paso 4.
4	Director CEID	¿Autoriza? Sí: Ir al paso 8. No: Ir al paso 5.
5	Director CEID	Documenta un comentario. Ir al paso 6.
6	Director CEID	Subproceso de envío de mensajes. Ir al paso 7.
7	Titular CEID	Realiza ajustes Ir al paso 2.
8	Director CEID	Subproceso de envío de mensajes Ir al paso 9.
9	Administrador CNID	Revisa la documentación Ir al paso 10.
10	Administrador CNID	¿Autoriza? Sí: Ir al paso 13. No: Ir al paso 11.
11	Administrador CNID	Documenta un comentario Ir al paso 12.
12	Administrador CNID	Subproceso de envío de mensajes. Destinatario: director y generador. Ir al paso 7.
13	Administrador CNID	Subproceso de envío de mensajes. Destinatario: director Fin del proceso.

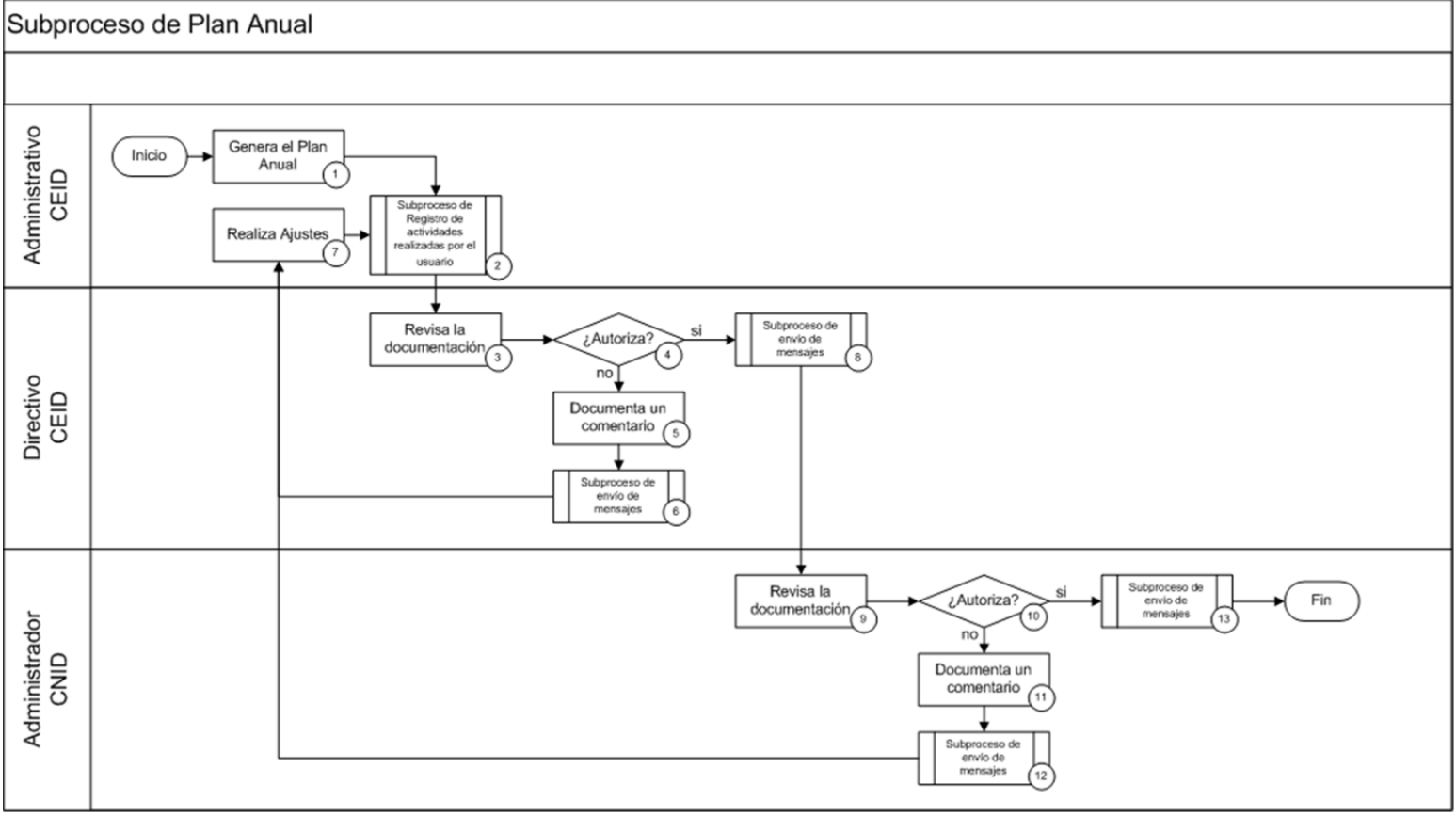
Tabla 4.5 Flujo de Datos de la Evaluación Previa.

4.2.6 Subproceso de Plan Anual

En este proceso se describen las actividades necesarias para realizar el Plan Anual. Antes de poder documentar este informe es necesario que el CEID cuente con su Evaluación Previa documentada y aprobada. El Plan Anual sigue el mismo flujo de aprobación de la Evaluación Previa. En este proceso intervienen los siguientes roles:

- a) Titular CEID. Este rol es el responsable de documentar el informe.
- b) Director CEID. Es el primero en revisar la información y da su visto bueno y aprobación para este informe. En caso de ser necesario, el director puede regresar la información al titular para su corrección.
- c) Administrador CNID. Tiene la responsabilidad de dar el visto bueno final del informe. Además de revisar la información, en caso de ser necesario puede regresar el documento al titular para su corrección.

Figura 4.9 Diagrama del Subproceso de Plan Anual.



Paso	Rol	Descripción
1	Titular CEID	Genera el Plan Anual Ir al paso 2.
2	Titular CEID	Subproceso de Registro de actividades realizadas por el usuario Ir al paso 3.
3	Director CEID	Revisa la documentación Ir al paso 4.
4	Director CEID	¿Autoriza? Sí: Ir al paso 8. No: Ir al paso 5.
5	Director CEID	Documenta un comentario Ir al paso 6.
6	Director CEID	Subproceso de envío de mensajes Ir al paso 7.
7	Titular CEID	Realiza ajustes Ir al paso 2.
8	Director CEID	Subproceso de envío de mensajes Ir al paso 9.
9	Administrador CNID	Revisa la documentación Ir al paso 10.
10	Administrador CNID	¿Autoriza? Sí: Ir al paso 13. No: Ir al paso 11.
11	Administrador CNID	Documenta un comentario Ir al paso 12.
12	Administrador CNID	Subproceso de envío de mensajes Ir al paso 7.
13	Administrador CNID	Subproceso de envío de mensajes Fin del proceso.

Tabla 4.6 Flujo de Datos del Plan Anual.

4.2.7 Subproceso de Recibo Fiscal

En este proceso se describen las actividades necesarias para documentar el recibo fiscal, los roles que intervienen son:

- a) Finanzas CEID. Es el responsable de digitalizar y documentar el recibo fiscal en el sistema.
- b) Administrador CNID. Debe de documentar los Adendums, el Anexo al Convenio y el Anexo Técnico.

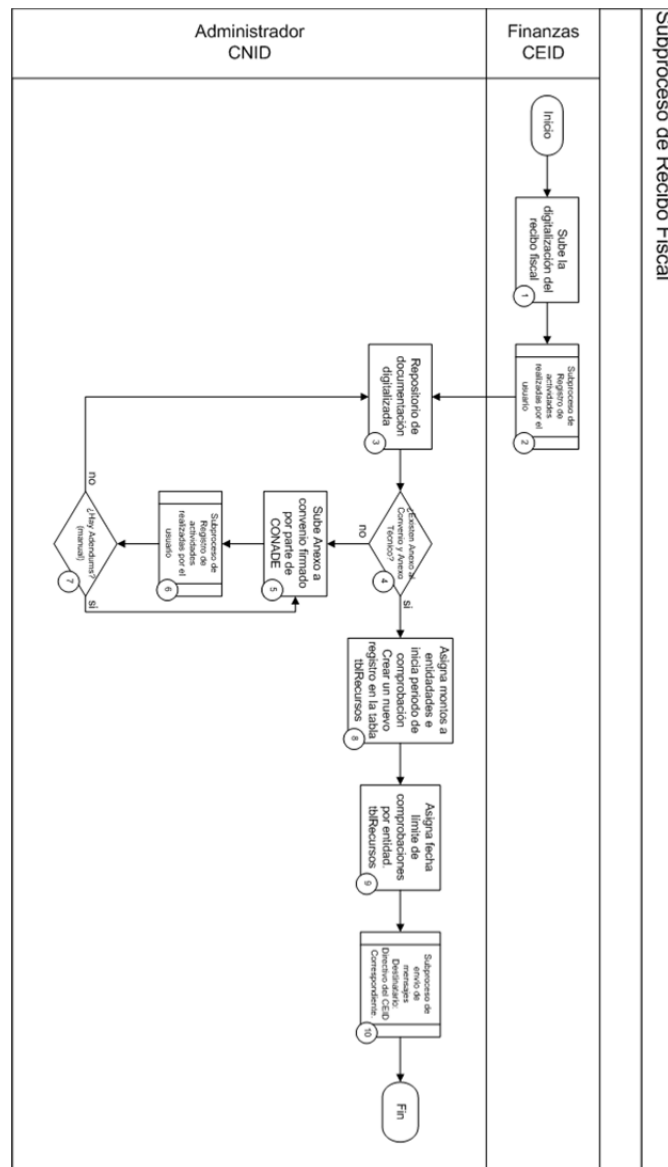


Figura 4.10 Diagrama del Subproceso de Recibo Fiscal.

Paso	Rol	Descripción
1	Finanzas CEID	Sube la digitalización del recibo fiscal Ir al paso 2.
2	Finanzas CEID	Subproceso de Registro de actividades realizadas por el usuario Ir al paso 3.
3	Administrador CNID	Repositorio de documentación digitalizada Ir al paso 4.
4	Administrador CNID	¿Existen Anexo al Convenio y Anexo Técnico? Sí: Ir al paso 8. No: Ir al paso 5.
5	Administrador CNID	Sube Anexo al Convenio firmado por parte de CONADE Ir al paso 6.
6	Administrador CNID	Subproceso de Registro de actividades realizadas por el usuario Ir al paso 7.
7	Administrador CNID	Verifica manualmente si hay Adendums. ¿Hay Adendums? Sí: Ir al paso 5. No: Ir al paso 3.
8	Administrador CNID	Asigna montos a entidades e inicia periodo de comprobación Ir al paso 9.
9	Administrador CNID	Asigna fecha límite de comprobaciones por entidad. Ir al paso 10.
10	Administrador CNID	Subproceso de envío de mensajes. Destinatario: Directivo del CEID correspondiente. Fin del proceso.

Tabla 4.7 Flujo de datos de Recibo Fiscal.

4.2.8 Subproceso de Comprobaciones

En este proceso se describen las actividades necesarias para realizar las comprobaciones, los roles involucrados son los siguientes:

- Finanzas CEID. Es el responsable de documentar este informe.
- Director CEID. Es el primero en validar y aprobar la información documentada. En caso de ser necesario puede regresar la información a Finanzas CEID para su revisión y corrección.
- Administrador CNID. Es el último en validar y aprobar la información documentada, en caso de ser necesario puede regresarla a Finanzas CEID para su revisión y corrección.

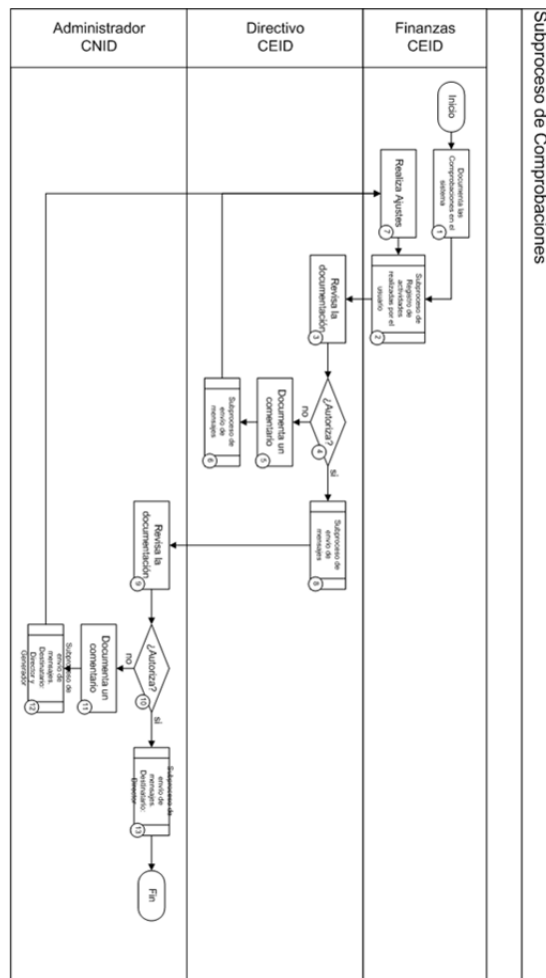


Figura 4.11 Diagrama del Subproceso de Comprobaciones.

Paso	Rol	Descripción
1	Finanzas CEID	Documenta las comprobaciones en el sistema Ir al paso 2.
2	Finanzas CEID	Subproceso de Registro de actividades realizadas por el usuario Ir al paso 3.
3	Director CEID	Revisa la documentación Ir al paso 4.
4	Director CEID	¿Autoriza? Sí: Ir al paso 8. No: Ir al paso 5.
5	Director CEID	Documenta un comentario Ir al paso 6.
6	Director CEID	Subproceso de envío de mensajes Ir al paso 7.
7	Finanzas CEID	Realiza ajustes Ir al paso 2.
8	Director CEID	Subproceso de envío de mensajes Ir al paso 9.
9	Administrador CNID	Revisa la documentación Ir al paso 10.
10	Administrador CNID	¿Autoriza? Sí: Ir al paso 13. No: Ir al paso 11.
11	Administrador CNID	Documenta un comentario Ir al paso 12.
12	Administrador CNID	Subproceso de envío de mensajes. Destinatario: Director CEID y Finanzas CEID Ir al paso 7.
13	Administrador CNID	Subproceso de envío de mensajes. Destinatario: Director CEID Fin del proceso.

Tabla 4.8 Flujo de Datos de las Comprobaciones.

4.3 Diagrama Entidad-Relación (DER)

A partir de los subprocesos y flujo de datos del sistema nos fue posible elaborar el Diagrama Entidad- Relación de la figura 4.12 que describe el diseño de la base de datos. Este diagrama puede verse de manera segmentada de acuerdo a la operación del sistema y la relación de los módulos descritos previamente con determinadas tablas del sistema.

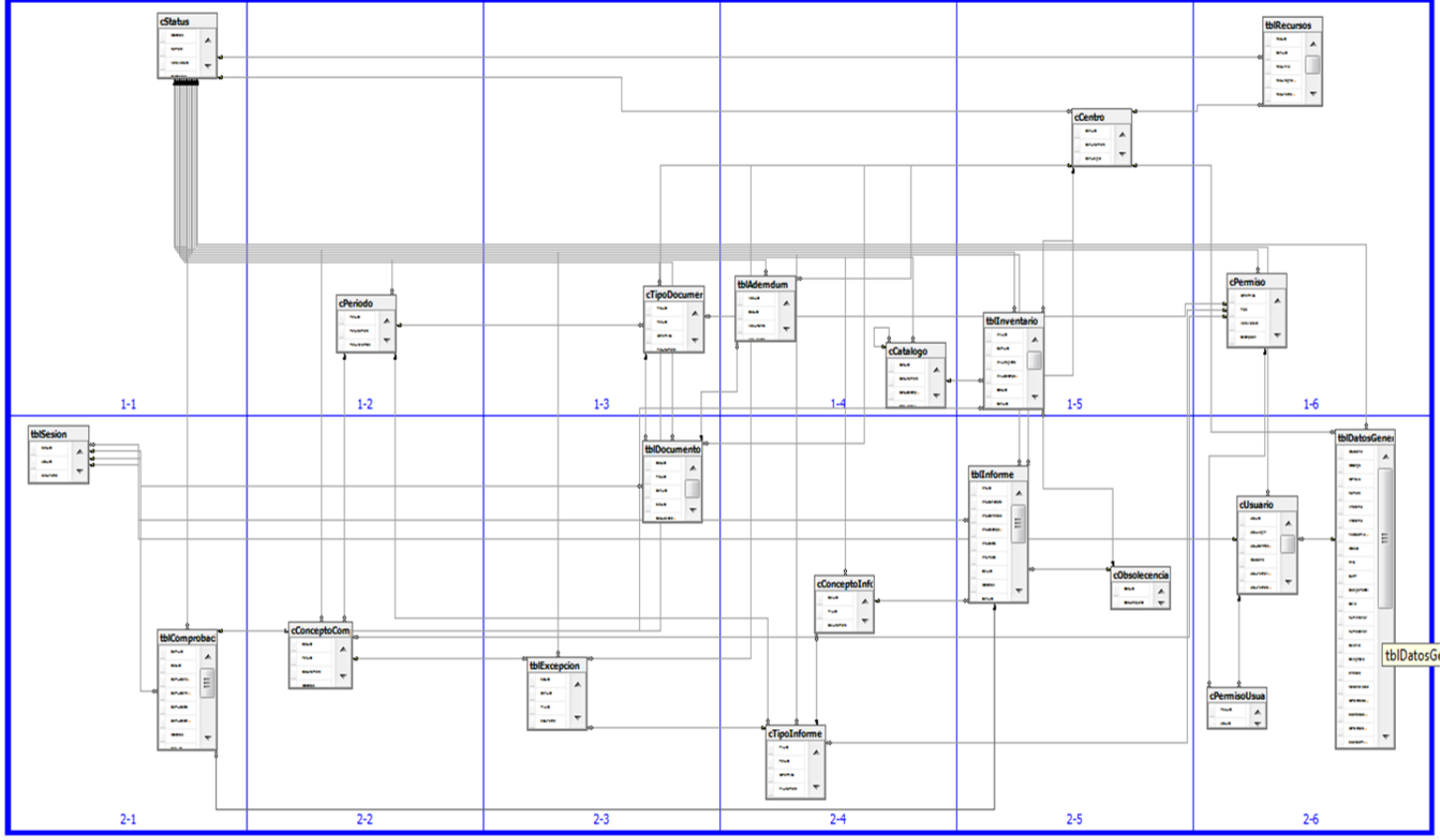


Figura 4.12 Diagrama Entidad – Relación de la Base de Datos del Sistema.

4.3.1 Sección de Control de Usuario

La figura 4.13 nos describe las tablas involucradas en los siguientes procesos:

- Acceso al sistema.
- Desbloqueo de cuenta de usuario.
- Registro de actividades realizadas por el usuario.
- Envío de mensajes.

Esta sección registra los datos personales de los usuarios del sistema y su relación con CEID al que pertenece, así como sus herramientas de autenticación y nivel de acceso al sistema, registrando todas sus actividades dentro de la plataforma, proveyendo de posibilidades de control estadístico y disciplinario de acuerdo al comportamiento de uso por parte de los usuarios y la normatividad operativa del CNID. El detalle de las tablas y los campos se expondrá en el diccionario de datos.

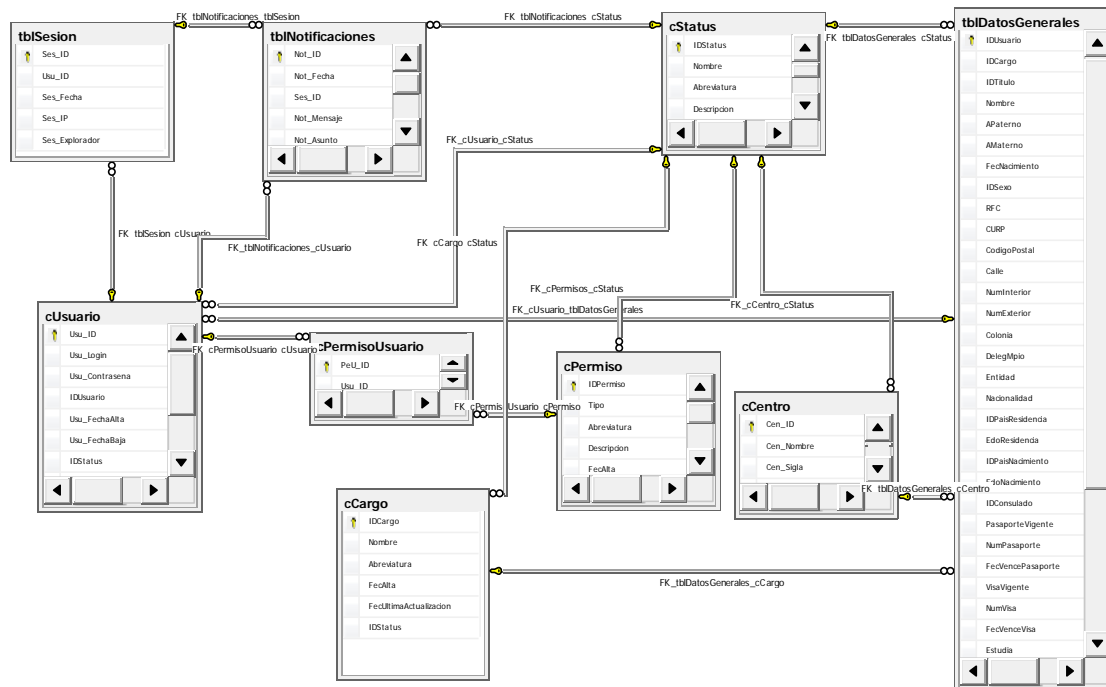


Figura 4.13 Sección de Control de Usuarios

4.3.2 Sección de Documentación

Esta sección es un repositorio de documentos legales almacenados en formato digital que otorgan identidad formal y legal a los CNID y CEID y definen la relación de colaboración, derechos y obligaciones entre esos actores.

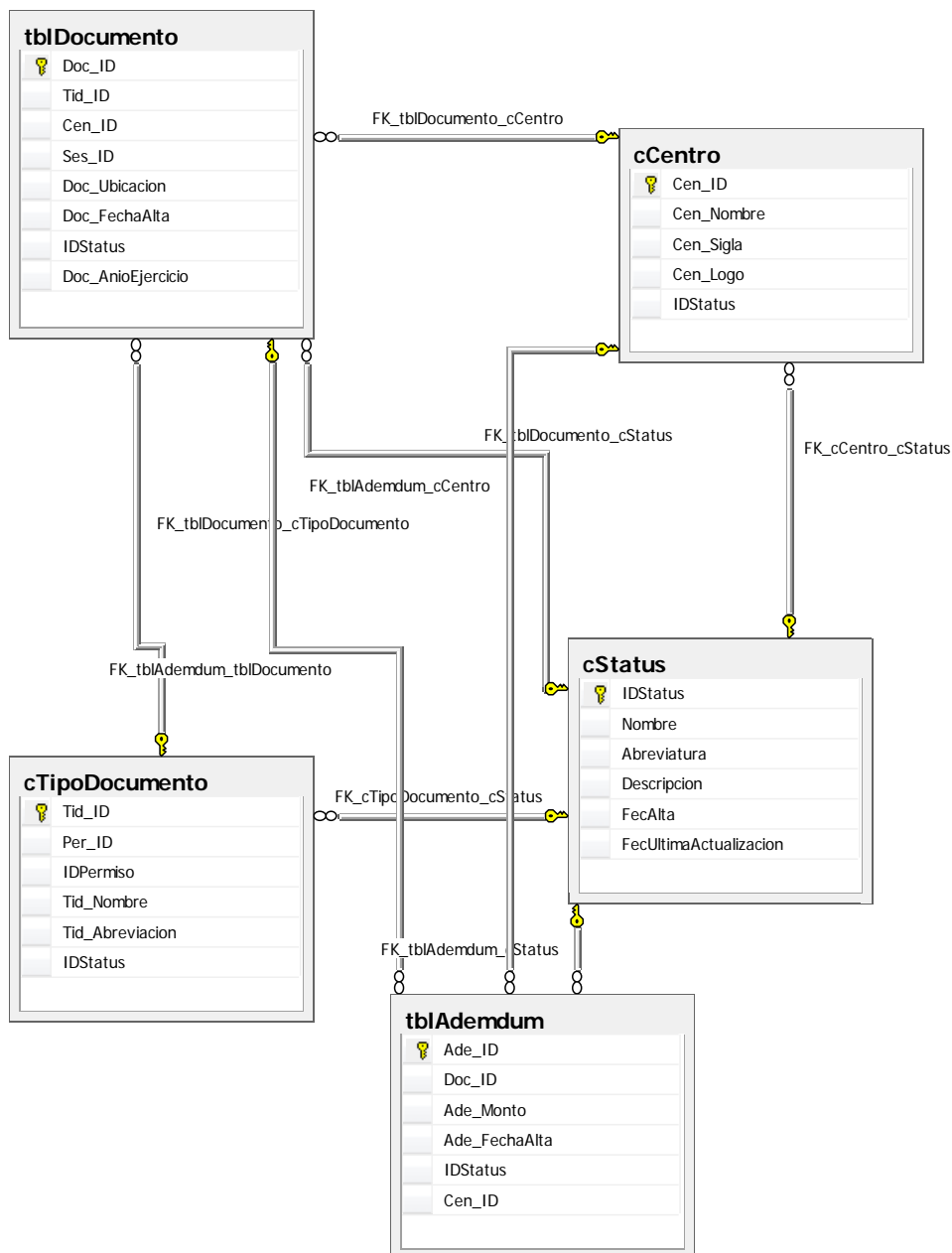


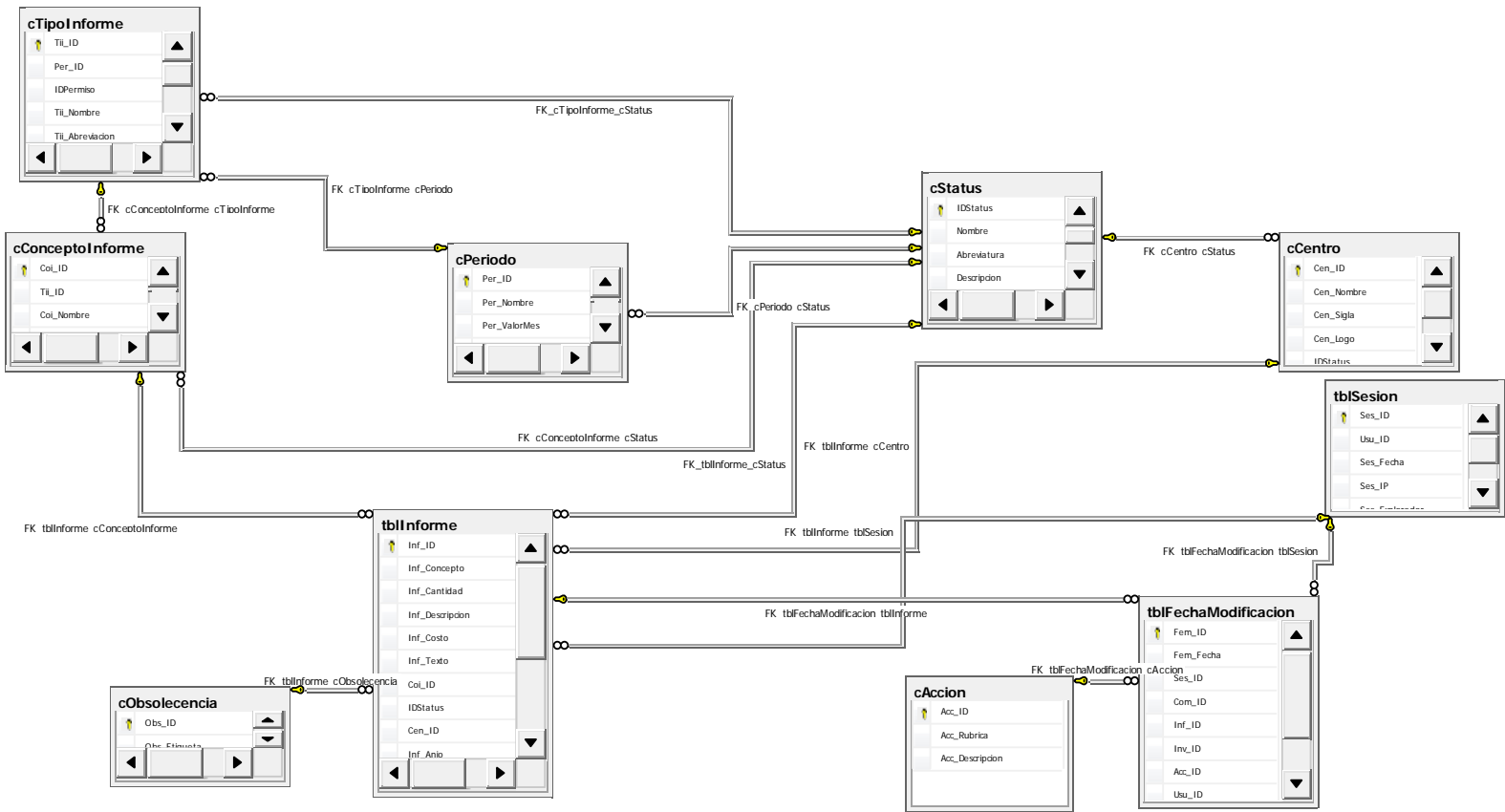
Figura 4.14 Sección de Documentación.

La figura 4.14 nos muestra como las tablas están organizadas para permitir tener un catálogo de documentos que permite agregar o inhabilitar documentos de acuerdo a las necesidades del usuario sin que esto implique un cambio a la estructura lógica del sistema. A partir de ello se construye una relación con el Centro Estatal y se genera el documento en una tabla específica.

4.3.3 Sección de Informes

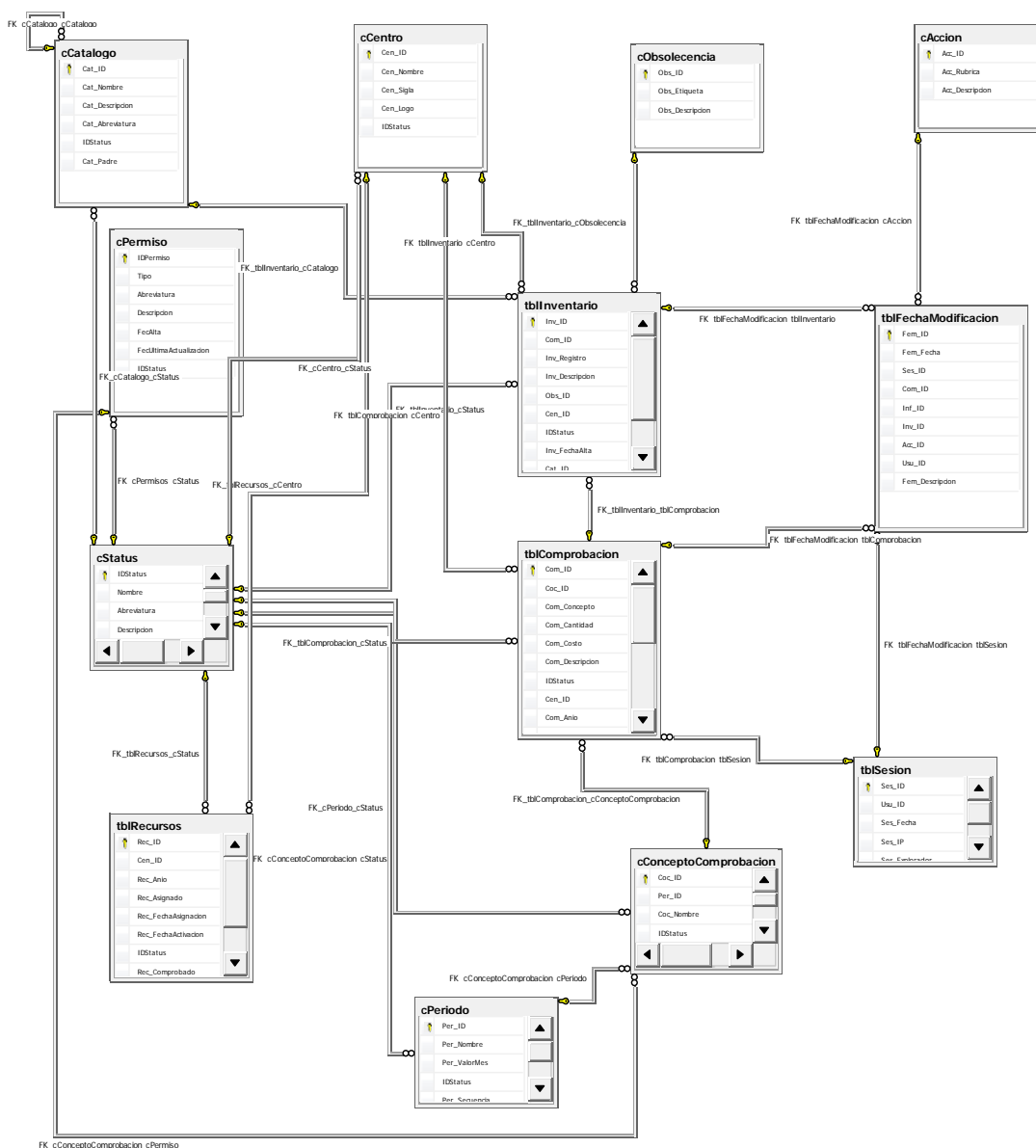
La normatividad del CNID dicta, entre las obligaciones de los CEID, la elaboración de diversos informes con periodicidades diferentes. La base de datos permite la generación de informes diversos con una serie de características flexibles, personalizables y con un amplio margen de opciones sin requerir una modificación a la estructura de las tablas involucradas. La figura 4.15 ilustra la situación.

Figura 4.15 Sección de Informes.



4.3.4 Sección de Comprobación

En la relación de colaboración CNID-CEID el manejo de recursos financieros es un ámbito que se debe controlar y monitorizar constantemente y de ahí que este módulo, a través de un diseño robusto, busque mantener un registro contante de los recursos entregados a los CEID y de la forma en que estos los ejercieron, manteniendo una constante documentación digital que respalde tal ejercicio.



4.16 Sección de Comprobación.

Figura

4.4 Diccionario de Datos

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE JERARQUIAS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
IDPermiso	Autonumérico		SI	PK	Identificador de los grados jerárquicos
Tipo	Varchar	20			Tipo de usuario
Abreviatura	char	4			Nombre corto del grado
Descripcion	Varchar	100			Descripción de los permisos que tiene el usuario
FecAlta	date/time				Fecha de alta en el sistema
FecUltimaActualizacion	date/time				Fecha de actualización de los valores de este registro
IDStatus	int		SI	FK	Estado del centro en el sistema (habilitado, eliminado, pendiente)

PROTOTIPO DE BASE DE DATOS CATÁLOGO DE CENTROS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Cen_ID	Autonumérico		SI	PK	Identificador de la relación de CEID - CNID
IDPermiso	int		SI	FK	Identificador de la jerarquía del CEID/CNID dentro del sistema
Cen_Nombre	Varchar	80			Nombre completo del centro (no siglas)
Cen_Sigla	Varchar	10			Siglas del centro
Cen_Logo	Varchar	50			Ruta donde está alojado el logo del centro
IDStatus	int		SI	FK	Estado del centro en el sistema (habilitado, eliminado, pendiente)

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE USUARIOS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Usu_ID	Autonumérico		SI	PK	Identificador del registro de los usuarios.
Usu_Login	nchar	10			Nombre de usuario dentro del sistema.
Usu_Contrasena	nchar	15			Contraseña del usuario
IDUsuario	int		SI	FK	Identificador de la persona que hace uso de la cuenta

Usu_FechaAlta	date/time				Fecha en la que se dio de alta el usuario dentro del sistema
Usu_FechaBaja	date/time				Fecha en la que se dio de baja el usuario dentro del sistema
IDStatus	int		SI	FK	Identificador de tipo de status del registro

PROTOTIPO DE BASE DE DATOS PARA AUTENTIFICACION DE USUARIOS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Ses_ID	Autonumérico		SI	PK	Identificador de la sesión
Usu_ID	int		SI	FK	Identificador del usuario que genera la sesión
Ses_Fecha	date/time				Fecha en que se genera la sesión
Ses_IP	nchar	15			Dirección IP desde donde se realiza la conexión
Ses_Explorador	Varchar	30			Identificador del navegador web que utiliza el usuario

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE PERIODOS TEMPORALES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Per_ID	Autonumérico		SI	PK	Identificador del período
Per_Nombre	nvarchar	50			Nombre del periodo
Per_Valores	int				Cantidad de meses que abarca el período
IDStatus	int		SI	FK	Estado del periodo dentro del sistema
Per_Secuencia	int				Secuencia del periodo respecto a las de su mismo tipo

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE DOCUMENTOS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Tid_ID	Autonumérico		SI	PK	Identificador del tipo de documento
Per_ID	int	50	SI	FK	Periodo de tiempo al que corresponde
IDPermiso	int		SI	FK	Identificador de jerarquía de quien puede alterar el registro
Tid_Nombre	nvarchar	30			Nombre del documento
Tid_Abreveación	char	5			Nombre corto del documento

IDStatus	int		SI	FK	Estado del elemento dentro del sistema
----------	-----	--	----	----	--

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE DOCUMENTOS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Doc_ID	Autonumérico		SI	PK	Identificador del documento
Tid_ID	int		SI	FK	Señala el tipo de documento del cual se trata
Cen_ID	int		SI	FK	Señala el centro al cual pertenece el documento
Ses_ID	int		SI	FK	Señala la sesión que ha cargado el documento
Doc_Ubicacion	varchar	90			Ubicación física del archivo en el disco
Doc_FechaAlta	date/time				Fecha de carga en el sistema
IDStatus	int		SI	FK	Estado del elemento en el sistema

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE TIPOS DE INFORMES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Tii_ID	Autonumérico		SI	PK	Identificador del tipo de informe
Per_ID	int		SI	FK	Señala el periodo de tiempo que le corresponde al tipo en particular
IDPermiso	int		SI	FK	Indicación del nivel requerido para acceder a este tipo de informe
Tii_Nombre	varchar	20			Nombre del tipo de informe
Tii_Abreveacion	char	5			Nombre corto del tipo de informe
Tii_FechaLimite	date/time				Fecha límite para cargarlo al sistema
IDStatus	int		SI	FK	Estado del elemento en el sistema

PROTOTIPO DE BASE DE DATOS PARA CONTROL DE EXCEPCIONES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Exc_ID	Autonumérico		SI	PK	Identificador de la excepción
Cen_ID	int		SI	FK	Señala el centro al que se le otorga esta excepción.

Tii_ID	int		SI	FK	Indicación del tipo de informe al que corresponde la excepción
Exc_Fecha	date	20			Fecha efectiva para la nueva excepción
Exc_Comentario	text	5			Comentario o descripción de la excepción y sus circunstancias.
Coc_ID	int				Indicación del concepto de comprobación al que se le conceda la excepción (excluye al tipo de informe)
IDStatus	int		SI	FK	Estado del elemento en el sistema

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE CONCEPTOS DENTRO DE UN INFORME					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Coi_ID	Autonómico		SI	PK	Identificador del concepto dentro de un informe
Tii_ID	int		SI	FK	Señala el tipo de informe al que corresponde este concepto
Coi_Nombre	varchar	50			Nombre del concepto manejado
IDStatus	int		SI	FK	Estado del elemento en el sistema

PROTOTIPO DE BASE DE DATOS PARA CONTROL DE MODIFICACIONES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Fem_ID	Autonómico		SI	PK	Identificador de los eventos realizados
Fem_Fecha	date/time				Fecha y hora en que se realizó una modificación al sistema
Ses_ID	int		SI	FK	Indica la sesión responsable del cambio al informe
Inf_ID	int		SI	FK	Indica el informe que fue alterado
Com_ID	int		SI	FK	Indica el comprobante que fue alterado

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE INFORMES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Inf_ID	Autonómico		SI	PK	Identificador del informe
Inf_Concepto	varchar	50			Etiqueta o concepto del informe
Inf_Cantidad	int				Cantidad del concepto a informar
Inf_Descripción	text				Descripción del concepto
Inf_Costo	money				Costo unitario del concepto

Inf_Texto	text				Campo adicional para usarlo de acuerdo a necesidades
Coi_ID	int		SI	FK	Concepto al que hace referencia este informe
IDStatus	int		SI	FK	Estado del informe dentro del sistema
Cen_ID	int		SI	FK	Centro que presenta el informe
Inf_Anio	char	4			Año en el que se hizo ejercicio de este concepto

PROTOTIPO DE BASE DE DATOS PARA CATÁLOGO DE CONCEPTOS DENTRO DE UNA COMPROBACIÓN					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Coc_ID	Autonumérico		SI	PK	Identificador del concepto dentro de una comprobación
Per_ID	int		SI	FK	Señala el periodo temporal al que corresponde este concepto
Coc_Nombre	varchar	50			Nombre del concepto manejado
IDStatus	int		SI	FK	Estado del elemento en el sistema
IDPermiso	int		SI	FK	Señala la jerarquía necesaria para acceder al recurso

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE COMPROBANTES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Com_ID	Autonumérico		SI	PK	Identificador de la comprobación
Com_Concepto	varchar	50			Etiqueta o concepto de la comprobación
Com_Cantidad	int				Cantidad del concepto a comprobar
Com_Descripción	text				Descripción del concepto
Com_Costo	money				Costo unitario del concepto
Coc_ID	int		SI	FK	Concepto al que hace referencia esta comprobación
IDStatus	int		SI	FK	Estado de la comprobación dentro del sistema
Cen_ID	int		SI	FK	Centro que presenta la comprobación
Com_Anio	char	4			Año en el que se hizo ejercicio de este concepto
Com_Factura	varchar	50			Ubicación física de la factura digitalizada

PROTOTIPO DE BASE DE DATOS PARA ALMACENAR LOS RECURSOS ECONÓMICOS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Rec_ID	Autonum		SI	PK	Identificador de la asignación de recursos

	érico				
Cen_ID	int		SI	FK	Señala el centro al que se le hace esta asignación
Rec_Año	char	4			Indica el año al que corresponde la asignación
Rec_Asignado	money				Cantidad monetaria de recurso asignado
Rec_FechaAsignación	date/time				Fecha en que se asigna el recurso
Rec_FechaActivación	date				Fecha en que esta activo el recurso para el uso por parte del CEID
IDStatus	int		SI	FK	Estado del elemento en el sistema
Rec_Comprobado	money				Cantidad monetaria de recurso comprobado

PROTOTIPO DE BASE DE DATOS PARA ALMACENAR EL INVENTARIO DE CADA CEID					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Inv_ID	Autonumérico		SI	PK	Identificador del elemento inventariado
Com_ID	int		SI	FK	Señala al documento que comprueba la adquisición del elemento
Inv_Registro	varchar	15			Indica el código de inventario asignado al activo
Inv_Concepto	varchar	30			Concepto al que pertenece el activo
Inv_Descripción	text				Descripción detallada del activo
Obs_ID	int		SI	FK	Señala el grado de obsolescencia
Cen_ID	int		SI	FK	Señala el centro al que pertenece el activo
IDStatus	int		SI	FK	Estado del elemento en el sistema
Inv_FechaAlta	date				Alta del activo en el sistema
Inv_FechaBaja	date				Baja del activo en el sistema

PROTOTIPO DE CATÁLOGO DE NIVELES DE OBSOLENCIA					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Obs_ID	Autonumérico		SI	PK	Identificador del elemento inventariado
Obs_Etiqueta	char	10			Señala al documento que comprueba la adquisición del elemento
Obs_Descripción	text				Indica el código de inventario asignado al activo

PROTOTIPO DE CATÁLOGOS DE ACCIONES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Acc_ID	Autonumérico		SI	PK	Llave primaria

Acc_Rubrica	char	4			Abreviatura de la acción
Acc_Descripcion	texto				Descripción de la acción

PROTOTIPO DE CATÁLOGOS DE CALIFICACIONES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Cal_ID	Autonumérico		SI	PK	Llave primaria
Cal_Nombre	Char	50			Nombre de la calificación
Cal_Valor	decimal				Valor de la calificación
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus
Cal_Abreviatura	Char	10			Abreviatura de la calificación

PROTOTIPO DE CATÁLOGOS DE CATÁLOGO					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Cat_ID	Autonumérico		SI	PK	Llave primaria
Cat_Nombre	char	50			Nombre del elemento de catálogo
Cat_Descripcion	char	150			Descripción del elemento de catálogo
Cat_Abreviatura	char	10			Abreviatura del elemento de catálogo
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus
Cat_Padre	numérico		SI	FK	Llave que hace referencia al padre del elemento de catálogo

PROTOTIPO DE CATÁLOGOS DE PERMISO - USUARIO					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
PeU_ID	numérico		SI	PK	Llave primaria
Usu_ID	numérico		SI	FK	Llave que hace referencia a la tabla cUuario
IDPermiso	numérico		SI	FK	Llave que hace referencia a la tabla cPermiso

PROTOTIPO DE CATÁLOGOS DE ESTATUS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
IDStatus	numérico		SI	PK	Llave primaria
Nombre	char	50			Nombre del estatus
Abreviatura	char	10			Abreviatura del estatus
Descripcion	char	100			Descripción del estatus
FecAlta	fecha				Fecha de alta del estatus
FecUltimaActualizacion	fecha				Fecha de modificación del estatus

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO ADENDUMS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Ade_ID	numérico		SI	PK	Llave primaria
Doc_ID	numérico		SI	FK	Llave que hace referencia a la tabla tblDocumento
Ade_Monto	dinero				Monto del Ademdem
Ade_FechaAlta	fecha				Fecha del Ademdem
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus
Cen_ID	numérico		SI	FK	Llave que hace referencia a la tabla cCentro

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO CUESTIONARIO					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Cue_ID	numérico		SI	PK	Llave primaria
Cue_Titulo	char	150			Título del cuestionario
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus
Cue_Abreviatura	char	10			Abreviatura del cuestionario

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE DATOS GENERALES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
IDUsuario	numérico		SI	PK	Llave primaria
Cen_ID	numérico		SI	FK	Llave que hace referencia a la tabla cCentro
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE NOTIFICACIONES					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Not_ID	numérico		SI	PK	Llave primaria
Not_Fecha	fecha				Fecha de la notificación
Ses_ID	numérico		SI	FK	Llave que hace referencia a la tabla
Not_Mensaje	texto				Mensaje de la notificación
Not_Asunto	char	150			Asunto de la notificación
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus
Not_Destinatario	numérico		SI	FK	Llave que hace referencia al destinatario

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE PREGUNTAS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Pre_ID	numérico		SI	PK	Llave primaria
Cue_ID	numérico		SI	FK	Llave que hace referencia a la tabla tblCuestionario
Pre_Factor	char				Factor del cuestionario
Pre_Consideraciones	char				Consideraciones de preguntas
Pre_Valor	numérico				Valor del Cuestionario
IDStatus	numérico		SI	FK	Llave que hace referencia a la tabla cStatus

PROTOTIPO DE BASE DE DATOS PARA ALMACENAMIENTO DE RESPUESTAS					
Nombre Campo	Tipo	Long	¿Es llave?	Tipo Llave	Descripción
Res_ID	numérico		SI	PK	Llave Primaria
Pre_ID	numérico		SI	FK	Llave que hace referencia a la tabla tblPregunta
Cen_ID	numérico		SI	FK	Llave que hace referencia a la tabla cCentro
Res_Impacto	decimal				Impacto de la respuesta
Res_Probabilidad	decimal				Probabilidad de la respuesta
Res_Otro	char	100			Respuesta opcional

4.5 Diseño de Módulos

Al término de los subprocesos y diagramas mencionados anteriormente pudimos definir claramente el funcionamiento de los diferentes módulos del sistema e identificamos los siguientes módulos funcionales del sistema:

- a) Módulo de Control de Acceso
- b) Módulo de Mensajería
- c) Módulo de Registro de Actividades
- d) Módulo de Informes
- e) Módulo de Comprobaciones

4.5.1 Módulo de Control de Acceso

El Módulo de Control de Acceso es aquel en el que intervienen todas las actividades relacionadas con el acceso al sistema, es decir; la pantalla de *login*, la administración de roles y permisos, la validación de permisos en los diferentes menús y visualizaciones. El módulo está formado por los siguientes subprocesos:

- a) Subproceso de Acceso al Sistema
- b) Subproceso de Desbloqueo de Cuenta de Usuario

4.5.2 Módulo de Mensajería

El Módulo de Mensajería tiene como objetivo el envío de notificaciones entre los diferentes usuarios, ayudándoles a estar informados respecto al flujo de aprobación de los informes y comprobaciones, además de alertas y mensajes de mantenimiento. El cual está formado por el Subproceso de envío de mensajes.

4.5.3 Módulo de Registro de Actividades

El Módulo de Registro de Actividades tiene el propósito de almacenar un registro detallado de las diferentes actividades que realizan los usuarios. Los eventos que se guardan en el registro son: acceso al sistema, envío de

información, validación de informes y comprobantes (aceptar y rechazar). Este módulo está formado por el subproceso de registro de actividades realizadas por el usuario.

4.5.4 Módulo de Informes

El objetivo del Módulo de Informes es el de registrar y dar seguimiento a los informes de Evaluación Previa y Plan Anual, además es el encargado de administrar el flujo de aprobación. Los subprocesos que forman este módulo son:

- a) Subproceso de Evaluación Previa
- b) Subproceso de Plan Anual

4.5.5 Módulo de Comprobaciones

El Módulo de Comprobaciones tiene el fin de registrar y dar seguimiento a las Comprobaciones y Recibo Fiscal así como administrar el flujo de aprobación. Lo forman los siguientes subprocesos:

- a) Subproceso de Recibo Fiscal
- b) Subproceso de Comprobaciones

Capítulo 5

Análisis de herramientas para la implementación del sistema

Al hablar de herramientas de desarrollo de soluciones informáticas podemos abordar un amplio espectro de opciones enfocadas a cubrir diferentes necesidades y centradas en aspectos puntuales. El apartado describe algunas herramientas de programación disponibles en el mercado para llevar a cabo sistemas con los requerimientos que se buscan. Se hace un recorrido a través de algunos de los diversos lenguajes de desarrollo web, detallando sus características, ventajas y desventajas; así como de los manejadores de bases de datos más representativos.

En ambos casos se realizarán comparativas entre las diversas herramientas y algunas particularidades que, tras un análisis complejo, derivan en una serie de propuestas que se ajustan de manera más certera al objetivo planteado en la concepción del proyecto.

5.1 Lenguajes de Desarrollo Web

Los inicios de Internet estuvieron marcados por la oferta de comunicación barata que se ofrecía a través del correo electrónico y la compartición de archivos, a través de terminales de comandos o de exploradores muy rudimentarios. Sin embargo, conforme pasó el tiempo fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos como HTML, que a través de hipervínculos revolucionó la manera de navegar por la web.

A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas y necesidades que requerían de una solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la web que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos. A continuación daremos una introducción a los diferentes lenguajes de programación enfocados a Internet.

5.1.1 Lenguaje HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Este es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language). Fue desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

Ventajas:

- a) Lenguaje sencillo que permite describir hipertexto.
- b) Texto presentado de forma estructurada y agradable.
- c) No necesita de grandes conocimientos sobre todo si se utiliza un editor de páginas web.
- d) Archivos pequeños.
- e) Despliegue rápido.
- f) Lenguaje de fácil aprendizaje.
- g) Lo admiten todos los exploradores.

Desventajas:

- a) Lenguaje estático.
- b) La interpretación de cada navegador puede ser diferente.
- c) Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- d) El diseño es más lento.
- e) Las etiquetas son muy limitadas.

5.1.2 Lenguaje Javascript

Es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications y es utilizado principalmente en el desarrollo de páginas web. Es similar a Java. Todos los navegadores en sus últimas versiones interpretan código Javascript.

El código Java script puede ser integrado dentro de las páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model).

Ventajas:

- a) Lenguaje de scripting seguro y fiable.
- b) Los script tienen capacidades limitadas por razones de seguridad.
- c) El código Java script se ejecuta en el cliente, liberando de carga al servidor.

Desventajas:

- a) El código es visible por cualquier usuario.
- b) El código debe descargarse completamente.
- c) Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting).

5.1.3 Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitios web. PHP es un acrónimo que significa “PHP Hypertext Pre-processor” (inicialmente se llamó Personal Home Page). Surgió en 1995 y fue desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado un servicio de PHP en el servidor como Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión .php. La última versión soporta la programación orientada a objetos.

Ventajas:

- a) Muy fácil de aprender.
- b) Se caracteriza por ser un lenguaje muy rápido.
- c) Soporta en cierta medida la orientación a objeto. Clases y herencia.
- d) Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- e) Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- f) Capacidad de expandir su potencial utilizando módulos.
- g) Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- h) Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- i) Incluye gran cantidad de funciones.
- j) No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- a) Se necesita instalar un servidor web.
- b) Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que el número de solicitudes aumente.
- c) La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- d) La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- e) Dificulta la segmentación en módulos de trabajo independientes.
- f) Dificulta la organización por capas de la aplicación.

5.1.4 Lenguaje ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitios web dinámicos. ASP significa en inglés (Active Server Pages) fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje necesitan, para su ejecución, tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl y Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Ventajas:

- a) Usa Visual Basic Script, siendo fácil para los programadores.
- b) Comunicación óptima con SQL Server.
- c) Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas:

- a) Código desorganizado.
- b) Se necesita escribir mucho código para realizar funciones sencillas.
- c) Tecnología propietaria.

5.1.5 Lenguaje ASP.NET

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba su antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para el funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. A partir del sistema Microsoft Windows 2003 se incluye este framework, solo siendo necesario instalarlo en versiones anteriores.

Ventajas:

- a) Completamente orientado a objetos.
- b) Controles de usuario personalizados.

- c) División entre la capa de aplicación o diseño y el código.
- d) Facilita el mantenimiento de grandes aplicaciones.
- e) Incremento de velocidad de respuesta del servidor.
- f) Mayor seguridad.

Desventajas:

- a) Mayor consumo de recursos.
- b) *Software* propietario.

5.1.6 JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma, creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Características:

- a) Programación en capas.
- b) Las páginas son compiladas en la primera petición.
- c) Permite separar la parte dinámica de la estática en las páginas web.
- d) Los archivos se encuentran con la extensión (jsp).
- e) El código JSP puede ser incrustado en código HTML.

Ventajas:

- a) Ejecución rápida del servlets.
- b) Crear páginas del lado del servidor.
- c) Multiplataforma.
- d) Código bien estructurado.
- e) Integridad con los módulos de Java.

- f) La parte dinámica está escrita en Java.
- g) Permite la utilización de servlets.

Desventajas:

- a) Complejidad de aprendizaje.

5.1.7 Lenguaje Python

Es un lenguaje de programación creado en el año 1990 por Guido Van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se dice que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- a) Programación orientada a objetos.
- b) Programación estructurada.
- c) Programación funcional.
- d) Programación orientada a aspectos.

Ventajas:

- a) Libre y fuente abierta.
- b) Lenguaje de propósito general.
- c) Gran cantidad de funciones y librerías.
- d) Sencillo y rápido de programar.
- e) Multiplataforma.
- f) Licencia de código abierto (Open source).
- g) Orientado a Objetos.
- h) Portable.

Desventajas:

- a) Lentitud por ser un lenguaje interpretado.

5.1.8 Lenguaje Ruby

Es un lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro “Matz” Matsumoto. Su sintaxis está inspirada en Python y Perl. Es distribuido bajo licencia de software libre (Open source).

Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla. Para los que deseen iniciarse en este lenguaje pueden encontrar un tutorial interactivo de Ruby.

Características:

- a) Sensible a mayúsculas y minúsculas.
- b) Dispone de manejo de excepciones.
- c) Ruby puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite.
- d) Portátil.

Ventajas:

- a) Permite desarrollar soluciones a bajo costo.
- b) Software libre.
- c) Multiplataforma.

Desventajas:

- a) Pobre eficiencia en cuanto a altas cantidades de usuarios accediendo al sistema.
- b) Pobre difusión.

5.2 Manejadores de Bases de Datos

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de

numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Para entender la estructura y función de una DBMS se deben tener claros los siguientes conceptos:

DDL

Data Description Language. Es un lenguaje de programación que sirve para definir las estructuras que almacenarán los datos, así como los procedimientos y funciones que permiten consultarlos. Lo que se busca con este lenguaje es hacer una descripción de los registros, campos y conjuntos que integran el modelo de datos del usuario.

Se usa para crear, modificar y destruir bases de datos y objetos relacionados a estas. Las sentencias DDL se compilan, dando como resultado un conjunto de tablas almacenadas en un tipo de archivo llamado diccionario de datos.

DML

Data Manipulation Language. Es una familia de lenguajes que tienen como objetivo la consulta, inserción, eliminación y actualización de datos en una base sin alterar de ninguna forma su estructura.

El ejemplo más conocido de esta tecnología es SQL.

El lenguaje de manipulación de datos se clasifica en:

- a) Lenguajes de consulta procedimentales. Se dan instrucciones al sistema para que realice algún procedimiento en la base y así obtener un resultado.
- b) Lenguajes de consulta no procedimentales. Se describe la información deseada sin un procedimiento específico para obtener dicha información.

QL

Query Language. Se refiere a algún lenguaje de programación utilizado para hacer consultas en una base de datos o un sistema de información empleando este parámetro como base para su clasificación.

Entre los lenguajes de consulta podemos encontrar a SQL, el cual es uno de los más reconocidos para el uso de bases de datos relacionales; otro es FQL, lenguaje de consulta usado en aplicaciones de Facebook.

GUI

Graphic User Interface. Hablando de forma general, una interfaz gráfica de usuario es aquella que nos permite la interacción con ciertos dispositivos de forma visual más que mediante comandos de texto. Enfocándonos en las bases de datos, podemos encontrar gran variedad de software en el mercado que nos permite realizar consultas de una forma más sencilla mediante una interfaz basada mayormente en imágenes. Ejemplo de este tipo de herramientas es Microsoft Access, phpMyAdmin y MySQL Lite Manager, por mencionar algunos.

QBE

Query By Example. Es un lenguaje de consulta desarrollado por IBM y es utilizado en bases de datos relacionales. Es el primer lenguaje de consultas gráfico y permite la visualización de tablas, introducción de comandos y condiciones.

El principio que rige QBE es que el parser puede transformar las acciones que el usuario lleva a cabo en sentencias para la manipulación de bases de datos. Esta característica permite reducir la carga al usuario y enfocarse en los aspectos más puntuales del lenguaje. Así mismo, nos provee un entorno más amigable y que permite mayor productividad tanto como para usuarios finales como para desarrolladores.

Actualmente QBE es soportado en varios manejadores de bases de datos, como Access y SQL Server, al igual en bases de datos orientadas a objetos.

Las funciones principales de un DBMS son:

- a) Crear y organizar la base de datos.
- b) Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos rápidamente.
- c) Manejar los datos de acuerdo a las peticiones de los usuarios.
- d) Registrar el uso de las bases de datos.
- e) 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.
- f) 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.
- g) Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- h) 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.

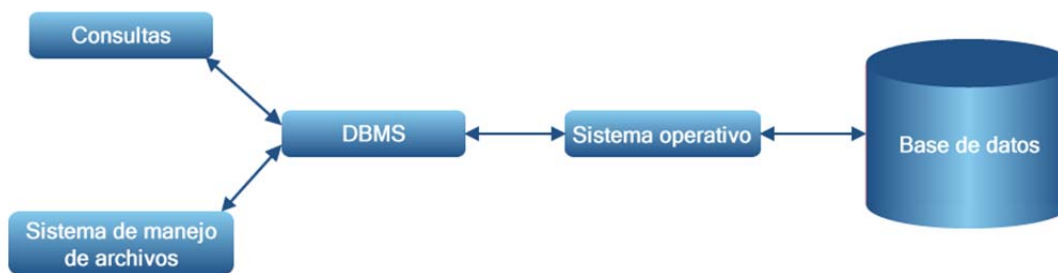


Figura 5.1 Estructura de un Manejador de Bases de Datos

La figura 5.1 muestra el DBMS como interface entre la base de datos física y las 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.

Todo lo anterior se resume en que un DBMS debe (deseablemente) cubrir el conjunto de características ACID (*Atomicity, Consistency, Isolation, Durability*).

5.2.1 DB2

DB2 es un popular DBMS surgido en 1982, creado por Informix, empresa que fue adquirida por IBM en el año 2001. Este manejador, dependiendo de su versión, está diseñado para poder trabajar en dispositivos que van desde PDA (Personal Digital Assistant) hasta mainframes. La última versión de este software propietario nos ofrece las siguientes características:

- a) Puede operar bajo las plataformas Windows, Linux y UNIX.
- b) Cumple con las características ACID.
- c) Soporta caracteres UNICODE.
- d) Maneja la tabla temporal, integridad referencial y GUI.
- e) Tamaño máximo de la base de datos: 512 TB.
- f) Tamaño máximo por tabla: 512 TB.
- g) Longitud máxima de números: 64 bits.
- h) Rango de fechas: 0001 – 9999.
- i) Como medidas de seguridad incluye:
 - 1. Encriptación de datos.
 - 2. Separación de rangos de usuario.
 - 3. Reglas de complejidad para contraseñas.
 - 4. Ejecución sin privilegios.
 - 5. Auditoría.
 - 6. Limitación de recursos.
 - 7. Certificación CC EAL4+

5.2.2 MySQL

MySQL es un muy popular DBMS con millones de instalaciones alrededor del mundo, proveyendo de acceso a múltiples usuarios a todas estas bases de datos. Fue desarrollado por MySQL AB que, ahora como

subsidiaria de Sun Microsystems, distribuye este software bajo licencias libres (GNU) y propietarias (versión Enterprise). La primera versión fue lanzada en 1996. La última nos ofrece las siguientes características:

- a) Puede operar bajo las plataformas Windows, Mac OS, Linux, BSD, Symbian, Solaris y UNIX.
- b) Cumple con las características ACID.
- c) Soporta parte de los caracteres UNICODE.
- d) Maneja la tabla temporal e integridad referencial.
- e) Tamaño máximo de la base de datos: “ilimitada”
- f) Tamaño máximo por tabla: 2 GB (16 GB en Solaris).
- g) Longitud máxima de números: 64 bits.
- h) Rango de fechas: 1000 – 9999.
- i) Como medidas de seguridad incluye:
 - 1. Encriptación de datos.
 - 2. Ejecución sin privilegios.
- j) No soporta las operaciones INTERSECT ni EXCEPT.

5.2.3 ORACLE

ORACLE es un poderoso DBMS desarrollado por Software Development Laboratories (hoy en día ORACLE Corporation) desde finales de la década de los 70s. Este software propietario está destinado a trabajar con bases de datos robustas, como nos lo indican las características que nos ofrece la última versión (11g):

- a) Puede operar bajo las plataformas Windows, Mac OS, Linux y UNIX.
- b) Cumple con las características ACID.
- c) Soporta caracteres UNICODE.
- d) Maneja la tabla temporal, integridad referencial.
- e) Tamaño máximo de la base de datos: ilimitado.
- f) Tamaño máximo por tabla: 4 GB x tablespace.
- g) Longitud máxima de números: 128 bits.
- h) Rango de fechas: -4712 – 9999.

- i) Como medidas de seguridad incluye:
 - 1. Encriptación de datos.
 - 2. Protección contra ataques de fuerza bruta.
 - 3. Reglas de complejidad para contraseñas.
 - 4. Auditoría.
 - 5. Limitación de recursos.
 - 6. Certificación CC EAL4+

5.2.4 PostgreSQL

PostgreSQL a diferencia de otros manejadores que son desarrollados por alguna compañía en particular, es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group) y lanzó su primera versión en 1989. La última versión nos ofrece, bajo la licencia BSD, las siguientes características:

- a) Puede operar bajo las plataformas Windows, Mac OS, Linux, BSD y UNIX.
- b) Cumple con las características ACID.
- c) Soporta caracteres UNICODE.
- d) Maneja la tabla temporal e integridad referencial.
- e) Tamaño máximo de la base de datos: ilimitado.
- f) Tamaño máximo por tabla: 32 TB.
- g) Longitud máxima de números: 64 bits.
- h) Rango de fechas: -4713 – 5,874,897.
- i) Como medidas de seguridad incluye:
 - 1. Encriptación de datos.
 - 2. Ejecución sin privilegios.
 - 3. Certificación CC EAL1

5.2.5 SQL Server

SQL Server es un DBMS desarrollado por Microsoft para el manejo de bases de datos robustas sobre el sistema operativo Windows, dicho manejador nos proporciona muchas ventajas para su interacción con desarrollos de .NET. Nos ofrece estas características:

- a) Opera bajo la plataforma Windows.
- b) Cumple con las características ACID.
- c) Soporta caracteres UNICODE.
- d) Maneja la tabla temporal, integridad referencial y GUI.
- e) Tamaño máximo de la base de datos: 524,258 TB (32767 archivos de 16 TB).
- f) Tamaño máximo por tabla: 524,258 TB.
- g) Longitud máxima de números: 64 bits.
- h) Rango de fechas: 0000 – 9999.
- i) Como medidas de seguridad incluye:
 1. Encriptación de datos.
 2. Reglas de complejidad para contraseñas.
 3. Ejecución sin privilegios.
 4. Auditoría.
 5. Limitación de recursos.
 6. Separación de rangos de usuario.
 7. Certificación CC EAL4+

5.3 Propuestas de Soluciones

Tras reconocer las opciones de desarrollo más representativas de aquellas disponibles en el mercado utilizadas para el desarrollo de aplicaciones similares a la planteada en este trabajo, corresponde el análisis y comparación de las mismas, así como la toma de la decisión final que define el conjunto de herramientas a utilizar.

El objetivo de esta sección es, precisamente, realizar la comparación de las opciones de lenguajes de desarrollo web así como de los manejadores de bases de datos, destacando sus propiedades en virtud de los requerimientos del sistema.

5.3.1 Comparativa de Bases de Datos

A continuación se muestra una tabla comparativa que permite observar las características que comparten los manejadores de bases de datos analizados.

Manejadores de Bases de Datos					
	DB2	MySQL	ORACLE	PostgreSQL	SQL Server
Plataformas Soportadas	Windows, Linux, UNIX	Windows, Mac OS, Linux, BSD, Symbian, Solaris y UNIX	Windows, Mac OS, Linux y UNIX	Windows, Mac OS, Linux, BSD y UNIX	Windows
ACID	Si	Si	Si	Si	Si
Tamaño Máximo de la Base de Datos	512 TB	Ilimitado	Ilimitado	Ilimitado	524,258 TB
Tamaño Máximo por Tabla	512 TB	2 GB (16 GB en Solaris)	4 GB x tablespace	32 TB	524,258 TB
Longitud Máxima de Números	64 bits	64 bits	128 bits	64 bits	64 bits
Rango de Fechas	0001 – 9999	1000 – 9999	-4712 – 9999	-4713 – 5,874,897	0000 – 9999
Certificaciones	CC EAL4+		EAL4+	EAL1	EAL4+
Notas		No soporta INTERSECT ni EXCEPT			GUI muy amigable.

Tabla 5.1 Manejadores de Bases de Datos

La tabla 5.1 muestra como los manejadores de bases de datos analizados comparten muchas características, entre ellas, el cumplimiento de las condiciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) una muy amplia capacidad de almacenamiento y características no muy distintas en el manejo de datos como números y fechas. La principal característica de dichos sistemas radica en su portabilidad, como SQL Server que ofrece una clara limitante a este renglón al depender, totalmente, para su funcionamiento de un ambiente Windows.

La figura 5.2 permite comparar la capacidad de almacenar datos por parte de las opciones consideradas en las líneas previas, siendo DB2 quien ofrece el menor poder de resguardo de información con 512 TB, frente a cantidades ilimitadas que brindan las demás opciones. Sin embargo, en la actualidad y para fines similares a los que se persiguen en el presente trabajo de tesis 512 TB no es, en absoluto, una cantidad despreciable de espacio disponible, e incluso rebasa en demasía la expectativa de uso de este sistema.

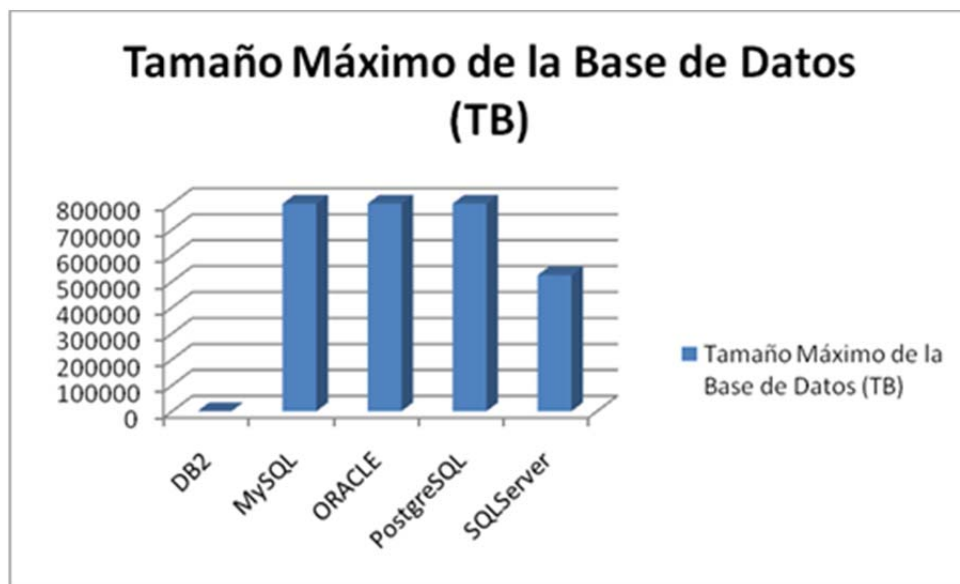


Figura 5.2 Gráfica comparativa de la capacidad total de almacenamiento de diferentes DMBS

La figura 5.3 muestra la oferta en cantidad de espacio a almacenar por tabla, en cada uno de los manejadores. Las condiciones, en apariencia, hablan de que todos los manejadores solo son capaces de tratar unos

pocos GB de información en cada tabla, con la excepción de SQL Server, que brinda, como límite, el mismo tamaño de la base.

Sin embargo, cabe hacer la aclaración de que ORACLE dedica 4 GB x Tablespace, donde un Tablespace es una unidad utilizada por el manejador que representa un segmento del disco etiquetado por el *software* de acuerdo a su funcionamiento interno. Esto significa que la columna de ORACLE cambiará su tamaño en el gráfico en función de los Tablespaces que sea capaz de etiquetar en el disco duro en uso.

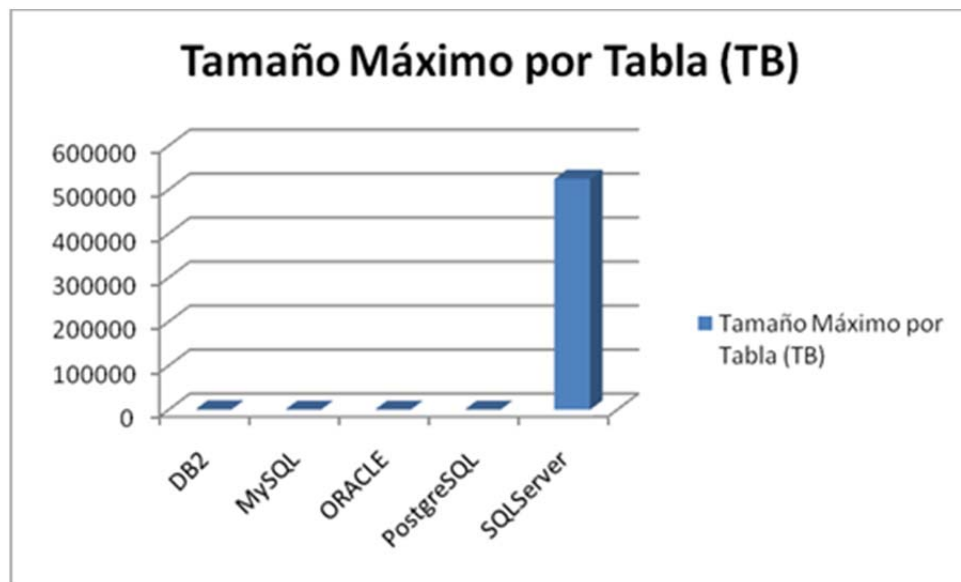


Figura 5.3 Gráfica comparativa de la capacidad de almacenamiento por tabla de diferentes DMBS

Cada manejador ha sido desarrollado con características específicas definidas por sus fabricantes en busca de un mayor aprovechamiento de los recursos de *hardware* disponibles sin descuidar las múltiples necesidades de los desarrolladores de sistemas. Una de estas características es la cantidad de bits disponibles para representar y almacenar cada tipo de variable en la base de datos, donde el tipo de dato numérico es común y necesario en todos los manejadores. Este tipo de variable define el rango de los números almacenables en función de los bits disponibles para la representación de cada cantidad y la figura 5.4 describe esta característica para los manejadores considerados. Se aprecia que ORACLE es capaz de otorgar una mayor gama de números a

presentar gracias a que cuenta con 128 bits para representar cada cantidad, mientras que el resto de los manejadores solo puede representar cantidades numéricas con 64 bits.

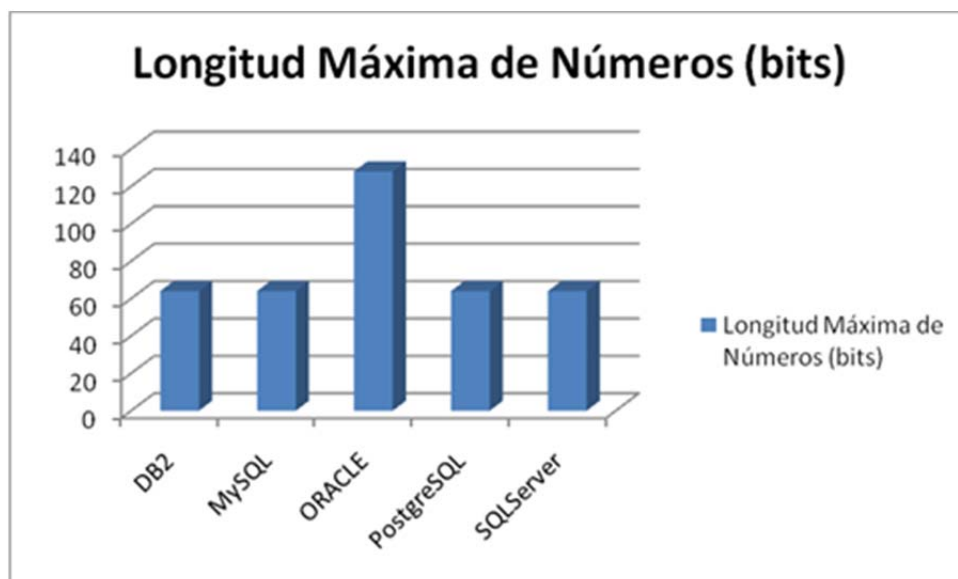


Figura 5.4 Gráfica comparativa de la máxima longitud numérica soportada por diferentes DMBS

Finalmente, pero sin demeritar su importancia, la tabla 5.2 describe las características de seguridad con las que cuentan los manejadores en análisis. Estas características ofrecen diversos grados de protección de la información almacenada y constituyen elementos adicionales a la seguridad alterna del sistema.

Características de Seguridad	DB2	MySQL	ORACLE	PostgreSQL	SQL Server
Encriptación de Datos	X	X	X	X	X
Separación de Rangos de Usuario	X				X
Reglas de Complejidad para	X		X		X

Password's					
Ejecución sin Privilegios	X	X		X	X
Auditoria	X		X		X
Limitación de Recursos	X		X		X
Protección a Ataques de Fuerza Bruta			X		

Tabla 5.2 "Características de seguridad de BD"

5.3.2 Comparativa de Lenguajes Web

Cada uno de los Lenguajes de Desarrollo Web fue diseñado con objetivos específicos y, con base en ello, sus capacidades son distintas. Esto se observa fácilmente, en la tabla 5.3.

	PHP	JavaScript	JSP	Python	Ruby	ASP .NET
Facilidad de Aprendizaje	Alta	Media	Media	Alta	Media	Media – Alta
Páginas Dinámicas	Si	Si	Si	Si	Si	Si
Estructura del Lenguaje	Lenguaje Estructurado y POO	Scripting	POO	Multiparadigma	POO	Scripting y POO
Facilidad de Diseño	Media	Media	Alta	Media	Alta	Alta
Licencia	Libre	Libre	Libre	Libre	Libre	Propietaria
Conexión a	Si	No	Si	Si	Si	Si

DB						
Lugar de Ejecución	Servidor	Cliente	Servidor	Servidor	Servidor	Cliente y Servidor
Requisitos	Explorador Web y Servidor Web	Explorador Web	Explorador Web y Servidor TOMCAT	Explorador Web y Servidor Web	Explorador Web y Servidor Web	Explorador Web y ISS

Tabla 5.3 "Características de los Lenguajes Web"

5.3.3 Propuestas

Como se aprecia en la figura 5.5, después del análisis realizado son tres las propuestas que finalmente se tienen:

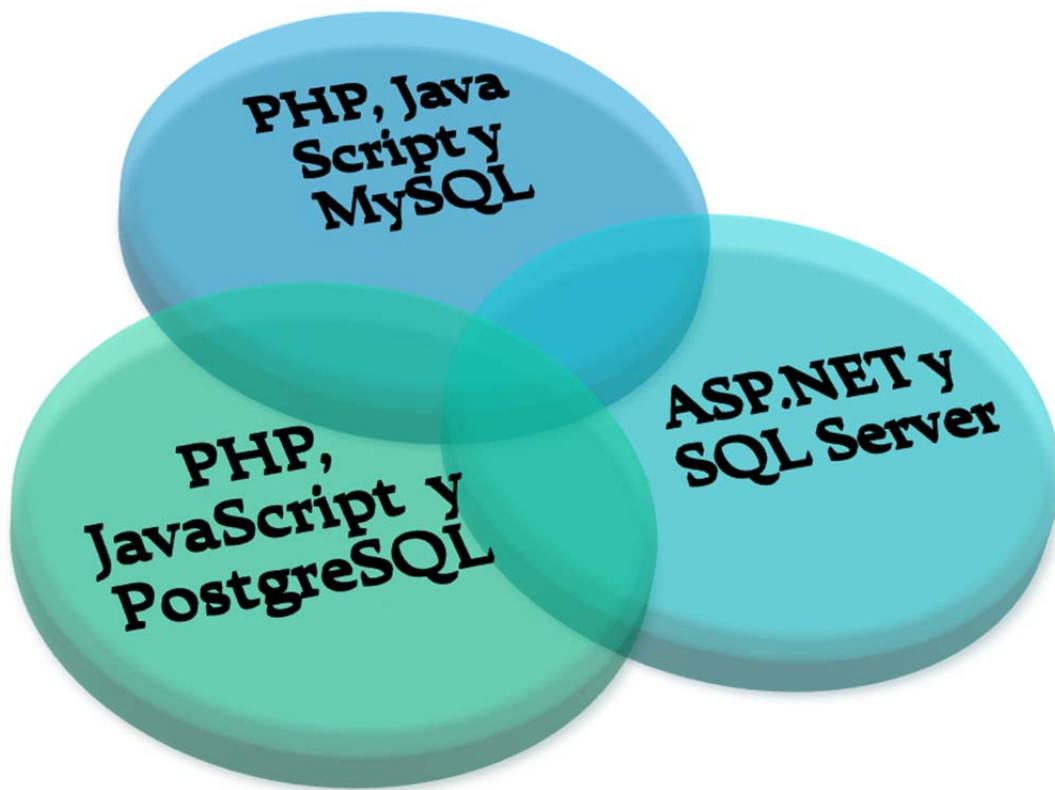


Figura 5.5 Propuestas

- a) PHP encargado de definir el comportamiento estructural del sitio, Java Script como opción de validación y apoyo en funcionalidades específicas; y MySQL, como el manejador de la base de datos.
- b) PHP encargado de definir el comportamiento estructural del sitio; Java Script como opción de validación y apoyo en funcionalidades específicas; y PostgreSQL, como el manejador de la base de datos.
- c) ASP.NET como encargado de definir el comportamiento estructural del sitio, así como todas las funciones y validaciones del sistema, y SQL Server como manejador de la base de datos.

5.4 Análisis

En el análisis se puede reflejar la viabilidad del sistema y con esto se puede ahorrar tiempo y costos, es decir, nos ayuda a analizar las diferentes propuestas, así como las variantes que puedan existir y de esta forma elegir la opción que más se adecue o se acerque a los objetivos planteados en el inicio del sistema.

5.4.1 Comparativa de las propuestas

Las propuestas mostradas en la tabla 5.4 nos ofrecen capacidades suficientes para el correcto funcionamiento del sistema, así como herramientas útiles para el desarrollo de todas las funcionalidades que serán demandadas por el proyecto.

Opción	Plataformas Soportadas	Tamaño Máximo por Tabla	Licencias	Estructura del Lenguaje	Facilidad de Diseño
PHP, Java Script y MySQL	Windows, Mac OS, Linux, BSD, Symbian, Solaris y UNIX	32 TB	Libre	Lenguaje Estructurado, Scripting y POO	Media

PHP, JavaScript y PostgreSQL	Windows, Mac OS, Linux, BSD y UNIX	2 GB	Libre	Lenguaje Estructurado, Scripting y POO	Media
ASP.NET y SQL Server	Windows	524,258 TB	Propietaria (Convenio con LMRS)	Scripting y POO	Alta

Tabla 5.4 Comparativa de propuestas

Una de las características que se ve con mayor disparidad es la gama de plataformas soportadas donde la propuesta hecha con base en las herramientas de Microsoft se ve limitada a operar sobre los sistemas operativos de la empresa, ya que de esta manera impiden que se proporcione una portabilidad del sistema. Mientras tanto, las opciones restantes son capaces de operar en una amplia gama de sistemas operativos donde los más populares, Windows, Linux y Unix, están incluidos.

La capacidad de almacenamiento, como se puede ver en la tabla 5.4 es amplia en los tres casos. Cabe aclarar que la versión libre de MySQL tiene limitaciones en varios rubros y el espacio es una de ellas.

Los paradigmas ofrecidos en los tres juegos de plataformas obedecen a las tendencias actuales de programación (POO) y a las capacidades de distribución de tareas en el cliente, scripting, lo que se traduce en una amplia funcionalidad a disposición.

Capítulo 6

Implementación e implantación del sistema

En el capítulo 4 se identificaron cada uno de los casos de uso a partir de los cuales se fundamenta el diseño lógico del sistema y en el capítulo 5 se hace el análisis de las tecnologías de desarrollo y almacenamiento de información consideradas como opciones para lograr concretar la herramienta SIGED.

En este capítulo se procede a describir el desarrollo de los módulos, así como a las partes complementarias que permiten el buen funcionamiento, proporcionando previamente una explicación de las opciones tecnológicas que exige el sistema para controlar el flujo, procesamiento y almacenamiento de la información.

6.1 Arquitectura cliente / servidor

La arquitectura cliente / servidor significa que los equipos clientes contactan a un servidor, un equipo generalmente muy potente en materia de capacidad de entrada / salida que proporciona servicios a los equipos solicitantes.

Los servicios son utilizados por programas denominados programas clientes que se ejecutan en los equipos periféricos, de ahí que se utilice el término “cliente” cuando se hace referencia a un programa que ha sido diseñado para ser ejecutado en un equipo que es capaz de procesar los datos recibidos de un servidor. Por ejemplo, en el caso del cliente FTP se trata de un *software* que procesa archivos en un servidor remoto, mientras que para el cliente de correo se trata del *software* que descarga los mensajes de correo electrónico recibidos en el equipo servidor que nos otorga dicha prestación.

Un sistema cliente / servidor funciona tal como se detalla en la figura 6.1:

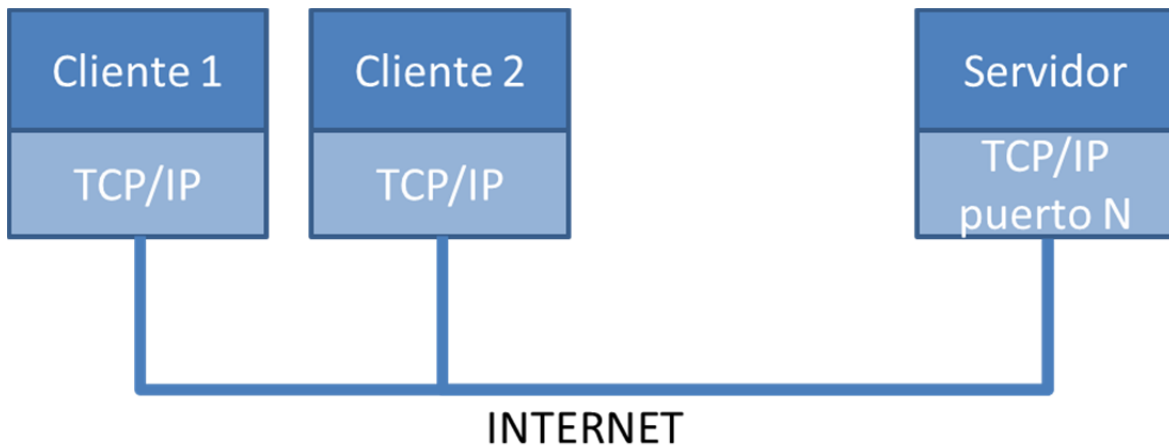


Figura 6.1 Diagrama de un sistema cliente / servidor.

El cliente envía una solicitud al servidor mediante su dirección IP y el puerto que está reservado para un servicio en particular que se ejecuta en el servidor. El servidor recibe la solicitud y responde con la dirección IP del equipo cliente y su puerto.

6.1.1 Arquitectura multicapa

La arquitectura multicapa se utiliza para describir a los sistemas cliente / servidor en donde el cliente solicita y el servidor (A) no responde directamente a la solicitud, sino que recurre a otro servidor (B) y efectúa una nueva solicitud y este tercero responde al segundo quien administra la información y responde finalmente al cliente. Esto significa que el servidor requiere de otra aplicación para proporcionar parte del servicio, tal y como se observa en la figura 6.2:



Figura 6.2 Arquitectura multicapa

Las aplicaciones de multiniveles distribuyen muchas de las funciones de los sistemas autónomos: presentan una interfaz de usuario, realizan el

proceso de pedido e informan el estado de la petición. Esta secuencia de comandos puede repetirse tantas veces como sea necesario.

La facilidad y la flexibilidad de los productos multiniveles es la base de muchas aplicaciones empresariales a gran escala. Esto responde a la necesidad de contar con un acceso más rápido a los datos y una reducción de tiempo al momento de desarrollar nuevas aplicaciones.

6.2 Modelos OSI y TCP/IP

El modelo OSI fue desarrollado en 1984 por la organización internacional de estándares llamada ISO, la cual es una federación global de organizaciones que representa a 130 países aproximadamente. El núcleo de este estándar es el modelo de referencia OSI, una normativa formada de siete capas que define las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones.

La utilidad de esta normativa estandarizada viene al haber muchas tecnologías, fabricantes y compañías dentro del mundo de las comunicaciones y al estar en continua expansión, se tuvo que crear un método para que todos pudieran entenderse de algún modo, incluso cuando las tecnologías no coincidieran. De tal forma que no importa la localización geográfica o el lenguaje utilizado. Todo el mundo debe atenerse a unas normas mínimas para poder comunicarse entre sí. Esto es, sobre todo, importante cuando hablamos de la red de redes, es decir, Internet.

Se debe pensar en las siete capas que componen el modelo OSI como una línea de ensamblaje en un ordenador, como se ve en la primera parte de la Figura 6.3. En cada una de las capas ciertas cosas pasan a los datos que se preparan para ir a la siguiente capa. Las siete capas se pueden separar en dos grupos bien definidos: grupo de aplicación y grupo de transporte.

En el grupo de aplicación tenemos:

- a) Capa 7: Aplicación. Esta es la capa que interactúa con el sistema operativo o aplicación cuando el usuario decide transferir archivos, leer mensajes, o realizar otras actividades de red. Por ello, en esta capa se incluyen tecnologías tales como http, DNS, SMTP, SSH, Telnet, etc.
- b) Capa 6: Presentación. Esta capa tiene la misión de coger los datos que han sido entregados por la capa de aplicación, y convertirlos en un formato estándar que otras capas puedan entender. En esta capa tenemos como ejemplo los formatos MP3, MPG, GIF, etc.
- c) Capa 5: Sesión. Esta capa establece, mantiene y termina las comunicaciones que se forman entre dispositivos. Se pueden poner como ejemplo, las sesiones SQL, RPC, NetBIOS, etc.

En el grupo de transporte tenemos:

- d) Capa 4: Transporte. Esta capa mantiene el control de flujo de datos y provee de verificación de errores y recuperación de datos entre dispositivos. Control de flujo significa que la capa de transporte vigila si los datos vienen de más de una aplicación e integra cada uno de los datos de aplicación en un solo flujo dentro de la red física. Como ejemplos más claros tenemos TCP y UDP.
- e) Capa 3: Red. Esta capa determina la forma en que serán mandados los datos al dispositivo receptor. Aquí se manejan los protocolos de enrutamiento y el manejo de direcciones IP. En esta capa hablamos de IP, IPX, X.25, etc.
- f) Capa 2: Datos. También llamada capa de enlaces de datos. En esta capa, el protocolo físico adecuado es asignado a los datos. Se asigna el tipo de red y la secuencia de paquetes utilizada. Los ejemplos más claros son *Ethernet*, *ATM*, *Frame Relay*, etc.
- g) Capa 1: Física. Este es el nivel de lo que llamamos llanamente hardware. Define las características físicas de la red, como las conexiones, niveles de voltaje, cableado, etc. Como habrás

supuesto, podemos incluir en esta capa la fibra óptica, el par trenzado, cable cruzados, etc.

Seguramente se habrá oído hablar de otro modelo paralelo al modelo OSI, llamado capas TCP/IP. Lo cierto es que son muy parecidas y las capas se entremezclan, como se aprecia en la Figura 6.3, solo que este último modelo solo utiliza niveles para explicar la funcionalidad de red. Las capas son las siguientes:

- a) Capa 1: Red. Esta capa combina la capa física y la capa de enlaces de datos del modelo OSI. Se encarga de dirigir los datos entre dispositivos en la misma red. También maneja el intercambio de datos entre la red y otros dispositivos.
- b) Capa 2: Internet. Esta capa corresponde a la capa de red. El protocolo de Internet utiliza direcciones IP, las cuales consisten en un identificador de red y un identificador de host, para determinar la dirección del dispositivo con el que se está comunicando.
- c) Capa 3: Transporte. Corresponde directamente a la capa de transporte del modelo OSI y donde podemos encontrar al protocolo TCP. El protocolo TCP funciona preguntando a otro dispositivo en la red si está deseando aceptar información de un dispositivo local.
- d) Capa 4: Aplicación. La capa 4 combina las capas de sesión, presentación y aplicación del modelo OSI. Protocolos con funciones específicas como correo o transferencia de archivos, residen en este nivel.

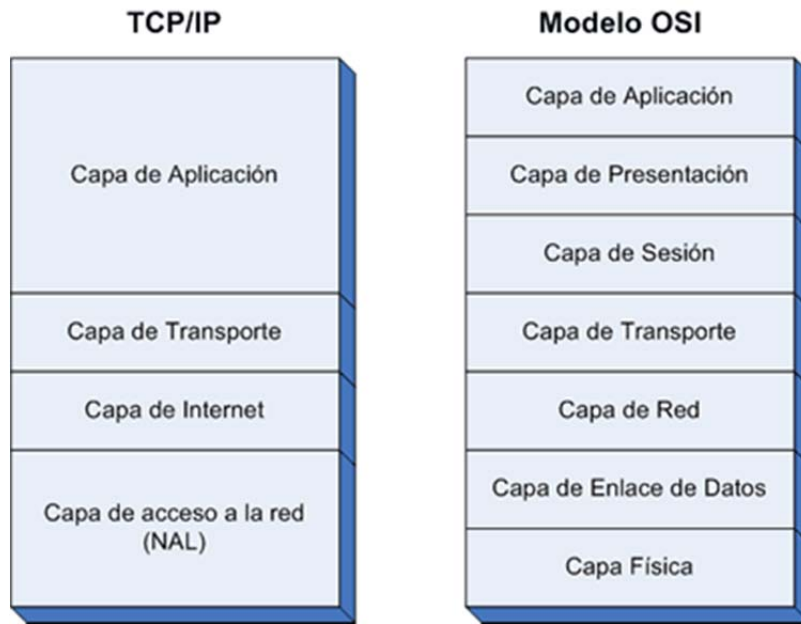


Figura 6.3 Modelos TCP/IP y OSI

6.3 Herramientas de Desarrollo

Microsoft C# es un lenguaje de programación diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Supone una evolución de Microsoft C y Microsoft C++ con un estilo sencillo y moderno, proporciona seguridad de tipos y está orientado a objetos. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de Common Language Runtime lo que se traduce en interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones.

Visual C# utiliza plantillas de proyecto, diseñadores, páginas de propiedades, asistentes de código, un modelo de objetos y otras características del entorno de desarrollo. La biblioteca para programar en Visual C# es .NET Framework.

6.3.1 Microsoft SQL Server

SQL Server es un sistema administrador de bases de datos relacionales basadas en arquitectura cliente / servidor (RDBMS) que usa Transact - SQL para mandar peticiones entre un cliente y el SQL Server.

El *software* cliente es responsable de la parte lógica y de presentar la información al usuario y generalmente corre en una o más computadoras cliente, aunque también puede correr en un Servidor con SQL Server.

SQL Server administra bases de datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc.) entre las múltiples peticiones.

La figura 6.4 muestra la arquitectura Cliente / Servidor de SQL server:



Figura 6.4 Arquitectura cliente / servidor de SQL Server

6.3.2 Servidor IIS

IIS, Servicios de Información de Internet (*Internet Information Services*, en inglés) es parte de la arquitectura de Windows Server. La función más importante de IIS es vincular los clientes que tienen acceso al sistema mediante el Protocolo de Transferencia de Hipertexto (HTTP) con los restantes servicios de Windows, como DHTML, ASP, etc. Además, IIS incluye un conjunto básico de funciones que los desarrolladores de sistemas pueden ampliar para definir una arquitectura personalizada de aplicaciones.

Define una funcionalidad básica que se puede utilizar para crear aplicaciones web. *Active Server Pages* (ASP) y otras tecnologías de Microsoft amplían esta funcionalidad básica para crear un entorno rico para el desarrollo de aplicaciones. La funcionalidad básica del servidor

se expone mediante la interfaz de programación de aplicaciones de servidor de Internet (ISAPI). Estas son las funciones básicas proporcionadas por IIS:

- a) Establecer y mantener conexiones HTTP.
- b) Leer peticiones HTTP y escribir respuestas HTTP.
- c) Modificar encabezados HTTP.
- d) Obtener la información del certificado del cliente.
- e) Administrar conexiones asincrónicas.
- f) Asignar localizadores de recursos universales (direcciones URL) a rutas físicas.
- g) Administrar y ejecutar aplicaciones.
- h) Transmitir archivos.

IIS y Servicios de componentes funcionan conjuntamente para formar la arquitectura básica para la creación de aplicaciones web y utiliza la funcionalidad proporcionada por Servicios de componentes para:

- a) Aislar las aplicaciones en procesos diferentes.
- b) Administrar la comunicación entre componentes COM (incluidos los objetos integrados de ASP).
- c) Coordina el proceso de transacciones para las aplicaciones ASP transaccionales.

IIS define las aplicaciones web como una colección de archivos de recursos agrupados en un espacio de nombres lógico. Al agrupar los recursos en aplicaciones, es posible compartir los datos en el espacio de nombres y ejecutar la aplicación en un procesos aislado.

Internamente, IIS coordina las aplicaciones aisladas mediante un objeto llamado administrador de aplicaciones web. Este objeto incluye una interfaz pública (IWAMAdmin) que puede utilizar para crear programas de administración de aplicaciones web. Al ejecutar una aplicación web en un proceso aislado, IIS utiliza los servicios de componentes para

coordinar el acceso simultáneo a los recursos y pasar información de contexto entre los componentes COM.

IIS utiliza el objeto `ObjectContext` de Servicios de componentes para proporcionar acceso a los objetos integrados de ASP.

6.3.3 Plataforma .NET

Es un conjunto de tecnologías diseñadas para transformar Internet en un sistema informático distribuido a escala completa. Proporciona nuevas formas de desarrollar aplicaciones a partir de colecciones de servicios web. La plataforma cuenta con un completo soporte de tecnologías de Internet independientes y basadas en estándares, incluyendo *Hypertext Transfer Protocol* (HTTP), *Extensible Markup Language* (XML) y *Simple Object Access Protocol* (SOAP).

Además de lo anterior, la plataforma .NET proporciona:

- a) Un modelo de programación coherente e independiente del lenguaje para todas las capas o niveles de una aplicación.
- b) Una interoperabilidad transparente entre tecnologías.
- c) Una fácil migración desde tecnologías existentes.

Los componentes principales del entorno .NET se describen a continuación:

Common Language Runtime (CLR). Es el mecanismo de ejecución para las aplicaciones .NET. Proporciona varios servicios, incluyendo la carga y ejecución del código, aislamiento de la memoria de las aplicaciones, administración de memoria, manejo de excepciones, acceso a metadata (información mejorada de tipos) y la conversión de MSIL (Lenguaje Intermedio Microsoft) a código nativo.

La *Base Class Library* (BCL). Proporciona un amplio conjunto de clases, lógicamente agrupadas en espacios de nombres jerárquicos que proporcionan acceso a las características más importantes del sistema operativo.

ADO.NET. Esta es una actualización evolutiva para la tecnología de acceso a datos *ActiveX Data Objects* (ADO) con mejoras importantes destinadas a la naturaleza desconectada del web.

ASP.NET. Esta es una versión avanzada de *Active Server Pages* (ASP) para el desarrollo de aplicaciones web (utilizando Formas web) y desarrollo de servicios web.

Common Language Specification (CLS). Esta especificación es responsable de hacer que muchas de las tecnologías antes mencionadas estén disponibles para todos los lenguajes que soportan .NET Framework. CLS no es una tecnología y no hay un código fuente para ella. Define un conjunto de reglas que proporcionan un contrato que rige la interoperabilidad entre los compiladores de lenguaje y las bibliotecas.

Win Forms. Modelo de programación y conjunto de controles. Proporciona una arquitectura sólida para el desarrollo de aplicaciones basada en Windows.

Visual Studio .NET. Ambiente de desarrollo, ofrece las herramientas que le permiten explotar las características del Framework .NET para crear aplicaciones concretas de una forma rápida y amigable.

6.4 Arquitectura del sistema

Debido a que este sistema va dirigido a una institución gubernamental a nivel nacional se optó por seguir una arquitectura de capas múltiples, con un servidor web que contendrá la aplicación, y con uno diferente donde estará la base de datos. Esto permite que la administración de los sistemas sea independiente, de tal manera que se pueden hacer mejoras de rendimiento tanto en las peticiones web como en las consultas a la base de datos.

Por otra parte, el protocolo utilizado entre los navegadores de los clientes y el servidor web es el http. De tal manera que el usuario final tiene acceso únicamente a pantallas de vista y de ninguna manera a los

archivos fuentes, ya que estos son procesados en el servidor de aplicación y con permiso de acceso local y por lo tanto tampoco tendrá acceso a la base de datos.

6.5 Interfaz de usuario

En cualquier sitio web intervienen diferentes tecnologías. Es fundamental el uso de páginas HTML, lenguajes *script* para realizar algunas operaciones y validaciones en el lado del cliente, acceso a bases de datos y servidor web entre otras. También se tomó como herramienta de presentación la utilización de Hojas de Estilos (CSS) que permiten estandarizar la apariencia en todas las pantallas y optimizar líneas de código.

Para realizar las interfaces de cada módulo del sistema primeramente se construyeron los prototipos de éstas de acuerdo a las características y funcionalidades con las que deberían contar. Para el diseño de estos prototipos se utilizó Microsoft Visual Studio, esta herramienta permite realizar de manera sencilla el diseño de interfaces con todos los elementos convencionales (cuadros de texto, botones, *combo box*, *frames*, etc.) y crear *Master Pages*. La característica que las *Master Pages* nos proporcionan es la habilidad de definir una estructura y elementos de interfaz comunes para nuestro sitio, tales como la cabecera de página o la barra de navegación, en una ubicación común para ser compartidos por varias páginas del sitio. Esto mejora la administración y mantenimiento de nuestro sitio y evita la duplicación innecesaria de código para estructuras o comportamientos del sitio que son comunes.

A partir de la idea establecida en los prototipos de las interfaces gráficas se construyeron los formularios tanto del menú principal como de los diferentes submódulos que constituyen el sistema. Con el código HTML de cada formulario y hojas de estilo se transformaron los formularios en interfaces gráficas amigables y fáciles de usar para el usuario.

Con base a lo mencionado anteriormente a continuación se muestran los prototipos realizados para los submódulos del sistema.

6.6 Aplicación web en .NET

La aplicación web con Visual Studio .NET (o proyecto) está integrada por los siguientes elementos:

- a) La configuración de red de la aplicación (puerto, dominio, seguridad, etc.)
- b) Un directorio virtual en IIS llamado SIGED, en el cual están los archivos que conforman la aplicación y controlan el acceso a los mismos. La creación de este se explica de forma más detallada en el Anexo I “Manual Técnico”.
- c) Un archivo de configuración llamado web.config en el cual se describen las variables públicas dentro de la aplicación y la cadena de conexión a la base de datos del sistema.

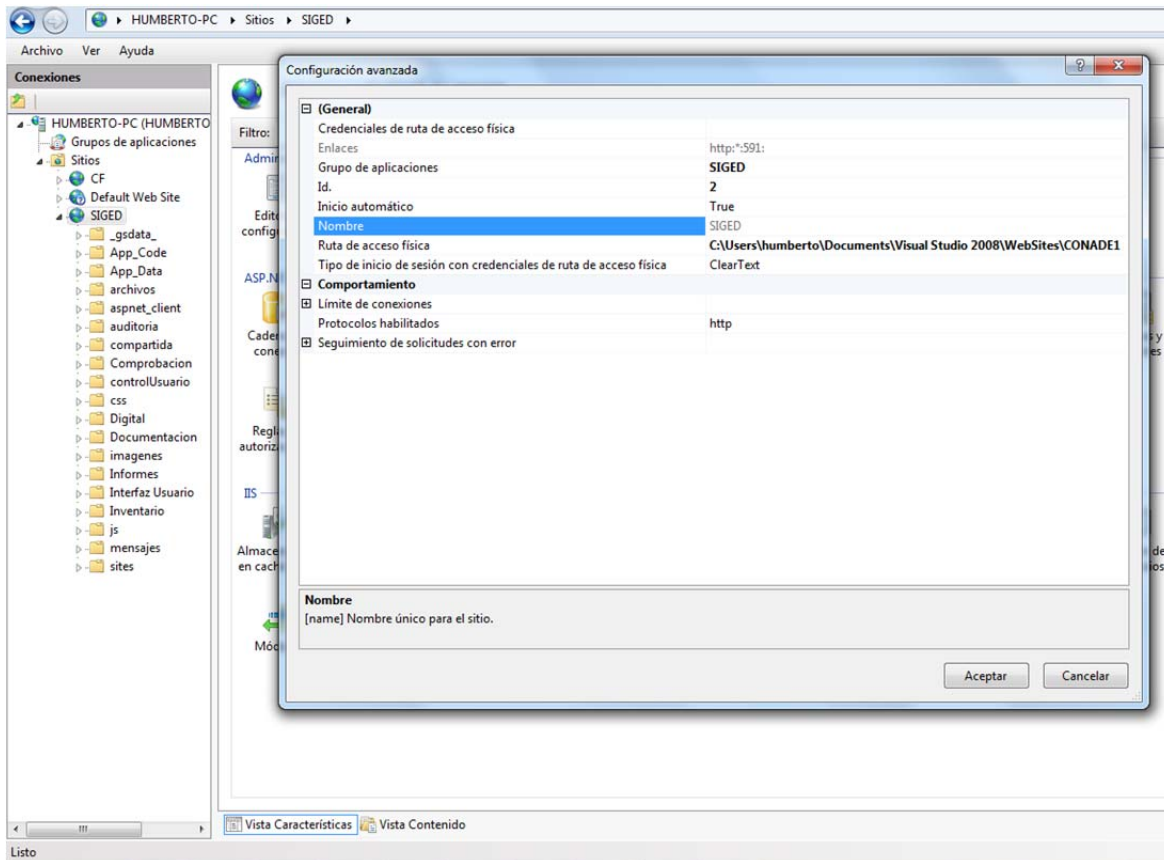


Figura 6.5 Configuración avanzada de IIS 7

El proyecto está dividido en diferentes directorios que representan cada uno de los módulos del sistema. Dentro de cada subdirectorio se encuentran los archivos *webform* que contienen la interfaz de usuario, así como los archivos de Código que describen el funcionamiento del sistema. Estos directorios se describen en el Anexo I “Manual Técnico”.

6.7 Base de Datos

El diseño de una base de datos se compone de tres etapas: diseño conceptual, lógico y físico. La etapa del diseño lógico es independiente de los detalles de implementación que depende del tipo de Sistema de gestión de base de datos (SGBD o DBMS) que se vaya a utilizar. La salida de esta etapa es el esquema lógico y la documentación que lo describe. Todo ello es la entrada para la etapa que viene a continuación, el diseño físico.

Mientras que en el diseño lógico se especificó qué se guarda, en el diseño físico se especifica cómo se va a guardar. Por ello, es importante que se tenga y se conozca muy bien toda la funcionalidad del DBMS concreto que se va a utilizar así como la plataforma de desarrollo sobre la que se va a trabajar. El diseño físico no es una etapa aislada, ya que algunas decisiones que se tomen durante su desarrollo, por ejemplo para mejorar las prestaciones, pueden provocar una reestructuración del esquema lógico.

Desde capítulos anteriores se hizo referencia al uso del paradigma de base de datos relacional para la operación del SIGED e, incluso, ya se ha abordado el diagrama entidad-relación y esta etapa del proceso de desarrollo traducirá ese modelo a una estructura lógica administrada por SQL Server 2008, acompañándolo de un soporte de procedimientos almacenados que darán funcionalidad a las pantallas y rejillas desplegadas en todos los módulos del sistema.

6.8 Seguridad

En la actualidad todo desarrollo de sistemas debe comprender lineamientos y parámetros mínimos de seguridad que respondan a mantener los principios de confidencialidad, disponibilidad e integridad de la información que almacenarán y administrarán y siguiendo esta línea es sistema incluye las siguientes líneas de defensa:

- a) Contraseñas no almacenadas en la base como texto plano, sino como un valor hash creado a partir de la contraseña asignada al usuario y un algoritmo conocido para tal fin.
- b) Bloqueo de cuentas de usuario tras repetidos intentos fallidos de ingreso.
- c) Validación de longitud máxima de caracteres y de no inserción de caracteres especiales en cada uno de los elementos de ingreso de información al sistema.

- d) Cada página busca y verifica los datos de sesión iniciada y ante cualquier movimiento en los datos del sistema, registra la dirección IP origen de la petición, fecha, hora y navegador desde el cual se hizo la consulta y/o petición al servidor.
- e) Política de respaldos frecuentes de la aplicación.
- f) Variables de sesión y cache con tiempo límite de vida de 30 minutos.

Estas medidas de seguridad buscan otorgar parámetros mínimos de seguridad y son complementarias a las ya existentes dentro de las instalaciones de la Dirección de Planeación y Tecnologías de la Información de la CONADE.

Capítulo 7

Pruebas y mantenimiento del sistema

Una vez que el sistema ha sido desarrollado, es necesario someterlo a una serie de pruebas que nos permitan identificar y mejorar aquellos puntos necesarios antes de entregarlo. Ya que se ha evaluado e implementado se deben establecer ciertas rutinas de mantenimiento las cuales permitan asegurar que el sistema continúe operando en el nivel requerido, dichas rutinas varían de acuerdo a las necesidades del sistema.

7.1 Pruebas y depuración del sistema

La prueba es un proceso que se enfoca sobre la lógica interna del *software* y las funciones externas. Es un proceso de ejecución de un programa que tiene como intención describir los puntos vulnerables dentro de un sistema.

Un proceso de ingeniería puede ser probado en las siguientes formas:

- a) Llevando a cabo pruebas que demuestren que cada función es completamente operativa.
- b) Desarrollando pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.
- c) Realizando pruebas de estabilidad, cobertura y rendimiento de arquitectura.

La prueba no es una actividad sencilla y no puede asegurar la ausencia de modificaciones en el *software*, por el contrario, conduce al beneficio principal: proporcionar retroalimentación mientras aún hay tiempo y recursos para hacer algo.

Para llevar a cabo esta fase de pruebas, el sistema fue examinado de acuerdo a cuatro aspectos principales:

7.1.1 Detección y modificación de programación lógica

Para poder detectar posibles fallas de programación en el sistema fue necesario contar con la participación de algunos usuarios finales para asegurar que éstas fueran completas e imparciales.

Estos usuarios se encargaron de proporcionar todo tipo de información dentro de los diferentes módulos y submódulos a fin de detectar resultados no esperados e imperceptibles al momento de desarrollar el sistema. Dichos resultados se estudiaron para modificar en algunos casos la lógica y la programación que los generaba y así obtener un desempeño del sistema más efectivo.

7.1.2 Realización de pruebas de funcionalidad de todos sus componentes, estimar tiempos de captura, transmisión y procesamiento de los datos

Todos los módulos fueron evaluados en cuanto a sus componentes y tiempos de respuesta y algunos de ellos requirieron de una mayor atención.

- a) Acceso
 - 1. Al Sistema
- b) Documentación
 - 1. De Informes
 - 2. De Comprobaciones
- c) Flujo de aprobación
 - 1. De Informes
 - 2. De Comprobaciones
- d) Alta
 - 1. De Documentos
- e) Registro
 - 1. De Actividades en el sistema
- f) Envío de Mensajes
 - 1. Durante las diferentes etapas del flujo de aprobación

2. Informativos por parte del Administrador CNID

7.1.3 Pruebas de usabilidad

Si bien es cierto que el término “Usabilidad” no tiene una validez como tal según la Real Academia de la Lengua Española, es ya bastante conocido dentro del medio de TI.

De acuerdo con lo señalado en el estándar ISO/IEC 25010 dentro del cual se identifican características de la calidad del *software* entre las cuales se encuentra la usabilidad, ésta se define como: “la capacidad de un producto de *software* para ser entendido, aprendido, utilizado y atractivo hacia el usuario cuando se usa bajo condiciones específicas”.

Aspectos de usabilidad a evaluar:

- a) Facilidad de aprendizaje
- b) Accesibilidad
- c) Flexibilidad
- d) Tiempo de respuesta
- e) Reducción de la carga cognitiva
- f) Recuperabilidad
- g) Buena imagen y estética

- "Un gramo de buen diseño vale kilos de soporte técnico." Brad Myers.

7.1.4 Identificar posibles mejoras de funcionalidad

Con base en las pruebas realizadas de la fase anterior se llevaron a cabo ciertas modificaciones en la interfaz y funcionalidad de los submódulos que requerían mejorar su desempeño. Estas modificaciones y detalles agregados, aun cuando fueron mínimos, permitieron minimizar tiempos de captura y respuesta, así como maximizar la efectividad del sistema.

7.2 Tipos de Mantenimiento

Si un problema es detectado por el usuario, inmediatamente puede notificarlo al administrador del sistema. Dicha petición debe ser atendida por el administrador y este procederá a diagnosticar de qué tipo de mantenimiento se trata. Atendiendo a estos fines, podemos establecer los siguientes tipos de mantenimiento:

- a) Correctivo. Cambios precisos para corregir errores del producto de *software*.
- b) Evolutivo. Incorporaciones, modificaciones y eliminaciones necesarias en un producto de software para cubrir la expansión o cambio en los requerimientos del usuario.
- c) Adaptativo. Modificaciones que afectan los entornos en los que el sistema opera, por ejemplo, cambio en las configuraciones del *hardware*, *software* de base, gestores de base de datos, comunicaciones, etc.
- d) Perfectivo. Acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración de código, definición más clara del sistema y optimización del rendimiento y eficiencia.

Una vez identificado el tipo de mantenimiento y su origen se determina un tiempo razonable para su modificación y prueba, haciéndolo del conocimiento del usuario.

Si se trata de un mantenimiento correctivo, los cuales son más comunes, se verifica y reproduce el problema, o se estudia la viabilidad del cambio propuesto por el usuario. En ambos casos se identifican, según el tipo de mantenimiento de que se trate, cuál es la más adecuada. El plazo y urgencia de la solución a la petición se establece de acuerdo con el estudio anterior.

Las tareas de los procesos de desarrollo que va a ser necesario realizar son determinadas en función de los componentes del sistema actual

afectados por la modificación. Estas tareas pertenecen a actividades de los procesos análisis, diseño y desarrollo.

Por último, y antes de la aceptación del usuario, es preciso establecer un plan de pruebas de regresión que asegure la integración del sistema de información afectado.

7.3 Revisiones periódicas

El monitoreo permanente del sistema asegura que las necesidades de mantenimiento sean identificadas y satisfechas cuando resulte necesario. Cuando el sistema es de uso prolongado, se puede establecer un mecanismo para recibir retroalimentación de los usuarios como una forma efectiva para determinar las necesidades de mantenimiento y modificación.

A los sistemas se les debe dar mantenimiento para asegurar que continúen operando en el nivel mostrado durante la etapa de prueba. Si los sistemas se deterioran, existe el riesgo de que no se desempeñen conforme a los estándares requeridos.

7.4 Respaldos de la base de datos del sistema

El sistema almacena información importante para el CNID. Por ese motivo, se realizarán respaldos cada mes por parte del administrador, asegurando de esta forma que la información esté segura en caso de alguna eventualidad.

De lo anterior, se menciona que la base de datos del sistema ha sido configurada para que crezca en capacidad de almacenamiento automáticamente. La única limitante es la capacidad del servidor.

7.5 Cambios en la lógica del sistema

Si un cambio se presentara en la lógica de programación del sistema, es posible realizar las modificaciones necesarias sin grandes complicaciones, ya que la parte de diseño y la lógica de programación se encuentran separadas, dando como resultado que la tarea de mantenimiento no se vuelva un gran problema.

Conclusiones

A partir de la creación del Sistema de Gestión Documental (SIGED) se ha logrado que el CNID ingrese al mundo de la información automatizada.

A lo largo del desarrollo de la aplicación se presentaron diversas situaciones que hicieron de este un proyecto más complejo, desde la labor de obtener información para llegar a las particularidades del manejo fiscal de la institución hasta el diseño, desarrollo e implementación del sistema, las cuales implicaron una serie de actividades que permitieron que el CNID creciera como organización y agilizará todos sus procesos de comprobación.

Crear un sistema para una institución pública implicó estudiar, involucrarse y familiarizarse a fondo con todo el personal involucrado en el proceso de comprobación a fin de entender y comprender el manejo de los procesos administrativos de la institución, de manera que los requerimientos del CNID se vean cubiertos y le permita llevar a cabo sus actividades de manera eficiente.

A lo largo de la construcción de este sistema y siguiendo la metodología de desarrollo de sistemas más adecuada para tal fin, se trabajó un desarrollo ordenado y profesional, siempre de la mano con el personal de la CONADE quienes son los usuarios finales y que cumplió con las expectativas y alcances planteados en la definición del problema.

En la parte práctica, el desarrollar e implementar una aplicación como SIGED exigió consultar, mejorar y asesorarse en cuanto al manejo de tecnologías como .NET y SQL Server por mencionar algunas de las diferentes herramientas utilizadas. El desarrollar la aplicación de forma modular permitió administrarla mejor, logrando con los módulos desarrollados un sistema hecho a la medida y satisfacer así las necesidades de la organización, permitiendo a su vez el crecimiento acorde a las necesidades del organismo y llevó a implementar otros programas de colaboración entre la CONADE y las entidades federativas a esta plataforma para continuar el proyecto de mejora gubernamental.

En la actualidad el SIGED se encuentra operando como proyecto piloto con algunos institutos del deporte. El sistema continúa su ciclo de vida en la etapa de planificación de las siguientes fases. Así mismo se le da mantenimiento y se realizan cambios en función de las necesidades del usuario.

Estos son solo algunos de los puntos que se pueden mencionar como parte de los beneficios que ha logrado la implementación de un sistema como SIGED y que ha contribuido a que una institución pública como la CONADE ofrezca mejores servicios.

Todo lo creado por los desarrolladores de este proyecto, ha sido en gran parte gracias a los conocimientos adquiridos en la Facultad de Ingeniería de la UNAM, Máxima Casa de Estudios de nuestro país.

El desarrollar este sistema nos ha dejado una gran experiencia profesional, proporcionándonos la confianza de que los conocimientos que adquirimos en los salones de clase nos permiten solucionar cualquier problema. Uno de los conocimientos de más valor fue el trabajar de la mano con los usuarios, entender su problemática y llegar a una solución óptima a través del punto de vista de la Ingeniería.

Al terminar este proyecto nos damos cuenta de todas las aportaciones personales que nos ha dejado. Ahora sabemos que somos capaces de realizar trabajos con mucha calidad y que podemos colaborar de manera responsable y profesional. Entendimos que el límite de nuestro potencial está determinado por nuestro propio esfuerzo.

Lo cierto es que la escuela no nos puede preparar para todos los casos a los que nos vamos a enfrentar en nuestra vida profesional, sin embargo nos damos cuenta que lo importante es saber aprovechar la base sólida que representan todos los conocimientos adquiridos en las aulas.

Anexo. Manual de instalación del sistema

Requerimientos del sistema

Los siguientes puntos presentan los requisitos mínimos de hardware y software para ejecutar el sistema

Software

- a) Microsoft SQL server 2005 Enterprise Edition o Posterior.
- b) Internet Information Service 6.0 o posterior

Para poder instalar Microsoft SQL server 2005 los requerimientos mínimos de un sistema operativo de 32 bits son:

COMPONENTE	REQUISITO
Software de internet	Se requiere SP1 de Microsoft, Internet Explorer 6.0 o posterior para la instalaciones de SQL Server 2005
Internet Information Service (IIS)	Internet Information Service (IIS) versión 6.0 o posterior.
ASP.NET	ASP.NET es necesario para Reporting Services. Durante la instalación de Reporting Services, el programa de instalación de SQL Server habilitará ASP.NET en el caso de que no se encuentre habilitado.

Instalaciones requeridas

El programa de instalación de SQL Server requiere Microsoft Windows Installer 3.1 o posterior y el SP1 de Microsoft Data Access Components (MDAC) 2.8 o posterior. Puede descargar SP1 de MDAC 2.8 del sitio web de Microsoft.

El programa de instalación de SQL Server instala los siguientes componentes de software requeridos por el producto:

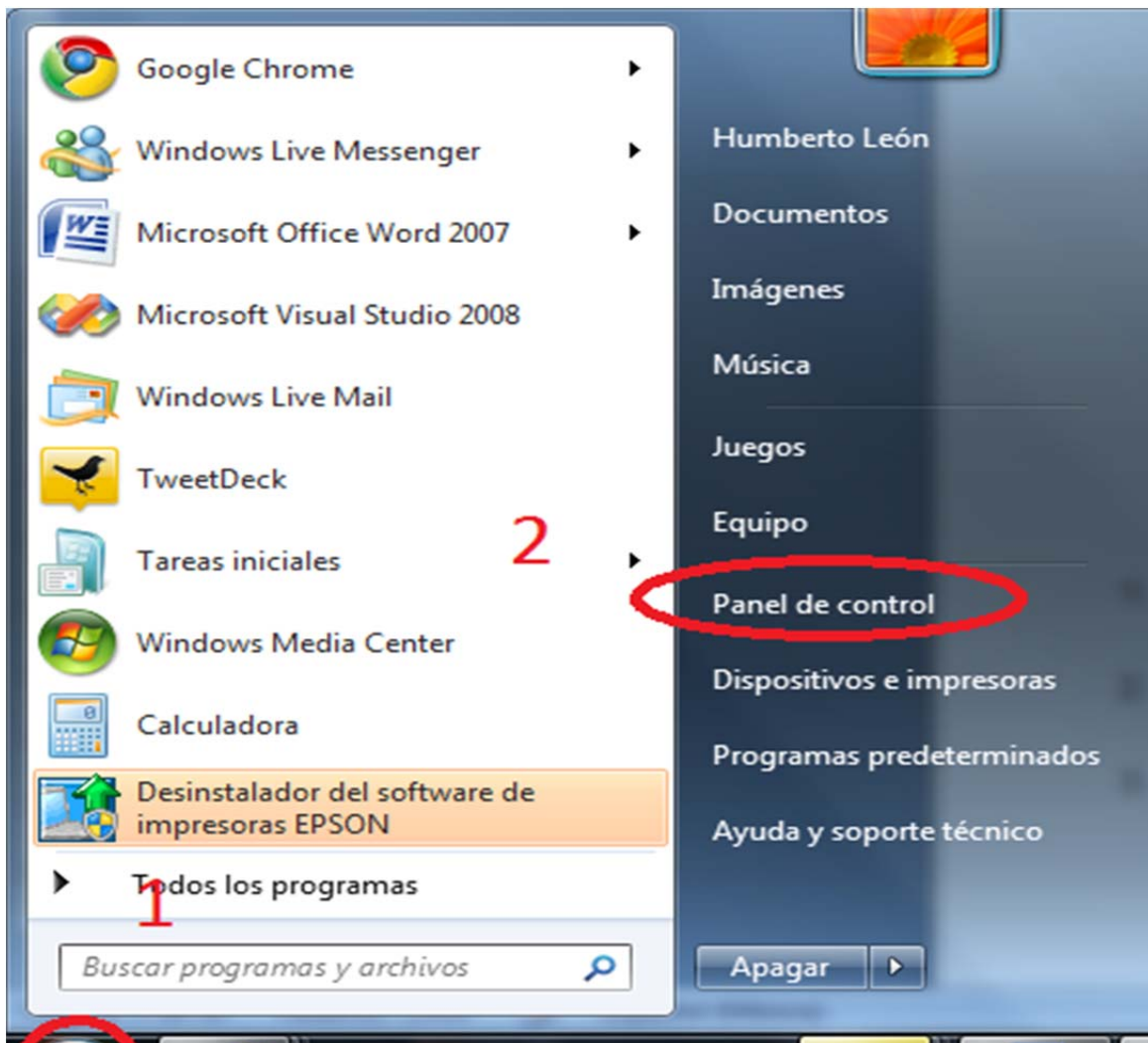
- a) Microsoft .NET Framework 2.0

- b) Microsoft SQL Server Native Client
- c) Archivos auxiliares de instalación de Microsoft SQL Server

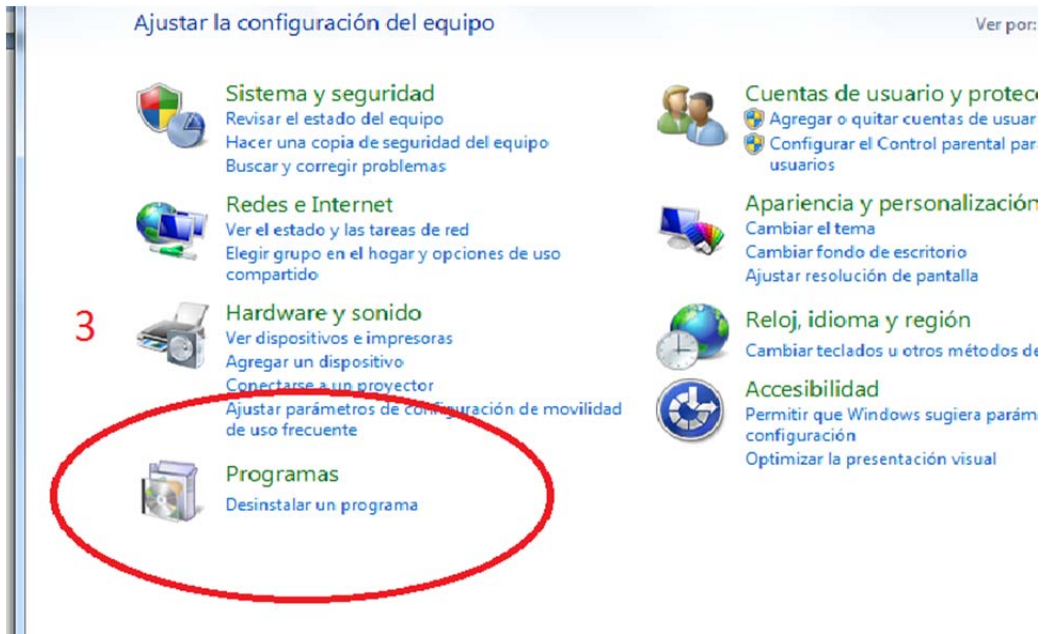
Instalación los servicios de Microsoft Internet Information Server (IIS)

Los pasos para instalar IIS en Windows 7 son los siguientes:

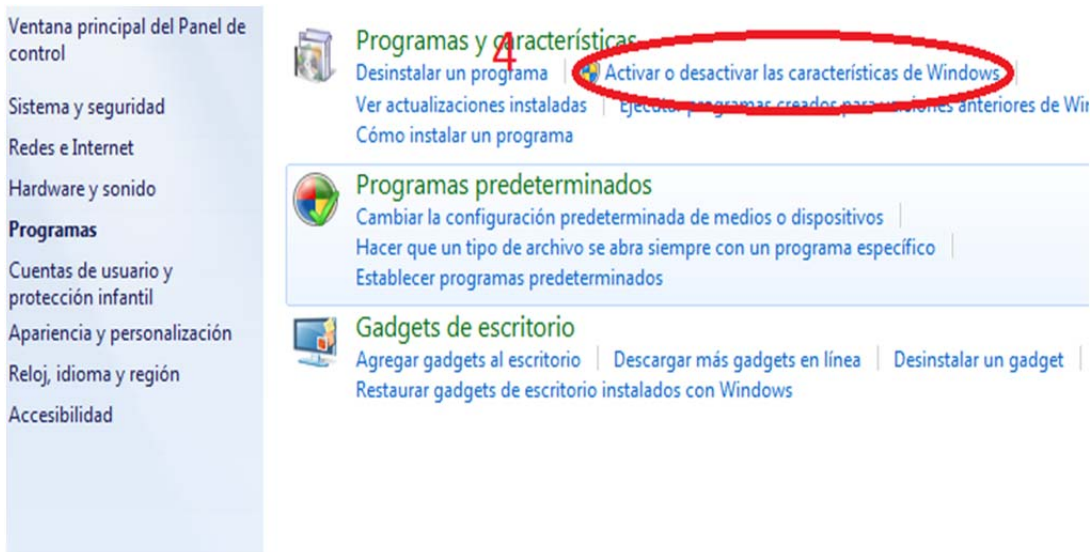
- a) Inicio
- b) Panel de control



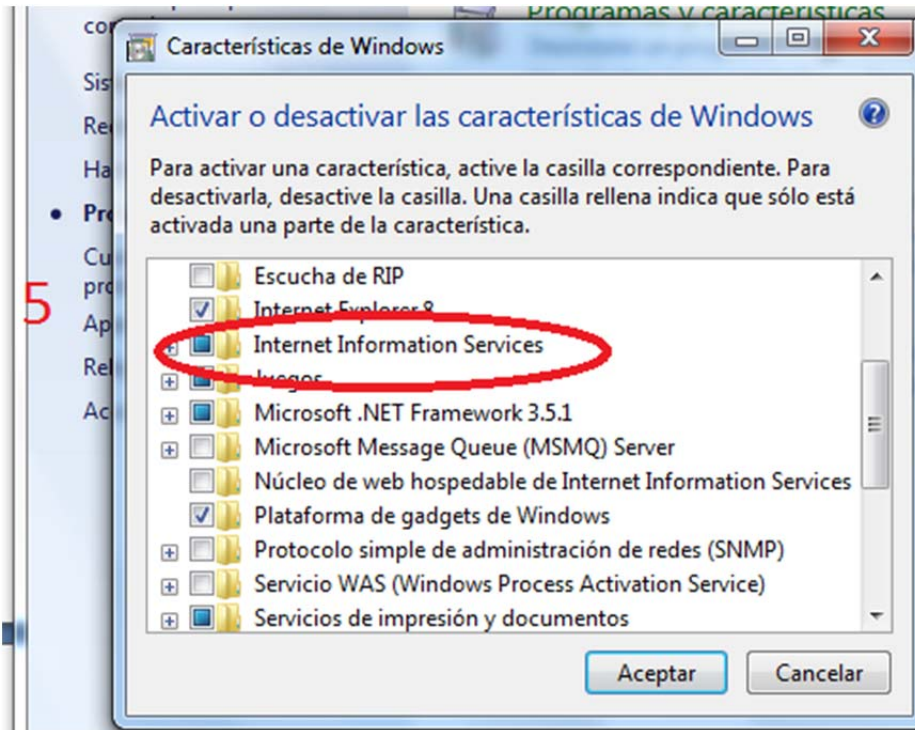
- c) Programas



d) Activar o desactivar características de Windows

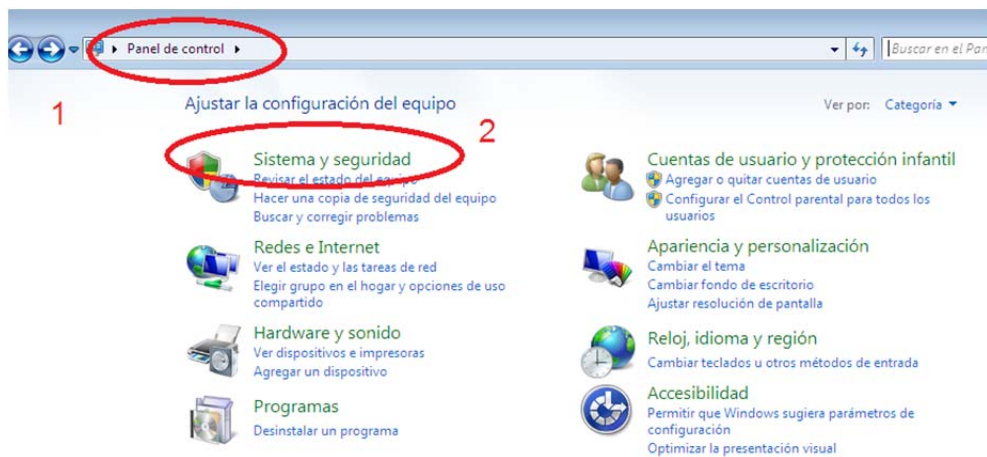


e) Se desplegará una lista, debe poner click en la pestaña de **Internet Information Services**

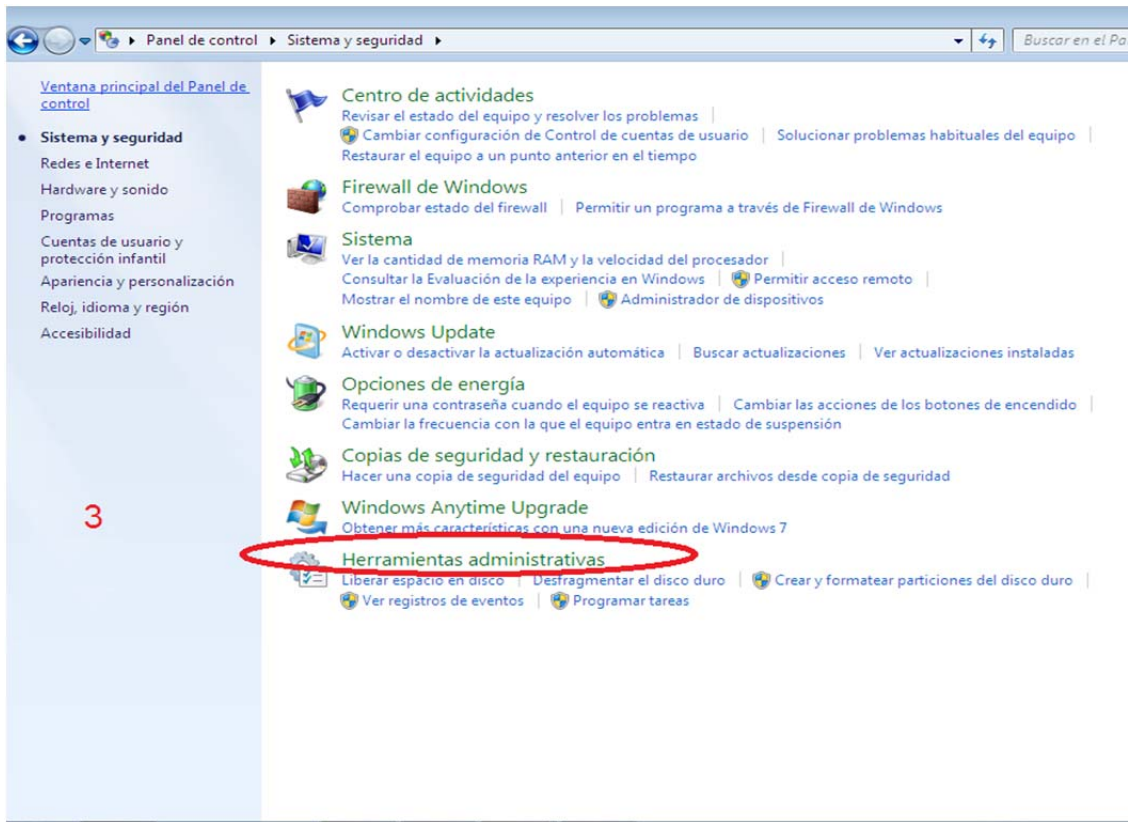


Para verificar si se ha instalado de forma correcta:

- a) Panel de control
- b) Sistema y seguridad



- c) Herramientas administrativas

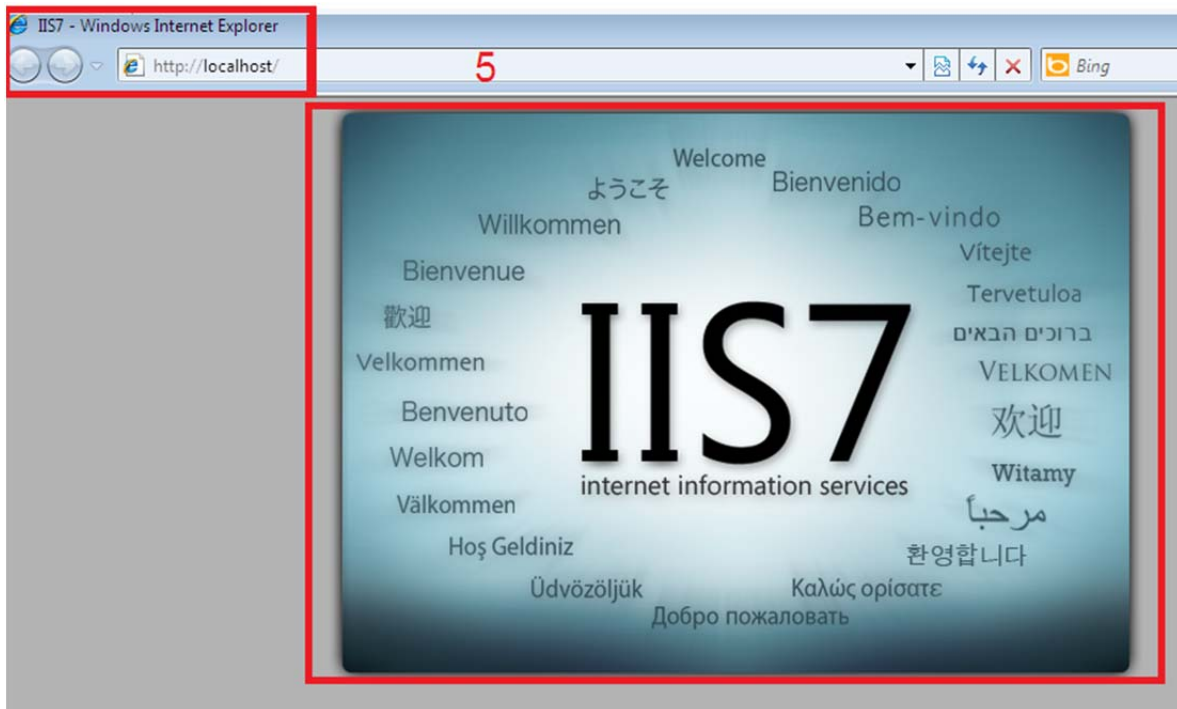


d) Aparecerá un cuadro de diálogo con los servicios del Win7, dentro de este debe localizar el **Administrador de IIS**.

Nombre	Fecha de modifica...	Tipo	Tamaño
Administración de equipos	13/07/2009 11:41 ...	Acceso directo	2 KB
Administración de impresiones	04/08/2010 01:06 ...	Acceso directo	2 KB
Administrador de Internet Information Services (IIS)	24/08/2010 09:18 a...	Acceso directo	2 KB
Configuración del sistema	13/07/2009 11:41 ...	Acceso directo	2 KB
Diagnóstico de memoria de Windows	13/07/2009 11:41 ...	Acceso directo	2 KB
Directiva de seguridad local	04/08/2010 01:06 ...	Acceso directo	2 KB
Firewall de Windows con seguridad avanzada	13/07/2009 11:41 ...	Acceso directo	2 KB
Iniciador iSCSI	13/07/2009 11:41 ...	Acceso directo	2 KB
Monitor de rendimiento	13/07/2009 11:41 ...	Acceso directo	2 KB
Orígenes de datos ODBC	13/07/2009 11:41 ...	Acceso directo	2 KB
Programador de tareas	13/07/2009 11:42 ...	Acceso directo	2 KB
Servicios de componentes	13/07/2009 11:46 ...	Acceso directo	2 KB
Servicios	13/07/2009 11:41 ...	Acceso directo	2 KB
Visor de eventos	13/07/2009 11:42 ...	Acceso directo	2 KB
Windows PowerShell Modules	13/07/2009 11:52 ...	Acceso directo	3 KB

4

e) Teclar en el navegador (<http://localhost/>), la página por default que se desplegará le indicará que está funcionando de manera correcta el IIS.



Para instalar IIS 7.0 en Windows server 2008

Este procedimiento se puede realizar mediante la interfaz

Para usar la interfaz de usuario

Haga clic en **Inicio**, elija **Herramientas administrativas** y, a continuación, haga clic en **Administrador del servidor**.

En **Resumen de funciones**, haga clic en **Agregar funciones**.

Utilice el **Asistente** para agregar funciones para agregar la función de servidor web.

Hardware

Requisitos netos del Sistema de Gestión Documental

Característica	Espacio en Disco Duro
Tamaño de la Aplicación.	14 MB
Tamaño de la Base de Datos Vacía	3.5 MB
Tamaño de la Base de Datos Poblada	25 MB
Tamaño Estimado de la Carpeta Digital (Depósito de Documentos PDF)	50 – 80 MB

Requisitos de espacio en disco duro para la instalación de SQL Server (32 y 64 bits)

Durante la instalación de SQL Server 2005, Windows Installer crea archivos temporales en la unidad del sistema. Antes de ejecutar el programa de instalación para instalar o actualizar SQL Server 2005, compruebe que dispone de 2,0 GB de espacio en disco en la unidad del sistema para estos archivos. Este requisito es aplicable incluso si instala todos los componentes de SQL Server en una unidad (del sistema) distinta de la predeterminada. Muchos de los archivos se instalarán en la unidad del sistema, que normalmente es C: Por ejemplo, los archivos de registro de la instalación se copiarán en la unidad del sistema, y es posible que requieran hasta 80 MB para una instalación independiente y más espacio para una instalación en un clúster de conmutación por error. Los requisitos de disco duro actuales dependen de la configuración del sistema y las aplicaciones y características que haya decidido instalar. En la siguiente tabla se muestran los requisitos de espacio en disco de los componentes de SQL Server 2005.

Características	Espacio en Disco Duro
Database Engine (Motor de base de datos) y archivos de datos, Réplica y Búsqueda de texto	280 MB
Analysis Services y archivos de datos	90 MB

Reporting Services y Administrador de informes	120 MB
Componentes del motor de Notification Services, componentes de cliente y componentes de reglas	50 MB
Integration Services	120 MB
Componentes de cliente	850 MB
Libros en pantalla de SQL Server y Libros en pantalla de SQL Server Compact Edition	240 MB
Ejemplos y bases de datos de ejemplo. Tenga en cuenta que, de forma predeterminada, los ejemplos y las bases de datos de ejemplo no se instalan.	410 MB

Requerimientos de sistemas operativos de 32 bits

SQL Server 2005(32 bits)	Tipo de procesador	Velocidad de procesador	Memoria(RAM)
Enterprise Edition	Procesador compatible con Pentium III o superior	Mínimo:600MHZ 1 GB en RAM o más	Minimo:512 MB Max:1 GB o superior

Requerimientos para sistemas operativos de 64 bits

SQL Server 2005(64 bits)	Tipo de procesador	Velocidad del procesador	Memoria RAM
Enterprise Edition	IA64 procesador Itanium superior	mínimo: IA64 mínimo: 1 GHz	IA64 mínimo: 512 MB IA64 recomendado: 1 GB o más

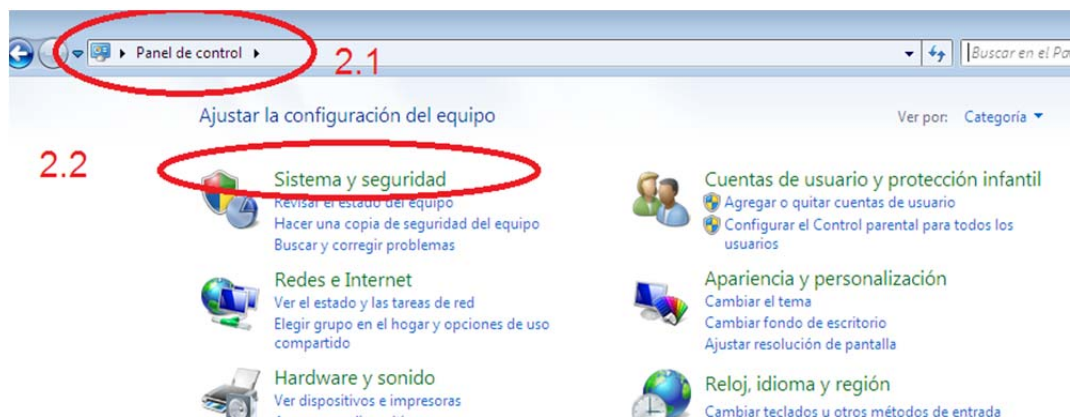
Instalación del sistema

Para realizar la instalación del sistema debe proceder de la siguiente manera:

Se deben colocar la totalidad de los archivos de la aplicación web en una sola carpeta cuya ubicación física en el servidor convenga a los fines del interesado.

Se deben seguir los siguientes pasos para configurar esa carpeta con el IIS

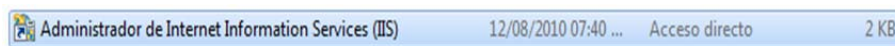
2.1 Abrir Panel de control



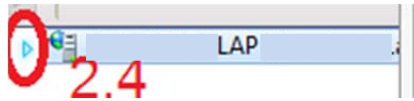
2.2 Acceder a la referencia de nombre **Sistema y Seguridad**, Seleccione la opción **Herramientas administrativas**



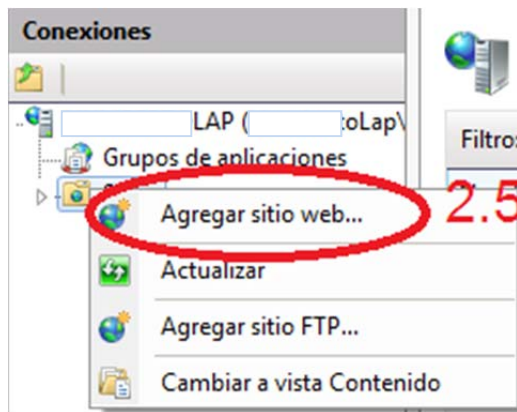
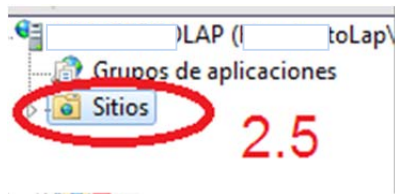
2.3 Entrar al **Administrador de Servicios de información de internet.(IIS)**



2.4 Dar click sobre la flecha



2.5. Dar click derecho sobre la carpeta **sitios**, luego debe seleccionar la opción de **Agregar sitio web**.

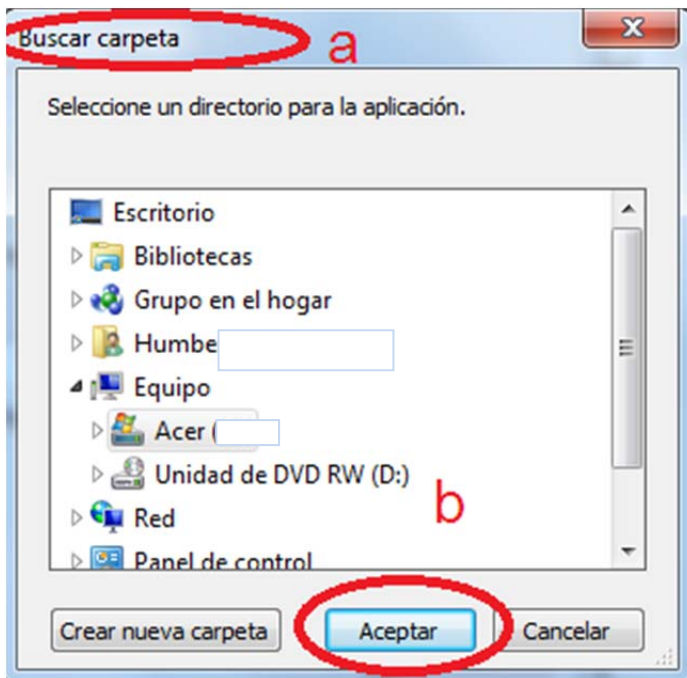
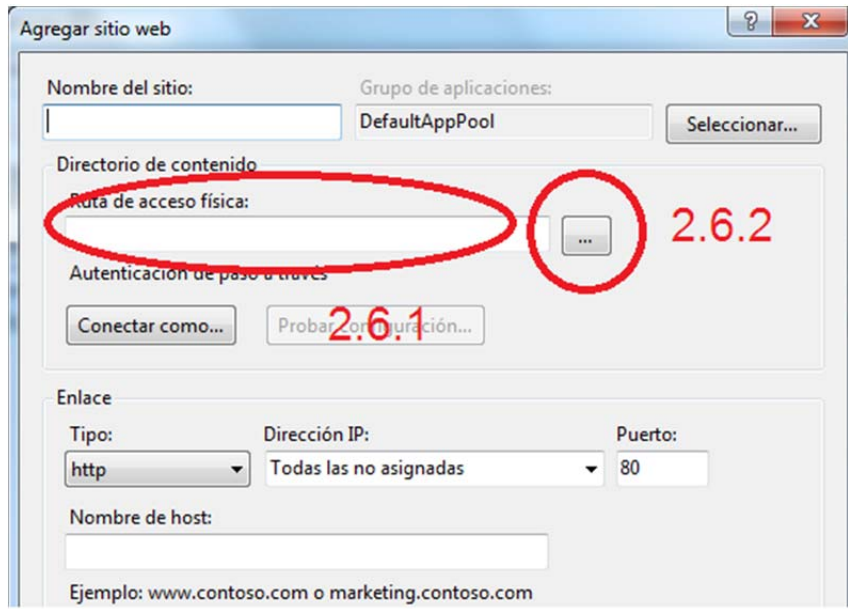


2.6 Es necesario especificar la ruta física de la estructura de archivos. Hay dos maneras de agregar el sitio web del sistema:

2.6.1. Puede acceder escribiendo en el cuadro de texto la ubicación donde se encuentra localizado el sistema.

2.6.2. Puede entrar mediante el botón cuyo identificador es unos puntos suspensivos, si eligió agregar el sitio web mediante esta forma:

- a) Buscar la carpeta del sistema, mediante el explorador identificado como **Buscar carpeta**.
- b) Una vez localizada la carpeta, dar clic en el botón aceptar



La aplicación tiene la siguiente estructura de carpetas:

Auditoria. Carpeta donde se guardan los archivos de interface de las bitácoras del sistema.

Comprobación. Carpeta que almacena los archivos interface involucrados en el proceso de asignación presupuestal y su posterior comprobación.

controlUsuario. Los archivos que determinan la interface y operación de las pantallas de inicio de sesión y administración de usuarios del sistema se encuentran aquí.

Css. Carpeta donde están almacenadas las hojas de estilo responsables de la apariencia de toda la aplicación

Digital. Carpeta contenedor de los archivos digitales cargados a través de la aplicación en los módulos de documentación y comprobación.

Documentacion. Carpeta responsable de alojar los archivos de interface y comportamiento del módulo de carga de documentación.

Imágenes. Carpeta contenedor de imágenes a utilizar en las diferentes áreas del sistema.

Informes. Carpeta donde se alojan los archivos de interface y comportamiento del módulo de Informes.

Interfazusuario. Carpeta donde se alojan los archivos de apariencia y contenido de las secciones de acceso público, no controlado por el módulo de control de acceso.

Inventario. Carpeta que almacena los archivos de la sección de carga y administración de inventario.

Js. Carpeta que almacena archivos con scripts ejecutables requeridos para la operación del sistema.

Mensajes. Carpeta contenedora de los archivos de interface y operación del buzón de mensajes.

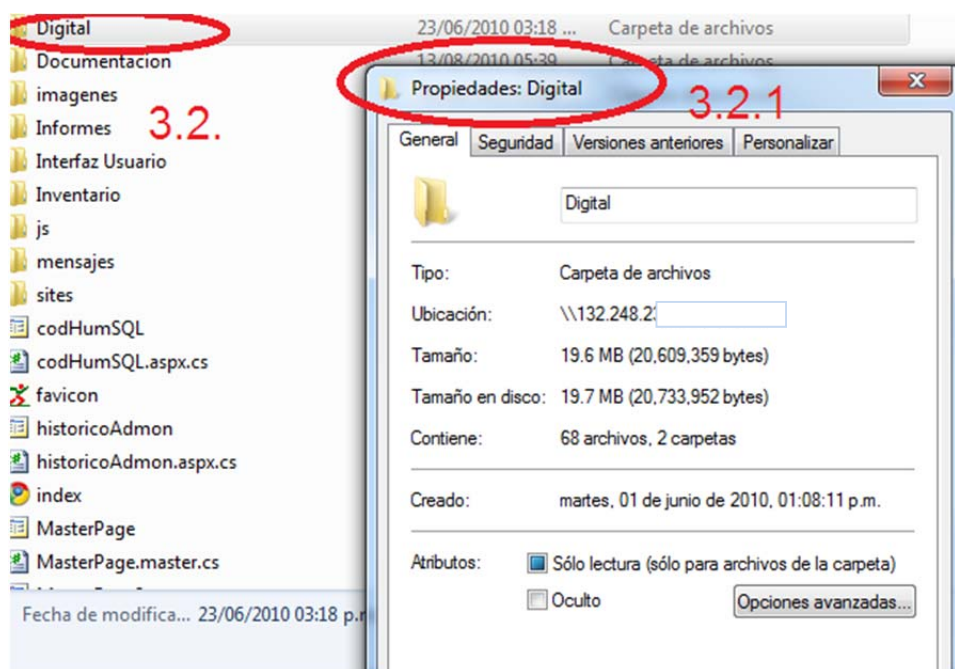
Sites. Los archivos de definición del sitio utilizados para la creación de tablas de contenido están definidos dentro de esta carpeta.

Para la correcta ejecución de la aplicación y, en particular, para poder habilitar la función de carga de archivos al servidor desde el cliente, se debe realizar la siguiente configuración de permisos de escritura de la carpeta **Digital**.

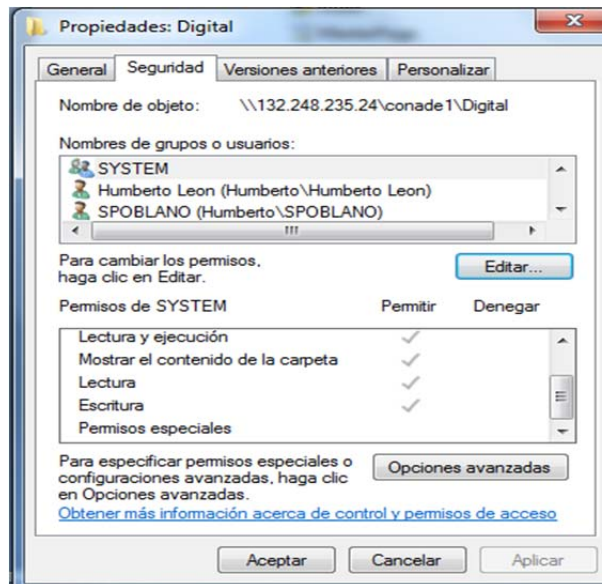
3.1 Acceda a la dirección donde está instalada la carpeta del sistema

3.2 De clic sobre la carpeta Digital.

3.2.1. Se desplegara una ventana cuyo identificador es Propiedades digitales



3.3 Acceder a la carpeta a la pestaña de seguridad y abrir la edición de usuarios.



3.4 Buscar, habilitar y otorgar permisos de escritura al grupo de usuarios IIS_IUSRS para esta carpeta.

Requerimientos mínimos de información en la base

Las siguientes tablas son requerimientos mínimos de información que necesita la base de datos para su correcto funcionamiento.

Vale la pena destacar que las abreviaturas utilizadas en la base son de vital importancia ya que funcionan como medio de control ante posibles disparidades de las llaves primarias, por lo que su valor no debe verse modificado, sin importar el valor de los demás registros.

cAcción

Acc_ID	Acc_Rubrica	Acc_Descripción
1	A	AGREGAR
2	B	BORRAR
3	C	CAMBIAR
4	D	ENVIAR
5	E	APROBAR
7	F	RECHAZAR

cPeriodo

Per_ID	Per_Nombre	Per_ValorMes	ID_status	Per_Secuencia
1	Anual	12	3	1
2	Primer Trimestre	3	3	1
3	Segundo Trimestre	3	3	2
4	Tercer Trimestre	3	3	3
5	Cuarto Trimestre	3	3	4

cTipoDocumento

Tid_ID	Per_ID	ID_Permisio	Tid_Nombre	Tid_Abreviación	IDStatus
1	1	6	Anexo al convenio	AC	3
2	1	6	Anexo Técnico	AT	3
3	1	6	Acta de instalación	AI	3
4	1	6	Ficha Técnica	FT	3
5	1	5	Recibo Fiscal	RF	3
6	1	6	Adem dum	AD	3

cConceptoInforme

Coi_ID	Tii_Nombre	IDstatus	Coi_ID
11	Infraestructura CEID	3	INFREP
12	Equipo Telecomunicaciones	3	TELCOMEPEP
13	Recursos Humanos	3	RHEP
14	Infraestructura CEID	3	RENAPA
15	Equipo Telecomunicaciones	3	PDPA
16	Recursos Humanos	3	RHPA
17	RENADE	3	RENAPA
18	Programas de difusión	3	PDPA
19	RENADE	3	RENAEP
20	Censo de infraestructura	3	CENEP

21	Pilotos	3	PILEP
22	Objetivos	3	OBJPA
23	Funciones Servicios	3	FYSPA
24	Censo de infraestructura	3	CENPA
25	Consumibles	3	CONSPA

cPermiso

IDPermiso	Abreviatura	Tipo	Descripción	FechaAlta	FechaUltimaActualizacion	IDStatus
2	ROOT	ROOT1	USUARIO SUPER ADMIN			1
3	FICE	FICE	Finanzas CEID			3
4	ADCE	ADCE	Titular CEID			3
5	DICE	DICE	Director de Instituto CEID			3
6	ADCN	ADCN	Administrador CNID			3
7	AUCN	AUCN	Administrador de Usuarios			3
8	AUCN	AUCN	Administrador de Usuarios			1
9	AUDI	AUDI	Administrador de Bitácoras			3

cTipoInforme

Tii_ID	Per_ID	IDPermiso	Tii_Nombre	Tii_Abreviación	Tii_FechaLimite	IDStatus
12	1	4	Evaluación Previa			
13	1	4	Plan Anual			
14	1	4	Técnicos			
15	2	4	Técnicos			
16	3	4	Técnicos			
17	4	4	Técnicos			
18	5	4	Técnicos			

29	1	4	Actividades		
30	2	4	Actividades		
31	3	4	Actividades		
32	4	4	Actividades		
33	5	4	Actividades		
34	1	4	Registros Metas	y	
35	2	4	Registros Metas	y	
36	3	4	Registros Metas	y	
37	4	4	Registros Metas	y	
38	5	4	Registros Metas	y	

cConceptoComprobacion

Coc_ID	Per_ID	Coc_Nombre	IDStatus	IDPermiso	Coc_Abreviatura
1	1	Computadoras	3	3	COMP
2	1	Hardware	3	3	HARD
3	1	Consumibles	3	3	CONSU
4	1	Internet/Teléfono	3	3	INTERNET
5	1	Periféricos	3	3	PER
6	1	Recursos Humanos	3	3	RH
8	1	Software	3	3	SOFT
9	1	Promoción/Difusión	3	3	PROMO
10	1	Reintegro	3	3	REINT
11	1	Uno al Millar	3	3	UAM

cStatus

IDStatus	Nombre	Abreviatura	Descripcion	FecAlta	FecUltimaActualizacion
1	ELIMINADO	E			
2	PENDIENTE	P			
3	ACTIVO	A			
4	FIJO	F			
8	BORRAR	B	NULL		
16	PENDIENTE DIRECTOR	PD	PENDIENTE DIRECTOR		
17	APROBADO DIRECTOR	AD	APROBADO DIRECTOR		

18	PENDIENTE CNID	PC	PENDIENTE CNID
19	APROBADO CNID	AC	APROBADO CNID
20	RECHAZADO	R	RECHAZADO
21	Leído	L	Leído
22	No Leído	NL	No Leído

Glosario

ACID. *Atomicity, Consistency, Isolation, Durability* (atomicidad, consistencia, aislamiento, durabilidad). Características de una base de datos para llevar a cabo transacciones.

ADO.NET. *ActiveX Data Object for .NET*. Conjunto de elementos contenidos en el framework de Microsoft para interactuar con datos, ya sea en ambientes conectados o desconectados.

ASP. *Active Server Pages*, son páginas web generadas de forma dinámica y que se ejecutan sobre un servidor IIS.

ASP.NET. Evolución de la tecnología ASP que facilita la programación por capas y hace uso de la programación orientada a objetos.

BCL. *Base Class Library*, librería estándar de uso común para los lenguajes basados en la tecnología .NET, contiene funciones como lectura de archivos, conexión a base de datos, manipulación de XML, etc.

BSD. *Berkeley Software Distribution*, sistema operativo basado en Unix.

C. Lenguaje de programación de nivel medio, típicamente usado en programación estructurada y permite la integración con otros elementos como lenguaje ensamblador o control de periféricos.

C#. Lenguaje de programación basado en C, es parte de la plataforma .NET y es orientado a objetos.

CASE. *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora. Herramientas que facilitan la elaboración de un producto de software en sus diferentes etapas, como pueden ser la planeación, cálculo de costos, generación de código, pruebas, etc.

CEID. Centro Estatal de Información y Documentación, organismo perteneciente a una entidad deportiva que fomenta, difunde y apoya al deporte mediante las tecnologías de información.

Cliente. Aplicación o equipo de cómputo que forma parte de una red y hace uso de servicios proporcionados por un servidor.

CLR. *Common Language Runtime*, colección de datos y funciones comunes a todos los lenguajes basados en .NET, ofrece servicios como recolección de basura e información de los ensamblados.

CNID. Centro Nacional de Información y Documentación de Cultura Física y Deporte, organismo rector de los CEID dependiente de la Dirección de Planeación y Tecnologías de la Información de la CONADE.

COM. *Component Object Model*, plataforma para la creación de componentes de software independiente del entorno. Es un medio común para realizar acciones sobre los objetos y permite su eliminación cuando se pierden las referencias a este.

CONADE. Comisión Nacional de Cultura Física y Deporte, organismo gubernamental encargado de fomentar y promover el deporte, la cultura física y la recreación.

CURP. Clave Única de Registro de Población, conjunto de datos alfanuméricos asignado a los mexicanos, se componen a partir del nombre completo, la fecha y lugar de nacimiento, el género y dos caracteres para evitar duplicidad.

DB2. Sistema de gestión de base de datos desarrollado por IBM y que tiene soporte nativo para XML.

DBMS. *Data Base Manager System*, Sistema de Gestión de Bases de Datos, tipo de software capaz de manejar grandes volúmenes de datos de una manera clara, sencilla y ordenada.

DCU. Diagrama de Caso de Uso, herramienta que ayuda a representar el comportamiento de un sistema y su relación con agentes externos como el usuario.

DER. Diagrama Entidad Relación, herramienta para el modelado de datos de un sistema de información, se compone de entidades con atributos y las relaciones que hay entre ellas.

DFD. Diagrama de Flujo de Datos, herramienta que muestra de forma gráfica la secuencia de procesos en un sistema de información.

DGADyR. Dirección General de Actividades Deportivas y Recreativas, dependencia de la UNAM que trata los asuntos relacionados con la práctica del deporte y el empleo del tiempo libre.

DHTML. *Dynamic HTML*, HTML Dinámico, se refiere al conjunto de técnicas que permiten crear páginas web con las que el usuario puede interactuar, se basan en el uso de HTML, hojas de estilo y algún lenguaje interpretado del lado del cliente como puede ser JavaScript.

DLL. *Dynamic - Link Library*, Biblioteca de Enlace Dinámico, archivo que contiene código ejecutable usado por Windows y que lo carga cuando un programa lo solicita.

DML. *Data Manipulation Language*, Lenguaje de Manipulación de Datos, lenguaje proporcionado por el manejador de base de datos para llevar a cabo tareas como la manipulación, edición e inserción de información.

DOM. *Document Object Model*, Modelo de Objetos del Documento, es una interface de programación de aplicaciones que permite la modificación de contenido, estructura y estilo de documentos HTML y XML.

DRA. Desarrollo Rápido de Aplicaciones, metodología de desarrollo de software que hace uso de construcción de prototipos y herramientas CASE para desarrollar un producto en tiempos considerablemente cortos.

DTE. Diagrama de Transición de Estado. Es una herramienta de modelado que sirve para describir el comportamiento requerido de los sistemas de tiempo real, al igual que la porción de la interfaz humana que la mayoría de los sistemas en línea tiene. El DTE se usa para

desarrollar un modelo esencial del sistema, es decir, un modelo de cómo se comportaría el sistema si hubiera la tecnología perfecta (ideal). El DTE se puede relacionar con el DFD.

FAQ. *Frequently Asked Questions*, Preguntas Frecuentes, sección de un sitio web donde se publican las dudas más comunes para los usuarios.

FQL. *Facebook Query Language*, lenguaje de sentencias para Facebook.

FTP. *File Transfer Protocol*, Protocolo de Transferencia de Archivos, protocolo de red para la transferencia de archivos basado en arquitectura cliente - servidor, normalmente hace uso del puerto 21.

GNU. Proyecto que busca la creación de un sistema operativo completamente libre.

Gobierno Electrónico. Se refiere a los procesos y estructuras creadas para la oferta electrónica de los servicios gubernamentales.

GUI. *Graphic User Interface*, Interfaz Gráfica de Usuario, se refiere a la parte visual de una aplicación que permite al usuario interactuar con ella mediante objetos gráficos.

Hardware. Corresponde a todas las partes tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

HTML. *HyperText Markup Language*, Lenguaje de Marcado de Hipertexto, lenguaje basado en el uso de etiquetas que permite describir la estructura y el contenido de una página web.

IEC. *International Electrotechnical Commission*, Comisión Electrotécnica Internacional. Organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. Numerosas normas se desarrollan conjuntamente con la ISO (normas ISO/IEC).

IIS. *Internet Information Services*, servidor web de Microsoft capaz de procesar páginas de tipo ASP y ASP.NET.

IMSS. Instituto Mexicano del Seguro Social, institución de seguridad social considerada por la CONADE como una entidad deportiva.

IP. *Internet Protocol*. Conjunto de datos numéricos o alfanuméricos que identifica a un equipo de cómputo dentro de una red, se dividen en públicas y privadas.

IPN. Instituto Politécnico Nacional, institución educativa pública y considerada por la CONADE como una entidad deportiva.

ISAPI. *Internet Server Application Programming Interface*, Interfaz de Programación de Aplicaciones para Servidor de Internet, API para aplicaciones basadas en el servidor web de Microsoft.

ISO. *International Organization for Standardization*, Organización Internacional de Normalización, organismo encargado de promover normas internacionales para productos y procesos.

IWAMAdmin. Interface que hereda métodos de la plataforma COM, sirve para la administración de aplicaciones.

J#. Lenguaje de programación con una sintaxis muy similar a Java pero que corre en entornos .NET.

Java. Lenguaje de programación orientado a objetos, multiplataforma y permite crear aplicaciones tanto para escritorio como para web.

JavaScript. Lenguaje de programación interpretado que permite la creación de páginas dinámicas, posee una sintaxis muy parecida a C.

JScript. Lenguaje de programación interpretado propiedad de Microsoft, de estructura idéntica a JavaScript pero cambia en nombre por cuestión de derechos de autor.

JSP. *Java Server Pages*, tecnología basada en Java para la creación de contenido dinámico para páginas web.

Linux. Implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo.

Mac OS. Sistema operativo de Apple para sus equipos de cómputo, en un principio era completamente desarrollado por Apple, en una etapa posterior se basó en UNIX.

MS SQL Server. Manejador de base de datos relacional de Microsoft, hace uso de los lenguajes T - SQL y ANSI SQL.

OMT. *Object-Modeling Technique*, Técnica de Modelado de Objetos, es un lenguaje para el diseño y modelado de software.

PDA. *Personal Digital Assistant*, Asistente Digital Personal, dispositivo portátil capaz de realizar tareas como la creación y lectura de documentos, navegar en Internet, reproducir archivos multimedia, administrar contactos y recordatorios, entre otras tareas.

PDF. *Portable Document Format*, Formato de Documento Portátil, formato de almacenamiento de documentos multiplataforma, no permite que se modifique el contenido ni la estructura del archivo.

Perl. Lenguaje de programación libre basado en C y Shell, ampliamente utilizado en aplicaciones web.

PHP. *PHP Hypertext Preprocessor*, lenguaje de programación libre orientado al desarrollo de aplicaciones web dinámicas, puede programarse de forma estructurada u orientada a objetos.

POO. Programación Orientada a Objetos, paradigma de programación basado en la abstracción de objetos reales y se extraen sus características y comportamiento que a su vez son agrupados en una clase.

Puerto. Interfaz, ya sea física o lógica, a través de la cual se pueden enviar y recibir datos.

Python. Lenguaje de programación de código abierto, multiplataforma y multiparadigma.

QBE. *Query By Example*, es el primer un lenguaje de tipo gráfico para consultas a base de datos, originalmente solo soportaba consulta de datos, después aceptaba operaciones de inserción, actualización y eliminación.

QL. *Query Language*, Lenguaje de Consulta, se refiere a un lenguaje informático para comunicarse con sistemas de bases de datos.

RDBMS. *Relational Data Base Management System*, es un sistema de administración de base de datos basado en el modelo relacional.

RENADE. Registro Nacional de Cultura Física y Deporte, sistema de información del gobierno federal donde se almacenan datos relacionados con la difusión y apoyo del deporte de cada entidad.

Ruby. Lenguaje de programación interpretado, multiparadigma y con licencia de software libre.

Script. Programa relativamente simple e interpretado en la mayoría de los casos, tiene como objetivo combinar componentes, interactuar con el sistema operativo o con el usuario.

Servidor. Equipo de cómputo que forma parte de una red y proporciona servicios a los clientes.

Servlet. Es un programa que se ejecuta del lado del servidor web y genera páginas dinámicas a través del lenguaje Java.

SGDB (SGBD). Sistema de Gestión de Base de Datos, tipo de aplicación informática cuyo objetivo es manejar grandes cantidades de información de una manera clara, sencilla y ordenada.

SIGED. Sistema de Gestión Documental, proyecto desarrollado por el CEID UNAM en conjunto con la CONADE para eficientes la administración y comprobación de recursos de las entidades deportivas.

SOAP. *Simple Object Access Protocol*, Protocolo de Acceso Simple a Objetos, protocolo para comunicar objetos de procesos distintos haciendo uso de XML.

Software. Equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

Solaris. Sistema operativo de tipo Unix desarrollado por Sun, usado en servidores y estaciones de trabajo.

SQL. *Structured Query Language*, Lenguaje de Consulta Estructurado, lenguaje de acceso a bases de datos relacionales haciendo uso del álgebra y el cálculo relacional.

Symbian. Sistema operativo usado en dispositivos móviles, fue desarrollado por varias empresas como Sony Ericsson, Samsung, LG, Motorola y principalmente Nokia.

TCP/IP. Conjunto de protocolos de Internet, se compone del Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP). Empleado para comunicar diferentes equipos sin importar la arquitectura o sistema operativo.

TI. Tecnologías de la Información, conjunto de técnicas y elementos que se ocupan del tratamiento y difusión de los datos, se enfocan principalmente en Internet, informática y telecomunicaciones.

UML. *Unified Modeling Language*, Lenguaje Unificado de Modelado, es un lenguaje gráfico que permite la planeación, desarrollo y documentación de un proyecto de software. Está basado en el uso de diagramas.

UNAM. Universidad Nacional Autónoma de México, institución educativa más importante de nuestro país, es considerada por la CONADE como una entidad deportiva.

Unicode. Estándar de codificación de caracteres para facilitar la escritura en diferentes lenguajes, la última versión está compuesta por más de 109,000 caracteres.

UNIX. Sistema operativo multitarea y multiusuario, escrito inicialmente por un grupo de desarrolladores de los Laboratorios Bell de AT&T.

URI. *Uniform Resource Identifier*, Identificador Uniforme de Recurso, es una cadena de caracteres que identifica un recurso como página, correo electrónico, documento, entre otros. Se compone de esquema, autoridad, ruta, consulta y fragmento.

URL. *Uniform Resource Locator*, Localizador Uniforme de Recursos, es una cadena de caracteres para localizar recursos de Internet. Comúnmente se compone de esquema, máquina, directorio y archivo.

VB.NET. Visual Basic .NET, lenguaje de programación orientado a objetos basado en Visual Basic.

W3C. *World Wide Web Consortium*, organización internacional que busca estandarizar los desarrollos en ambientes web.

Web. Sistema de documentos de hipertexto entrelazados y accesibles por medio de Internet.

Web 2.0. Término relacionado con la interacción del usuario dentro de un sitio, como pueden ser redes sociales, blogs, wikis, etc.

Windows. Sistema operativo de Microsoft actualmente usado en equipos cliente, servidores y más recientemente en dispositivos móviles.

XML. *eXtensible Markup Language*, Lenguaje de Marcas Extensible, lenguaje desarrollado por el W3C que busca facilitar el intercambio de información estructurada entre diferentes plataformas.

XSS. *Cross - Site Scripting*, vulnerabilidad de seguridad informática encontrada mayormente en aplicaciones web y que consiste en la inyección de scripts en las páginas.

Bibliografía

1. Liberty, J. y Hurwitz, D., Programación con ASP.NET 2.0, Creación de servicios y aplicaciones web, Ediciones Anaya Multimedia, Madrid 2007.
2. Solórzano P., F., Introducción a la Programación Estructurada y al Lenguaje C, Facultad de Ingeniería de la UNAM, México 1995.
3. Zacker, C. y Rourke, J., PC Hardware, Manual de referencia, McGraw-Hill, Madrid 2001.
4. Jimenez N., A. y otros, Sistema de Información Deportiva, Tesis para obtener el Título de Ingeniero en Computación, Facultad de Ingeniería, UNAM, 2002.
5. Gudiño S., Diana, Sistema de Administración e Inventario para una Distribuidora Comercial, Tesis para obtener el Título de Ingeniero en Computación, Facultad de Ingeniería, UNAM, 2009.
6. IBM, Guía rápida de iniciación para servidores DB2, DB2 Versión 9.5 for Linux, Unix and Windows, EUA, 2007.
7. Pavón P., Jacobo, Creación de un portal con PHP y MySQL, 3ª Edición, RA-MA Editorial, México 2007.
8. Ceballos, F., Java 2, Curso de Programación, 3ª Edición, Alfaomega Grupo Editor, 2006.
9. Schmuller, J., Aprendiendo UML en 24 Horas, Prentice Hall, 2003.
10. Puvvala, Jawahar, Pota, Alok, .NET for Java Developers: Migrating to C#, Addison Wesley Professional, 2009.
11. Fairley, F., "Ingeniería de Software", 1a ed., Mc Graw Hill, México, 1992
12. Hanse, G., "Database Management an Design", 2a ed., Prentice-Hall Inc., Upper Saddle River, New Jersey, 1996
13. Shepard, G., "Microsoft ASP .NET 3.5", 1a ed., Microsoft Press, Redmon Washington, 2008
14. Singh, M., "Service-Oriented Computing Semantic, Process, Agents", 1a ed., John Wiley & Sons Ltd, West Sussex, Englands, 2005.
15. Prellsman, R., Ingeniería del Software, un enfoque práctico, 6ª. Edición, McGraw-Hill, México 2007.
16. Yourdon, E., Constantine, L., Structured Desing, Prentice-Hall, 1979.
17. Elmasri, R., Navathe, S., Fundamentals of Database Systems, 3a Edición, Addison-Wesley, 2000.
18. Freeman, A., Jones, A., Programing .NET Security, 1a Edición, Editorial O'Reilly, 2003.
19. Lawrence, S., Ingeniería del Software, Teoría y Práctica, 1ª Edición, Pearson Education, Buenos Aires, 2002.
20. Schach, S., Ingeniería de Software Clásica y Orientada a Objetos, 6ª Edición, McGraw-Hill, México 2006.
21. Pender, T., UML Biblie, Wiley Publishing, Indianapolis, 2003.

22. Davis, J., Lee, T., Microsoft Windows Server 2003, protocolos y servicios TCP/IP, McGraw-Hill, Madrid 2003.
23. Richard E. Fairley, Ingeniería de Software, 1ª Edición, Mc Graw Hill, México 1992.
24. Gary W. Hansen, Database Management an Design, 2ª Edición, Prentice-Hall Inc., Upper Saddle River, New Jersey 1996.
25. George Shepard, Microsoft ASP .NET 3.5, 1ª Edición, Microsoft Press, Redmon Washington 2008.
26. Munindar P. Singh, Michael N. Huhns, Service-Oriented Computing Semantic, Process, Agents, 1ª Edición, John Wiley & Sons Ltd, West Sussex, Englands 2005.

Referencias Web

27. PostgreSQL, About, <http://www.postgresql.org/about/> , 2011.
28. PostgreSQL, About Matrix, <http://www.postgresql.org/about/featurematrix> , 2011.
29. Microsoft SQLServer, Compare Microsoft SQL Server Editions, <http://www.microsoft.com/sqlserver/en/us/product-info/compare.aspx> , 2011.
30. MySQL, The Main Features of MySQL,
31. <http://dev.mysql.com/doc/refman/5.5/en/features.html> , 2011.
32. Wikipedia, Comparison of relational database management systems, http://en.wikipedia.org/wiki/Comparison_of_databases , 2011.
33. Ruíz Berenice, Pruebas de Usabilidad en Sitios Web, <http://www.sg.com.mx/content/view/1088> , 2010.
34. La Web del Programador, Cursos y Manuales, <http://www.lawebdelprogramador.com/> , 2011.
35. Centro de Información de RedUNAM, Home, <http://www.nic.unam.mx/>, 2010.
36. Microsoft MSDN, Desarrollo .NET, <http://msdn.microsoft.com/es-es/library/aa139615.aspx>, 2011.
37. Microsoft MSDN, Technical Reference (Database Engine), <http://msdn.microsoft.com/en-us/library/bb500275.aspx>, 2011.
38. MySQL, MySQL 5.0 Reference Manual, <http://dev.mysql.com/doc/refman/5.0/es/index.html>, 2011.
39. PostgreSQL, PostgreSQL 8.4.8 Documentation, <http://www.postgresql.org/docs/8.4/static/index.html>, 2009.
40. Oracle, Oracle Database Documentation, <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>, 2011.
41. IBM, Documentación en PDF de IBM DB2 Versión 9.5, ftp://ftp.software.ibm.com/ps/products/db2/info/vr95/pdf/es_ES/db2fe.htm, 2010.