



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN**

**SISTEMA DE ADMINISTRACIÓN
DOCUMENTAL (SAD).**

T E S I N A

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN**

**PRESENTA:
DIANA ELGUERA ALMAZÁN**

DIRECTOR DE TESINA: M.I. JUAN CARLOS ROA BEIZA

Noviembre 2015.



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.

Agradecimientos con todo mi cariño

A mi mamá Olga.

De todo corazón agradecerle por ser una ejemplo en mi vida, aunque ya no se encuentra en este mundo sé que aún me guía (mi angelito).

A mis hijos:

Daniel y Hannah porque ellos son el motor de mi vida para seguir progresando y enfrentarme a nuevos retos.

A mi Esposo

Martín, por su paciencia y su incondicional apoyo muchas gracias. (tqm).

A mi Hermanas y mi tía

Adriana, Olga y mi tía Linda, que son mi apoyo en todo momento.

A la UNAM

Que siempre estaré agradecida por mi formación.

Al Maestro Juan Carlos Roa. Beiza

Porque no sólo ha sido un asesor sino un magnifico maestro en mi vida. Te agradezco, por tus palabras certeras para continuar en la culminación de este trabajo.

Mil gracias.

A mis compañeros de equipo del PAT

- | | |
|-----------------------------|-----------------------------|
| * Álvaro Hernández Márquez | número de cuenta 305069562 |
| * Miguel Álvarez Solís | número de cuenta 306135204 |
| * Gustavo Saldivar Quiñonez | número de cuenta 085602698 |
| * Gustavo Jiménez Pérez | número de cuenta 081162400. |

Muchas gracias

Agradecimientos

A mi madre:

Estoy muy agradecido por todo el apoyo que siempre me ha brindado, permitiéndome llegar hasta donde hoy me encuentro, gracias por preocuparse por mí todos estos años, sin su enorme dedicación diaria durante mi niñez lo más probable es que hoy ni siquiera me encontrara vivo, gracias por apoyarme en mis momentos más difíciles. Todo este trabajo no es únicamente mío sino que gran parte le pertenece a ella por lo cual siempre le estaré infinitamente agradecido. Espero nunca haberla defraudado con mis decisiones tomadas a lo largo de mi vida. Por último, gracias por sacar adelante a mis hermanos y a mí, esto es algo que nunca podremos pagarle ninguno de nosotros.

A mi padre:

Le estoy agradecido que gracias a que él siempre ha trabajado hemos podido llegar tan lejos con lo poco o mucho que nos pudo dar ya que sin él, hubiera sido más difícil de lo que fue.

A mis hermanos:

Les agradezco por el apoyo brindado en las diferentes etapas de mi vida, por los consejos que me ofrecieron para que yo fuera una mejor persona, aunque a veces peleamos mucho nunca dejaron de ser parte de mi familia con la cual siempre podré contar en todo momento.

A todos mis amigos:

Gracias amigos por todas las experiencias que vivimos a lo largo de la carrera, por aguantar todos mis errores y compartir conmigo una importante etapa de nuestras vidas, siempre les estaré agradecido por todo su apoyo.

A la UNAM y Facultad de Ingeniería:

Gracias por permitirme ser parte de ella brindándome la oportunidad de cumplir con mi formación académica y mostrarme que para conseguir algo debes de luchar por ello, siempre la llevaré en el corazón ya que nunca dejaras de ser mi segunda casa.

Al Palacio de Minería

Gracias por el apoyo brindado para la elaboración del trabajo de tesis, en especial al PAT por permitirme concluir esta etapa de mi vida.

M.I. Juan Carlos Roa Beiza

Por dedicarnos el tiempo, apoyo y paciencia durante el desarrollo del trabajo de tesis.

A mis compañeros de tesis:

Gracias al equipo del PAT formado por: Gustavo Jiménez Pérez número de cuenta 081162400, Miguel Álvarez Solís número de cuenta 306135204, Gustavo Saldivar Quiñonez número de cuenta 085602698 y Diana Elguera Almazán número de cuenta 085115123.

I am not a product of my circumstances.

I am a product of my decisions.

Stephen Covey

Álvaro Manuel Hernández Márquez

Agradecimientos:

Antes que a nadie debo dar gracias a Dios por permitirme tener salud y vida para concluir con mi preparación profesional.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por su gran contribución a nuestra formación como profesionistas a través de sus profesores.

A nuestro asesor el M.I. Juan Carlos Roa Beiza y al Programa de Apoyo a la Titulación (PAT) de la Facultad de Ingeniería de la UNAM, por haber creado esta gran opción y facilitar el culminar una más de nuestras metas.

Al grupo de compañeros de la Facultad de Ingeniería y de la Facultad de Estudios Superiores Acatlán, que participaron en la elaboración de esta tesis:

- * Álvaro Hernández Márquez número de cuenta 305069562*
- * Miguel Álvarez Solís número de cuenta 306135204*
- * Gustavo Saldivar Quiñonez número de cuenta 085602698*
- * Diana Elguera Almazán número de cuenta 085115123*

A mi mejor amigo Edgar por todo tu apoyo y confianza para la realización de este trabajo. Raúl gracias por tu apoyo.

A ti Erik por tus consejos y apoyo incondicional, mejor mentor no podría tener, gracias.

Dedicatorias:

A mi Mamá Chelo como un pequeño tributo por todos esos años dedicados a mi desarrollo físico e intelectual.

A todos mis seres queridos que estuvieron conmigo ayer y están hoy, a todos ellos quienes trataron de que me titulara, les agradezco todo su apoyo, su esfuerzo no fue en vano. En especial a mis hermanos, a mis hijos Omar, Sandy, Fabián y a mi Papi Gustavo

Gustavo Jiménez Pérez.

Agradecimientos:

A mis compañeros de Tesis Diana Elguera Almazán número de cuenta 085115123, Gustavo Jiménez Pérez número de cuenta 081162400, Miguel Álvarez Solís número de cuenta 306135204 y Álvaro Hernández Márquez número de cuenta 305069562. quienes tuvieron que sacrificar algunas horas y fines de semana para culminar este trabajo. Gracias por su esfuerzo y dedicación para unificar diferentes puntos de vista con el fin de llegar a esta meta que, para algunos de nosotros da mayor significado concluir la después de varios años de estar lejos de nuestra alma mater la cual nos ha permitido enfrentar las adversidades del mundo laboral que nos deja, así como enojos y frustraciones en algunas ocasiones, en otras miles de satisfacciones al poder hacer algo por nuestros semejantes aplicando todo lo aprendido en sus instalaciones.

A Claudia... mi esposa, amiga y compañera de buenos y malos momentos en los cuales ha estado ahí, junto a mí alentándome para seguir siempre adelante sin importar lo tortuoso que en ocasiones pueda estar el camino. Quien influyó decisivamente para retomar este sendero que me permitió culminar este trabajo de tesis y presentar mi examen profesional. Gracias por ser el motivo de mi existencia y por compartir grandes momentos a mi lado... gracias... sinceramente y de corazón gracias por todo tu apoyo para lograr este objetivo.

A mis Hijos Diana e Irving, quienes desde su llegada a este mundo me han motivado a darlo todo y a terminar todo lo que se empieza, aun cuando el tiempo o esfuerzo a invertir para lograr un objetivo parezca imposible de ser suficiente. Gracias porque sin darse cuenta, lograron que estando en el suelo me levantara para continuar adelante... siempre adelante, tratando de no volver la vista atrás. Gracias por forzarme a retomar este tema de titularme a fin de servir de ejemplo para ustedes y a su vez, puedan en tiempo y forma hacer lo mismo para enfrentar la vida con la honestidad y frente siempre en alto... y recordar que... en esta vida lo difícil cuesta algo de tiempo... lo imposible un poco más!

Gustavo Saldivar Quiñonez

Agradecimientos:

A mis padres.

Les agradezco infinitamente por dejarme la mejor herencia que pudieron darme: mi educación universitaria así como formación de valores que me han convertido en el hombre que soy. Siempre estaré en deuda con ustedes.

Gracias mamá por haberme acompañado a lo largo de toda mi vida, sin tu amor, paciencia, apoyo y comprensión difícilmente hubiera podido llegar a terminar mi carrera universitaria. Eres una madre ejemplar y me siento tan afortunado de que tú seas mi madre.

Gracias papá por todo el esfuerzo que hiciste por mantenerme y darme lo necesario en todos estos años.

A mis hermanos.

Edgar Hugo a pesar de que físicamente ya no estas con nosotros sé que no dejaste de estar a mi lado, donde quiera que estés te agradezco todos los momentos que pasamos juntos, y en especial a ti te dedico esta tesis.

Teresa eres la mejor hermana del mundo, muchas gracias por todo tu apoyo económico y moral que me brindaste en todo momento, eres y seguirás siendo un ejemplo para mí.

Oscar gracias a ti me interesó estudiar una Ingeniería, tú fuiste la primera persona que confió en mí para lograr esta meta, muchas gracias por tu apoyo y consejos a lo largo de este tiempo.

A la UNAM.

En especial a la Facultad de Ingeniería por brindarme una oportunidad de obtener una formación profesional de excelencia.

Al PAT.

Por abrirme las puertas y permitirme realizar el presente trabajo de tesis de una manera rápida y concisa.

Al mi director de tesis el M.I. Juan Carlos Roa Beiza que es un ingeniero en toda la extensión de la palabra, desde que lo conocí, siempre ha sido una inspiración de superación, gracias por compartir su experiencia y conocimiento.

Mis compañeros del PAT integrado por: Álvaro Hernández Márquez número de cuenta 305069562, Gustavo Jiménez Pérez número de cuenta 081162400, Gustavo Saldivar Quiñonez número de cuenta 085602698, Diana Elguera Almazán número de cuenta 085115123.

Muchas gracias compañeros, hicimos un equipo fuerte, combinando experiencia, conocimiento, estrategia, compromiso y trabajo. Estoy orgulloso de trabajar con ustedes.

A mis amigos.

Miguel González, eres uno de mis mejores amigos, muchas gracias por ayudarme y ser como un hermano.

Daniel Mejía y Andrés Piña, amigos desde hace mucho tiempo, gracias por su apoyo.

Sofía Mireya Camacho, gracias por ser una persona especial, por compartir parte de tu tiempo conmigo, tu humor y tus palabras siempre llegaban en el momento mejor indicado.

Compañeros del trabajo.

Ileana Ibarra, Alfredo Lora, Rogelio Villafranco y Christian Zugaide, Muchas gracias por su apoyo y afecto.

Vianey Méndez, Angélica Torres y Ángel Melesio muchas gracias por apoyarme con mi titulación, sin ustedes hubiera sido muy difícil obtener tiempo terminar éste proyecto.

Miguel Álvarez Solís

ÍNDICE

| | |
|--|-----------|
| CAPÍTULO I ENTORNO DEL PROBLEMA..... | 1 |
| 1.1 Introducción..... | 2 |
| 1.2 Normatividad que debe ser cumplida para el manejo documental..... | 6 |
| 1.3 Diagrama de logística de administración de expedientes..... | 12 |
| 1.4 Métodos para digitalizar información..... | 17 |
| 1.5 Acotamiento del problema..... | 27 |
| | |
| CAPÍTULO II MARCO TEÓRICO..... | 31 |
| 2.1 Características, ventajas y desventajas en las bases de datos relaciones, y metodología de Yourdon/De Marco..... | 32 |
| 2.2 Características, ventajas y desventajas de JavaScript, jQuery y XHTML..... | 46 |
| 2.3 Características, ventajas y desventajas de la arquitectura WEB..... | 53 |
| 2.4 Características, ventajas y desventajas de la base de datos SQL-Server..... | 60 |
| 2.5 Características, ventajas y desventajas de la arquitectura cliente/servidor..... | 66 |
| | |
| CAPÍTULO III PLANTEAMIENTO DEL PROBLEMA..... | 75 |
| 3.1 Análisis del problema..... | 76 |
| 3.2 Requerimientos generales y particulares de la aplicación..... | 81 |
| 3.3 Análisis y posicionamiento de los lenguajes para el back-end y front-end | 85 |
| 3.4 Separación de los posibles módulos según la metodología de Yourdon/De Marco | 97 |
| 3.5 Diseño a nivel de bloques de la interfaz del usuario..... | 100 |

| | |
|---|------------|
| CAPÍTULO IV ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN..... | 109 |
| 4.1 Introducción..... | 110 |
| 4.2 Creación de la base de datos e implementación del Sistema de Administración Documental (SAD)..... | 146 |
| 4.3 Creación del front-end con la funcionalidad del órgano rector..... | 161 |
| 4.4 Generación de pruebas, mantenimiento y reportes..... | 176 |
| 4.5 Tipos de mantenimiento..... | 191 |
| Conclusiones..... | 193 |
| Referencias..... | 196 |
| Apéndices..... | 199 |
| Apéndice 1 Instalación del servidor Apache | 200 |

CAPÍTULO I

ENTORNO DEL PROBLEMA

1.1 Introducción

La institución que tomamos como base para el desarrollo del presente trabajo, es un órgano gubernamental del Distrito Federal que tiene como objetivos impulsar una política económica que promueva las decisiones del mercado, regule el crecimiento económico, brinde certeza jurídica y transparencia a las inversiones para una distribución equitativa de los beneficios a toda su población.

Ésta busca impulsar la capacidad competitiva del Distrito Federal a través de su especialización productiva y su inserción en los mercados local, regional y global para convertirla en una economía líder. Para lograrlo promueve créditos, asesoría técnica y administrativa para la micro, pequeña y mediana empresa. Dichas acciones mejoran la calidad de vida de la población ya que se crean nuevas capacidades y habilidades laborales profesionales para el fortalecimiento del empleo.

Día a día los usuarios de las áreas de administración generan y resguardan documentación física relacionada con la gestión de recursos humanos, adquisiciones, finanzas y pagos a proveedores de servicios. Pero el resguardo de expedientes, implica ciertas dificultades:

- Contar con un área cada vez mayor donde se resguarden los archivos en papel de todos los expedientes.
- Deterioro de los expedientes por causas como la humedad y tiempo de vida útil del papel.
- Posibilidad de extravíos de documentos físicos.
- Búsquedas lentas y poco eficientes de documentos entre una gran cantidad de expedientes.

Esta situación ha impedido que los procesos de las áreas administrativas de la organización sean eficientes, por ello surge la necesidad de crear un sistema a la

medida capaz de almacenar digitalmente toda la documentación de una manera automática, segura, fácil de usar y administrar. Además la organización podrá obtener los siguientes beneficios:

- Tener la capacidad para generar expedientes digitales.
- Compartir información entre diferentes departamentos y usuarios de la institución.
- Buscar, consultar y acceder a documentos de manera rápida y segura.
- Concentrar el acervo documental en un repositorio único de datos.
- Crear un respaldo de seguridad para evitar pérdida de información.
- Disminución del uso de papel (ahorro de recursos institucionales).

La aplicación se desarrollará con tecnología de software que posea el usuario y con software libre para no incurrir en costos adicionales pero que sea lo suficientemente robusta para cumplir con los requerimientos del usuario final.

El presente trabajo se encuentra dividido en 4 capítulos: entorno del problema, marco teórico, análisis y planteamiento del problema, y por último diseño y construcción de la aplicación.

En el primer capítulo se realiza un análisis del entorno del problema, describiendo el estado actual en el que se encuentra la organización sin el Sistema de Administración Documental (SAD). Además se describe la normatividad y logística involucrada con el manejo de información de la organización. Se dará una reseña de los métodos de digitalización de información existentes en el mercado. Por último, se delimitará el problema para la búsqueda de la mejor solución.

En el segundo capítulo se analiza el marco teórico, en el cual se describen los lenguajes de programación sobre los que se desarrollará nuestra aplicación, así como la metodología que se seguirá para cumplir con nuestros objetivos del sistema. Además se identificarán las características principales, así como las ventajas y desventajas de los métodos, la plataforma usada y las tecnologías implementadas.

El tercer capítulo, planteamiento del problema, está orientado al entorno actual y los problemas a los que se enfrentan los usuarios de la organización desde la recepción de la documentación, resguardo, búsqueda y el tiempo de vida útil del papel. También se considerarán los lenguajes de programación implementados para el back-end y front-end de la aplicación. Y por último pero no menos importante se hará un diseño de bloques de la interfaz del usuario que permitirá una construcción más fácil y eficiente.

En el cuarto capítulo, análisis, diseño y construcción de la aplicación, se describe el proceso de diseño mediante los diferentes diagramas sobre los cuales se fundamenta el SAD, posteriormente se encontrará el diseño, desarrollo e implementación de la base de datos, así como el diseño del sistema y el desarrollo del mismo. Debido a que la mayoría de los modelos de esta metodología son gráficos permite esquematizar el sistema completo y que sea de fácil entendimiento para todas las partes involucradas, disminuyendo posibles errores de interpretación u omisiones.

Se describirá también el diseño de la interfaz con el usuario tratando de que sea lo más amigable posible y sencilla de utilizar. Se mostrarán algunos ejemplos de pruebas y se definirá la seguridad que integrará al sistema para minimizar la probabilidad de ataques o pérdida de información por mal manejo de los datos.

Las aplicaciones y metodologías que se utilizarán para el desarrollo del SAD se basan en las mejores prácticas y procesos que manejan los sistemas de control de archivos en diversas áreas industriales, utilizando herramientas y soluciones que se enfocan en las necesidades claves del usuario y la organización, tanto tácticas como estratégicas, para maximizar el desempeño y funcionalidad adecuándose a las necesidades de la organización.

Con todo lo que se ha expuesto hasta el momento podemos exponer los objetivos que se persiguen con este trabajo, los cuales son los siguientes:

- Automatizar la búsqueda y consulta de documentos entre las diferentes áreas que conforman una organización.
- Lo anterior mediante una aplicación WEB.
- Administrar los expedientes físicos y lógicos relacionados con la gestión organizacional.
- Optimizar y agilizar las tareas relacionadas con la administración de documentos en una organización.
- Evitar el extravío de documentos y/o expedientes.
- Crear una bitácora para rastrear la consulta o alteración de los expedientes en base a su usuario y contraseña.
- Proteger la información contenida en expedientes confidenciales de personal no autorizado.
- Organizar la consulta y guarda de la información histórica.
- Ofrecer una solución con las herramientas que posee el usuario de las que ya tiene licencia.

1.2 Normatividad que debe ser cumplida para el manejo documental.

En la organización se ha determinado apegarse a lo establecido en los Capítulos III, IV y V de la Ley de archivos del Distrito Federal, para el manejo del acervo documental institucional.

A continuación se enuncian los criterios que deben ser considerados por las áreas responsables que conforman la organización, para la recepción, registro, seguimiento, organización, clasificación, localización, despacho, uso, transferencia, resguardo, conservación, selección y destino final de los documentos y expedientes que se generen, obtengan, adquieran o conserven por cualquier título en sus archivos, como resultado de su gestión.

1.2.1 Políticas para la administración de un sistema de archivo:

- Todo empleado de la organización está obligado a garantizar la integridad y conservación de los expedientes y documentos que, en razón de sus atribuciones, le corresponde resguardar, así como facilitar la consulta y uso.
- El eje fundamental para la administración de documentos y la archivística lo constituye el ciclo vital de los documentos de archivo, el cual está integrado por las siguientes fases:
 - Activa o de gestión, 1ª edad de los documentos.- Archivo de trámite o de gestión. Está conformada por los documentos que se encuentran en trámite.

Las áreas responsables de la organización, generadoras de los documentos como producto de una acción administrativa y con una función específica, deberán organizar sus archivos de trámite. De la misma manera, deberán conservar los documentos originales como testimonios, prueba y continuidad de la gestión administrativa necesaria para la adecuada toma de decisiones.

La Unidad de Archivo de Trámite (UAT) es la responsable de la administración de los documentos de archivo en gestión, de uso cotidiano y necesario para el ejercicio de las atribuciones y funciones de cada área administrativa.

- Semiactiva o de conservación precautoria, 2ª edad de los documentos de archivo.- Archivo de concentración. Está conformada por los documentos que habiendo concluido su trámite y luego de haber sido valorados, sean transferidos por la Unidad de Archivos de Trámite a la Unidad de Archivo de Concentración (UAC) para su conservación precautoria.

La Unidad de Archivo de Concentración es la responsable de la guarda, custodia y conservación de documentos de archivo en la fase semiactiva, cuya frecuencia de consulta ha disminuido y que permanecen en él hasta que concluye su plazo de conservación en razón de sus valores primarios de carácter administrativo, legal y fiscal.

- Inactiva o histórica, 3ª edad de los testimonios documentales.- Archivo Histórico. Conformada por los documentos que habiendo completado su vigencia en la Unidad de Archivo de Concentración, sean transferidos para completar su Ciclo Vital a la Unidad de Archivo Histórico.

La Unidad de Archivo Histórico o equivalente, es la responsable de organizar, describir, conservar, preservar, administrar y divulgar la memoria documental institucional.

- Todos los expedientes que se encuentren en el Archivo de Concentración, estarán a disposición del área que los haya generado y transferido, presentando para tal efecto la solicitud firmada, siendo la única que los podrá recibir en préstamo. La Unidad de Archivo de Concentración establecerá los mecanismos de consulta necesarios para garantizar que los expedientes archivados puedan ser ágilmente localizados y que los expedientes en préstamo puedan ser devueltos una vez concluido el plazo máximo de consulta que será de 30 días hábiles. Dicho tiempo podrá ser prorrogado a solicitud expresa y debidamente justificada por 30 días hábiles más a cada prórroga. También podrán ser solicitados por la autoridad competente en el ejercicio de sus atribuciones legales.
- La Unidad de Archivo de Concentración es la responsable de garantizar la óptima operación y mantenimiento de los archivos y sus instalaciones, así como de la

conservación en buen estado de sus acervos, asegurándoles condiciones de seguridad e higiene y las dimensiones que permitan resguardar, conservar y localizar con prontitud todos los documentos que los integren.

- La integración de expedientes deberá hacerse en fólderes, carpetas o legajos plenamente identificados con las carátulas estandarizadas establecidas en la organización, con el objeto de homologar su clasificación y descripción. Los expedientes deberán foliarse para preservar la integridad de la información que contienen y también deberá evitarse incorporar elementos que dañen o lastimen a los documentos que los constituyen, como clips o grapas.
- Los archivos que contengan datos personales deberán mantenerse actualizados y ser utilizados exclusivamente para los fines legales para los que fueron creados.
- Contenido de los expedientes: Por regla general, los expedientes deben contener los documentos originales o copias certificadas de los mismos. Excepcionalmente, pueden contener copias simples para los casos en que jurídicamente se permita o requiera su manejo.

Las fotocopias de documentos y los expedientes multiplicados para controles internos son instrumentos que facilitan la operación administrativa y no serán considerados como documento de archivo.

Las notas y tarjetas informativas podrán darse de baja en la propia oficina en el momento en que pierdan su utilidad funcional. Las síntesis informativas, revistas, diarios y otros materiales documentales similares, no se considerarán como parte del archivo. Éstos, una vez concluida su vida útil, podrán canalizarse a otros acervos como bibliotecas o centros de documentación o, en su caso, podrán darse de baja para su reciclamiento.

Los documentos de fax o impresiones de correo electrónico sin valor legal, son instrumentos de comunicación empleados para optimizar tiempos y agilizar trámites; no excluyen el trámite de recepción o envío del documento original y sólo serán materia de archivo temporal hasta que se reciban los originales.

Las áreas generadoras de la organización, sólo conservarán en sus archivos de trámite y en sus expedientes los documentos de entrada y los “Acuses de recibo”, documento que hacen las veces del original emitido.

Las unidades administrativas deberán implementar un control para el seguimiento de los números de oficios que giran así como de la correspondencia que reciben y turnan a sus áreas respectivas para su atención. Por ser éste un control interno no será causa de apertura de serie, por lo que solamente permanecerá en las áreas durante el tiempo que ésta estime pertinente.

- Los documentos de archivo que se clasifiquen como de acceso restringido, bajo las figuras de información reservada o confidencial por las unidades administrativas que conforman la organización, deberán identificarse y organizarse de conformidad con las presentes disposiciones y para su correcta administración y control, se aplicarán las disposiciones técnicas siguientes:

Si dentro de una serie documental se ha restringido el acceso a un solo expediente o a un grupo de expedientes, éstos deberán registrarse en un inventario especial dentro de cada área administrativa en el que se señalen los datos de identificación siguientes:

- Nombre de la unidad generadora o usuaria de los expedientes de acceso restringido;
- Título del expediente;
- Serie documental a la que pertenece;
- Año del expediente y en su caso fecha aproximada de cierre del mismo;
- Plazo de reserva del expediente;
- Justificación jurídica de la reserva;
- Fecha de apertura pública del expediente; y
- Unidad responsable de la custodia de los expedientes de acceso restringido.

1.2.2 Apertura y clasificación de expedientes.

Con la finalidad de asegurar que los documentos de archivo de la organización se encuentren debidamente integrados en cuanto a su descripción, valoración, agrupación, presentación, accesibilidad y funcionalidad, se establecieron las siguientes políticas y/o normas de operación:

- Los documentos deben corresponder a un mismo asunto para ser integrados al mismo expediente, unidos por relación de causa y efecto con base a la secuencia del trámite.
- Los documentos se integran de manera sucesiva y natural, siguiendo el orden cronológico ascendente o descendente de acuerdo a la fecha de su recepción o generación.
- Los expedientes se pueden guardar en fólder, carpeta u otro medio que garantice su integridad y fácil manejo.
- Todos los expedientes deben estar debidamente identificados de acuerdo a su origen estructural y funcional.
- Los registros en cada área se harán por expediente y no por documento.
- Cada expediente y/o carpeta deberá identificarse con la carátula estandarizada correspondiente, la cual permite la plena identificación y agrupación, misma que debe ser adherida con pegamento líquido (no utilizar lápiz adhesivo) para asegurar su permanencia.
- Los expedientes deben contener los documentos originales o los que hacen las veces de estos. Solamente se podrán integrar copias de los documentos cuando el asunto que desahoga el expediente así lo requiera como único soporte existente y sin el cual perdería la ilación del mismo asunto.
- El grosor de los expedientes dependerá de la forma en que se encuadernen y sujeten los documentos con el fin de evitar que se dañen o se desprendan los documentos que lo integran, y así facilitar también su manejo. El grosor promedio será de cinco centímetros.
- Para la clasificación de los expedientes, las áreas podrán tomar las claves o títulos que se asignan por normatividad a ciertos asuntos (parques industriales, mercados, abasto, aperturas, etc.).
- Los funcionarios de estructura responsables de las áreas de la organización, podrán nombrar un responsable del archivo de trámite correspondiente.

1.2.3 ¿Por qué tener un archivo bien ordenado y clasificado?

Derivado del importante incremento del volumen documental de la organización, se hace necesaria la homologación de la gestión de la documentación administrativa, así como la aplicación de criterios de calidad y eficacia en la organización y acceso a la documentación que genera todas las áreas que la integran.

Ya que toda la documentación producida por la organización forma parte de su patrimonio documental y de su memoria histórica, homologar los procedimientos de organización de los documentos en las oficinas facilitará una gestión rápida y eficaz por parte de las unidades administrativas durante el periodo inicial en el que los expedientes se tramitan y custodian en las propias oficinas, asegurando que los documentos se conserven, se mantengan organizados y accesibles para la comunidad de la organización, en el contexto en el que se generaron. Con esto asentamos las bases para el establecimiento de un Sistema que permita la gestión global e integrada de la documentación institucional, independientemente de su soporte físico y de la fase del ciclo de vida en que se encuentre.

1.3 Diagrama de logística de administración de expedientes.

El diagrama de logística da un panorama de la administración de documentos y la archivística lo constituye el ciclo vital de los documentos de archivo, el cual se muestra en la figura 1.3.1:

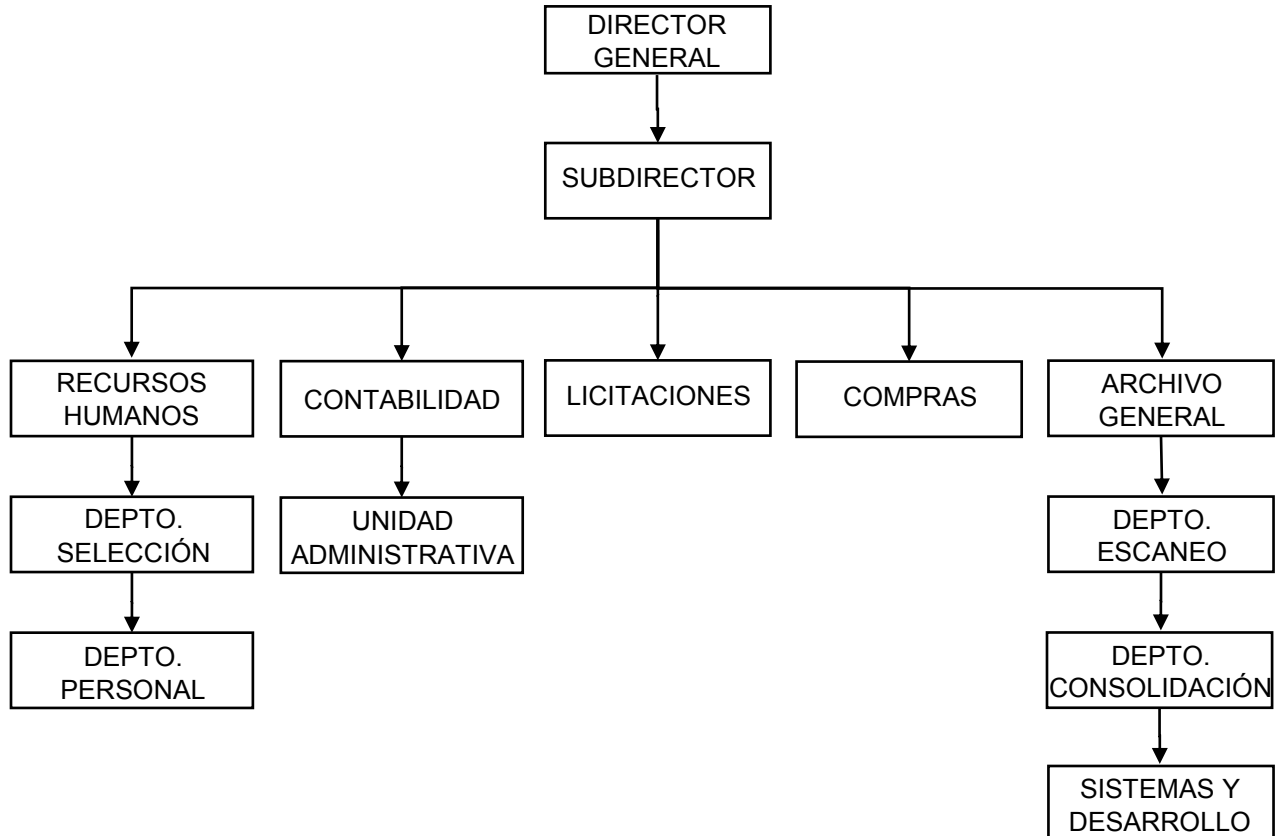


Figura 1.3.1 Diagrama funcional sintetizado.

En el diagrama sólo se muestran las áreas de interés para este proyecto, de las cuales se hará una breve descripción a continuación:

- **Director general:** es la persona encargada de que toda la logística del negocio funcione adecuadamente, de acuerdo a los lineamientos establecidos para ello. Es la persona encargada de tomar las decisiones de todas las áreas.

- **Subdirector:** es el encargado de filtrar todos los asuntos hacia el director general, éste debe resolver los asuntos más inmediatos y que no requieran de un visto bueno del director.
- **Recursos humanos:** es el encargado de todas las contrataciones de personal y de las cuestiones administrativas del mismo, en base a las características de cada trabajador o empleado.
- **Departamento de selección:** éste es el encargado de establecer los estándares para cada uno de los puestos en los que se requiera personal de base y de confianza.
- **Departamento de personal:** éste se ocupa de todos los asuntos de los trabajadores de base en todos los aspectos de tal forma que no se presenten problemas en la dependencia.
- **Contabilidad:** éste lleva a cabo todos los procesos contables de la dependencia, y se encarga de que el presupuesto se vaya ejerciendo adecuadamente según las necesidades de la dependencia.
- **Unidad administrativa:** es la encargada de recopilar y salvaguardar toda la información contable que se vaya generando en cada uno de los departamentos y de administrar las partidas de cada uno de ellos, para que no se sobregiren.
- **Departamento de licitaciones:** es el encargado de llevar a cabo las licitaciones de los insumos o servicios que requiera la dependencia para su buena operación.
- **Compras:** es el encargado de efectuar las compras que han sido autorizadas y revisadas por la unidad administrativa, licitaciones y contabilidad.
- **Archivo general:** éste se encarga de resguardar toda la información generada en la dependencia vigilando que ésta no se deteriore, que la confidencial solo pueda ser consultada por los entes autorizados, prestar la información que le es solicitada por cada uno de los departamentos y vigilar que sea devuelta en tiempo y forma.
Cabe aclarar que cada departamento es responsable de su propia información mientras esta no sea transferida al archivo general y tienen la obligación de cuidarla y resguardarla para su uso interno.

- **Departamento de consolidación:** se encarga de organizar e inventariar la información para consolidarla de una forma en que pueda ser fácilmente consultada por quien lo requiera.
- **Departamento de escaneo:** cuando la información llega al archivo general este departamento tiene la encomienda de ir escaneando los archivos poco a poco para convertirlos en archivos electrónicos, y que el departamento de sistemas y desarrollo pueda aprovecharla para su consulta (éste último es de reciente creación).
- **Sistemas y desarrollo:** éste es de reciente creación y su finalidad es satisfacer las necesidades cada vez más imperantes de consulta de información electrónica, y es el encargada de desarrollar, probar y poner en marcha todos los sistemas de la dependencia, para lograr que esta se integre a la era de las tecnologías de la información y pueda brindar a sus usuarios un mejor desempeño de sus labores.

A este departamento estará enfocado el desarrollo de nuestro sistema para que pueda ser usado por la dependencia, una vez que pase por todo el proceso de desarrollo, prueba y capacitación del personal que hará uso de este, el cual se elegirá de acuerdo a que posea los conocimientos básicos de computación para su rápida inmersión en el sistema.

Ahora explicaremos de una forma sintetizada como es el proceso de recopilación de un archivo por las diferentes áreas y cuál es el proceso general que sigue, hasta que es resguardado en el archivo general, vea figura 1.3.2.

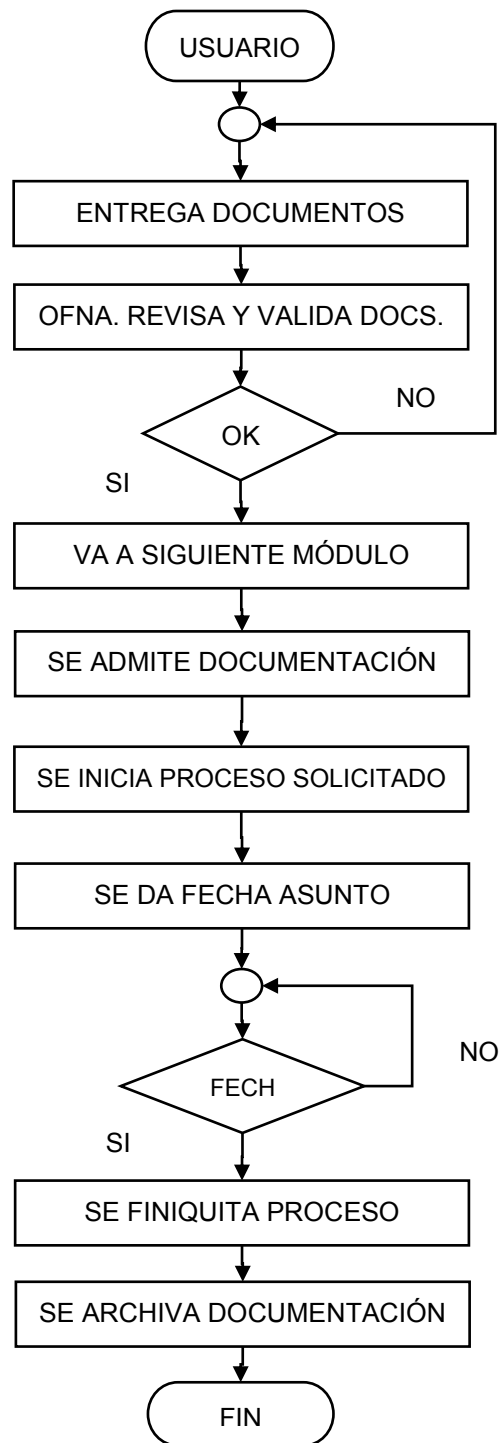


Figura 1.3.2 Proceso de entrega de documentos.

A continuación se describirá el proceso anterior brevemente:

- El usuario llega a la oficina y solicita un trámite.
- Se le pide la documentación correspondiente y se canaliza a módulo.
- Hace entrega de documentos y se revisan los mismos.
- Si falta alguno se le pide vuelva con éstos.
- Si están completos se le canaliza al siguiente módulo.
- Se admite su solicitud.
- Oficina canaliza solicitud para que sea atendida por el departamento correspondiente.
- Se le da una fecha para que se atienda su asunto y se le pide volver en una fecha próxima.
- Al retorno del usuario se le da la respuesta a su asunto.
- Se finiquita el asunto.
- Por último se archiva en el departamento correspondiente para que después sea escaneado en el área de digitalización.

Como se puede apreciar, de todo este proceso sale un volumen inmenso de documentos de los diferentes departamentos, los cuales son archivados en forma local y después de mucho tiempo se envían al archivo general.

El proyecto pretende desarrollar un sistema que capture los nuevos documentos que se van aceptando de los usuarios directamente a su ingreso, en los diferentes departamentos, y posteriormente ir subiendo los documentos digitalizados en el mismo sistema para tener toda la información disponible cuando se requiera.

Se ha planteado ir incorporando poco a poco a todos los departamentos para que hagan uso del sistema, sin embargo se comenzará por uno y se analizarán los resultados, se evaluará como responde el sistema y se harán los ajustes al mismo hasta lograr dejarlo en óptimas condiciones.

1.4 Métodos para digitalizar información.

Un programa de digitalización es calificado como satisfactorio cuando es posible retirar de circulación el material original, toda vez que la imagen digital se convierte en un sustituto equivalente de los documentos.

El área de archivo en una organización se caracteriza por el tipo de soporte que ofrece y los medios que soportan la información: papel, videos, discos, casetes. Esta área debe permitir realizar la digitalización de información analógica, sin afectar o agredir las condiciones actuales del medio que contiene la información.

En el caso del papel, deberán identificarse variables relativas al formato; por ejemplo, ¿Cabén los documentos en el cristal de digitalización del escáner? ¿Las condiciones físicas del material soportan sin riesgo la tracción del alimentador? ¿El grosor de los documentos permite la alimentación automática? ¿Los documentos están libres de elementos extraños (clips, grapas, ganchos, etc.) y preparados físicamente (sueltos y sin dobleces) para un trabajo automatizado de carácter masivo?

1.4.1 Métodos de digitalización en base a las clases de equipos.

Sin duda existe en el mercado un equipo de digitalización apropiado para una documentación que posee un archivo en particular y lo primero que resulta importante es identificar la relación directa, entre el equipo de digitalización y los documentos que se vayan a digitalizar.

Un escáner es un periférico que permite digitalizar documentos, incluyendo en la definición las cámaras digitales y cualquier dispositivo capaz de crear una representación informática de soportes analógicos.

Los escáneres funcionan utilizando el principio básico de la transparencia de la luz. Se coloca la imagen, enfrentada al bloque lector y al cabezal lector compuesto por el CCD

(Charge Couple Device), el sistema de iluminación y un conjunto de lentes se desplazan barriendo la imagen. La luz reflejada se convierte en energía eléctrica y la velocidad del movimiento del cabezal determina la resolución. Cuanto menor sea la velocidad del lector, más información se extraerá de la imagen.

A continuación se describen los principales métodos de escaneo:

Escáner de cama plana.

El escáner de cama plana es un dispositivo de captura diseñado para documentación que no puede ser procesada masivamente a través de alimentadores automáticos debido a las condiciones físicas del material a digitalizar.

A pesar de que se trata de uno de los procesos más lentos de captura documental, la mayor bondad que representa está enfocada en una agresión mínima del documento original, básicamente la causada por la intensidad de luz, ya que es una lámpara la que se desplaza y realiza el escaneo del documento.

Escáner de alimentación automática.

La mayoría de los escáneres dispuestos en el mercado fueron diseñados para producción en serie de documentación con características particulares, generalmente orientados en áreas de negocio específicas.

La adopción de un escáner a las necesidades particulares de los archivos debe comenzar por identificar si el equipo es compatible con los documentos y en el caso de los escáneres de alimentación automática, si la documentación está “preparada” físicamente para soportar una tracción automática. Esto significa que la documentación en muy buen estado o con muy buena preparación podrá ser digitalizada a partir de alimentadores automáticos, mientras que otra requerirá del trabajo mecánico con la cama plana del dispositivo.

Una doble lectura puede resultar conveniente para documentos con información por ambas caras de la hoja, lo que requerirá entonces de un sistema dúplex de captura, a partir de alimentación automática, bondad que se verá ampliada si el alimentador soporta diferentes tamaños y pesos; combina alimentación justificada en varios sentidos (derecha, centro, izquierda); si la bandeja de elevación acepta gran cantidad de hojas y provee literalmente un funcionamiento sin atascos de papel. Los tamaños que acepta, medios que soporta (transparencias, negativos, diapositivas, entre otros), condiciones físicas como encuadernaciones, tamaños, papeles, formatos, etc., se convierten en elementos de criterio útil para decidir el dispositivo a adquirir por el área responsable en una organización.

Escáner de tambor.

Éste es el sistema que más fielmente reproduce el documento original. Este tipo de escáner no es para el usuario aficionado debido a su elevado precio, por lo que está reservado a trabajos profesionales de artes gráficas (laboratorios, imprentas, etc.).

El sistema consiste en un tambor de cristal de gran pureza al cual se le pega el documento. Este tambor gira a alta velocidad y un sensor va recogiendo fragmentos del documento.

Cámara digital.

Una cámara digital es un dispositivo cuyo principal objetivo es captar fotografías en formato digital. Todas las cámaras utilizan un dispositivo CCD o un sensor CMOS, para transformar las imágenes o el video en datos electrónicos.

Los dispositivos CCD transportan la carga a través del chip hasta un convertidor analógico-digital que convierte el valor de cada píxel en un valor digital, midiendo la carga que llega. Dependiendo del número de bits del conversor, obtendremos un mayor o menor números de gamas de color.

Las cámaras digitales son similares en clasificación a las tradicionales de película; las estándar y las réflex, diferenciadas básicamente porque estas últimas posibilitan el uso de objetivos que exponen la luz sobre un sensor de imagen. Se caracterizan porque incluyen un sensor de mayor tamaño que las cámaras compactas lo que permite capturar distancias focales mayores y con ello obtener un mayor control sobre la profundidad de campo.

Escáner de planos.

El escáner de planos también conocido como escáner de gran formato, es uno de los dispositivos más adecuados para la digitalización de material planimétrico y cartográfico.

Si bien algunas especificaciones de carácter profesional pueden ser suplidas a partir de cámaras digitales para capturar grandes formatos, la relación 1:1 existente en una digitalización de planos con éste tipo de escáner, supera ampliamente las capturas digitales a partir de una cámara.

Se trata de dispositivos especialmente diseñados para digitalizar documentos que en los otros escáneres, se harían seccionadamente. Hacer una toma digital única facilita no solo el servicio de consulta e impresión, sino que evita la fotocomposición de partes, una tarea tediosa para el usuario y costosa debido al tiempo invertido.

Toda vez que se trata de alimentaciones mecánicas que generan tracción sobre los documentos originales, la gran mayoría de estos dispositivos han sido dotados de sábanas de recubrimiento para los documentos que físicamente puedan ser agredidos, especialmente diseñadas para que no afecten técnicamente la generación de imágenes y protejan a la vez la integridad de estas valiosas fuentes de información.

La tecnología ha aportado grandes avances frente a estos dispositivos de captura permitiendo velocidades aceptables, modalidades monocromáticas y de color, alta resolución de escaneo, control de ajuste automático de color, tecnología avanzada de

CCD de hasta 48 bits y sistemas de tracción total que protegen y cuidan los documentos originales.

Escáner de transparencias y diapositivas.

El funcionamiento de este tipo de escáneres es similar al escáner plano de sobremesa, salvo que en lugar de escanear documentos opacos, lo hace con transparencias (negativos y diapositivas).

La calidad es superior al escáner plano, si bien el mayor problema suele estar en la cantidad de “ruido” que muestra, particularmente lo relacionado con motas de polvo e impurezas del negativo.

Escáner de microfilmes.

La micrografía consiste en la reproducción fotográfica de documentación a tamaño muy reducido, precisando de un lector para su posterior consulta. Los formatos tradicionales han sido la película de carrete de 35 y 16 milímetros y la microficha, normalmente de 15 X 10.5 cm. Esta técnica ha representado un sistema de duplicación documental cuya mayor importancia radica en la conformación de archivos de seguridad con finalidades de consulta, economía de espacio y reducción del deterioro de documentos.

Los escáneres de microfilm son en esencia de transparencias en donde la fuente de alimentación es un documento soportado en película de microfilm. Combinan en un único sistema la posibilidad de visualizar microfilmes con la impresión de documentos en papel y en archivo electrónico. Esto permite imprimir o escanear al mismo tiempo.

En términos generales del uso cotidiano que se ha establecido para estos dispositivos se relaciona con una digitalización no masiva, discrecional por parte de los usuarios, en donde a partir de una consulta se determina si se requiere digitalizar el documento microfilmado, la cual se hace a partir de lectores/impresores; sin embargo, para efectos de proceso masivo, el mercado cuenta con dispositivos de digitalización de microfilmes,

dotados de características de nivel profesional, disponiendo en estos dispositivos de poderosas densidades ópticas y alta resolución lo que garantiza un escaneo escalable.

Sistemas híbridos digital-microfilm.

También denominado microfilmador electrónico, es uno de los dispositivos que mayor nivel de aplicación representa para garantizar que los documentos digitalizados puedan estar disponibles en el futuro lejano.

El proceso consiste en la captura de imágenes a partir de las bondades y facilitación que proporcionan los escáneres de alta velocidad. La tecnología digital procesa las imágenes para lograr la mayor calidad posible. Para un acceso a largo plazo, las imágenes están respaldadas en un microfilm a prueba de obsolescencia y para acceso inmediato están disponibles en un sistema de imágenes.

Una película de características perfectas es posible de obtener a través de los microfilmadores electrónicos con imágenes derechas y perfiladas de acuerdo con las imágenes digitalizadas.

1.4.2 Características técnicas.

Velocidad.

Identifica la medida y cantidad de imágenes que puede procesar un dispositivo en una unidad de tiempo. Todos los escáneres presentarán variaciones de desempeño, relacionadas con las características técnicas de digitalización establecidas y por tanto, bien vale la pena preguntarse si un escáner es capaz de procesar lotes de hasta 150 páginas por minuto, cuál será su desempeño y velocidad, si el escaneo es a color, en modalidad de grises o bitonal.

Métodos de captura.

Existen básicamente tres métodos de captura digital para documentos soportados en papel. blanco y negro, escala de gris y color.

A pasar de ser una tendencia tecnológica que está siendo revaluada, especialmente porque la mayoría de dispositivos permiten captura de las tres modalidades, existen aún escáneres que bien tienen limitantes de captura en color o bien deben ser adaptados –física y lógicamente–, para que soporten esta modalidad.

Técnicamente existe la justificación porque la captura de color hace más costoso el dispositivo y el procesamiento de las imágenes; por tanto, para lograr los niveles adecuados, muchos equipos prefieren “especializarse” en lograr alto volumen con una modalidad hasta niveles de gris.

1.4.3 Características técnicas de las imágenes (inherentes al escáner)

Resolución.

Técnicamente la resolución identifica el número de puntos (píxeles) que se identifican en una unidad de medida, generalizada a pulgadas. Es la capacidad de distinguir los detalles espaciales finos. La resolución es la medida de lo detallada y precisa que es una imagen. La escala de resolución depende del dispositivo que se mida. Las digitalizaciones se miden en píxeles por pulgada –ppi– mientras que las tramas pueden medirse en líneas por pulgada (lpi), siendo claro que en todos los casos cuanto mayor sea la resolución, más detallada y definida será la imagen.

Dimensiones de píxel.

Son las medidas horizontales y verticales de una imagen, expresada en píxeles. Las dimensiones de píxel se pueden determinar multiplicando tanto el ancho como la altura por el número de puntos por pulgada a que se ha establecido la resolución.

Profundidad de bit.

Este concepto es determinado por la cantidad de bits utilizados para definir un píxel. A mayor profundidad, mayor será la cantidad de tonos (escala de gris o color) que pueden ser representados.

Compresión.

La compresión se utiliza para reducir el tamaño del fichero de imagen para su almacenamiento, procesamiento y transmisión. El tamaño del archivo para imágenes digitales puede ser muy grande, complicando las capacidades informáticas y de redes de muchos sistemas. Todas las técnicas de compresión abrevian la cadena de código binario en una imagen sin comprimir, a una forma de abreviatura matemática, basada en complejos algoritmos. Los sistemas de compresión pueden caracterizarse como sin pérdida o con pérdida. Los primeros abrevian el código binario sin desechar información, por lo que, cuando se “descomprime” la imagen es idéntica bit por bit al original. Los sistemas con pérdida, como JPEG, utilizan una manera de desechar la información “menos importante”, basada en un entendimiento de la percepción visual. La compresión sin pérdida se utiliza con mayor frecuencia en el escaneado bitonal de material de texto. La compresión con pérdida típicamente se utiliza con imágenes tonales en donde la simple abreviatura de información no tendrá como resultado un ahorro de archivo apreciable.

Calidad.

La calidad de la imagen conjuga un amplio número de variables, porque se trata no solo de las características de la captura, partiendo de las condiciones del original, sino del dispositivo de salida que se tenga previsto para la imagen.

Las políticas normales han limitado el tamaño de los ficheros reduciendo la resolución, la profundidad de bits y haciendo uso de la compresión. El objetivo es aumentar la velocidad de entrega, sin comprometer demasiado la calidad de la imagen. Todas las condiciones técnicas de la captura influyen en el resultado final, siendo destacables dos aspectos fundamentales: la resolución y la transmisión de color y tonalidad en forma efectiva.

La satisfacción del usuario con las imágenes en la pantalla dependerá de las posibilidades de los sistemas de visualización. Además de la velocidad de entrega, los usuarios están interesados en la calidad de la imagen (legibilidad y fidelidad de color

para una tarea); la presentación completa de las imágenes en pantalla y las representaciones exactas de las dimensiones de los documentos originales.

Formato de archivo.

Los formatos de archivo de imagen se relacionan más con la información del encabezamiento acerca de cómo leer e interpretar un fichero que con los bits que forman la imagen. Los formatos de archivo cambian en términos de resolución, profundidad de bits, capacidad de color y soporte para compresión y metadatos.

Algunos de los factores que se deben tener en cuenta al elegir un formato de archivo son los siguientes: profundidad de bits aceptadas, técnicas de compresión aceptadas, gestión de color, soporte técnico, capacidad de metadatos, etc. Existen muchos formatos de archivo de imagen, pero los más usados en la actualidad son TIFF – Tagged Image File Format–; GIF –Graphics Interchange Format–; JPEG –Joint Photographic Expert Group–; JFIF TPEG File Interchange Format–; FlashPix; PhotoCD; PNG –Portable Network Graphics–; PDF –Portable Document Format–.

Reconocimiento óptico de caracteres (OCR).

El reconocimiento óptico de caracteres permite la introducción rápida de textos en un ordenador para el acceso electrónico de documentos, gracias a la combinación del uso de un escáner y una aplicación OCR.

Esto es de una gran utilidad cuando en una organización se tiene una cantidad muy grande de información impresa pero sin la fuente en electrónico. El escáner permite obtener una imagen gráfica del texto procesable por medios electrónicos, y el OCR se encarga de la conversión de la imagen a texto. La utilidad de estos sistemas es clara cuando se tiene la necesidad de modificar documentos que no han sido procesados de manera automática; en estos casos, en lugar de teclear todo el documento podría utilizarse un reconocedor.

Actualmente los OCR son capaces de leer y reconocer textos manuscritos o escritos a máquina, de imprenta o de impresora, convirtiéndolos en texto ASCII, editable por el usuario con cualquier programa de tratamiento de textos convencional. Un buen OCR es capaz de leer texto de imprenta en varios tipos de letra, en distintos formatos (columnas, cuadros, etc.) y con caracteres especiales (acentos, signos ortográficos, etc.).

Hasta hace relativamente poco tiempo los OCR eran muy imprecisos y propensos a múltiples errores. Hoy en día se emplean sistemas inteligentes que con aprendizaje interactivo o con ayuda del usuario, consultan diccionarios y se corrigen a sí mismos.

Los OCR se basan en algoritmos de reconocimiento de patrones provenientes de investigaciones en el campo de la inteligencia artificial. Funcionan aislando los posibles caracteres y comparándolos con los patrones de imagen correspondientes a caracteres conocidos hasta que encuentren el de mayor parecido, en un proceso similar al de reconocimiento de huellas digitales. El tipo de algoritmo es lo que marca la diferencia de unos reconocedores a otros.

Dentro de una organización normalmente existe un área o departamento que se encarga de realizar el proceso de digitalización de documentos utilizando los métodos previamente expuestos.

1.5 Acotamiento del problema.

Hoy en día debido a que en la mayoría de las empresas se trabaja o se necesita trabajar con una gran cantidad de documentos e información, es una ventaja muy importante contar con un sistema de gestión documental eficiente que permita un fácil almacenamiento y una búsqueda rápida aunque no todos los documentos tengan la misma importancia.

En los últimos años, las empresas alrededor de todo el mundo tanto privadas como públicas, se han visto afectadas por dos cuestiones fundamentales: la acelerada evolución tecnológica y el aumento de la competencia, en gran medida gracias a la primera cuestión. La gestión de documentos ha pasado de ser poco notorio, a tener una enorme importancia estratégica en la toma de decisiones y en los procesos.

Es importante conocer bien la estructura de la empresa. Los archivos de la empresa algunas veces se conservan con mucha dificultad. La información es diversa, en algunas ocasiones efímeras y no siempre se guarda de la misma manera. Por lo tanto, no solo debemos tomar en cuenta a directivos y trabajadores de la empresa, sino debemos considerar minuciosamente la influencia que estos tienen a la hora de ejercer sus funciones.

El archivo de una empresa es el conjunto de documentos producidos por la misma, en el desarrollo de las actividades que le son propias, de manera organizada y automática, los cuales se conservan como testimonio de la información.

En gran medida los archivos de lo que se guardará registro son administrativos de los cuales hay pocos intermedios y en menor medida históricos.

Realizar un histórico de archivos nos permitirá conocer cuáles fueron las razones para la toma de decisiones en la empresa e investigar todo lo relacionado de ellas y ver su evolución.

Son muchos los parámetros a tomar en cuenta a la hora de poner en marcha un método de organización de archivos. Sabiendo previamente que una empresa tanto privada como pública realiza una gran cantidad de documentos, debemos delimitar todos los archivos a un conjunto, los cuales sean primordiales a la hora de toma de decisiones o en su defecto que estos sirvan para llevar un registro de transacciones.

En estos días los archivos fundamentales en una empresa están divididos en dos categorías:

- Administración.
- Contabilidad.

La administración es la parte fundamental en una empresa ya que es la encargada de llevar la correcta aplicación de los recursos humanos, financieros y materias de la empresa.

Tomando en cuenta el criterio y procedimiento que se debe seguir para cada unidad administrativa que conforma una empresa, conforme al registro, organización, clasificación, resguardo y conservación de los documentos o expedientes que se generen, obtengan, adquieran o conserven, como resultado de la gestión, deberemos tomar en cuenta los siguientes puntos.

- Las fotocopias de documentos y los expedientes multiplicados para controles internos son instrumentos que facilitan la operación administrativa y no serán considerados como documento de archivo.
- Las síntesis informativas, revistas, diarios y otros materiales documentales similares, no se considerarán parte importante del archivo en el expediente.
- Los documentos de fax o impresiones de correo electrónico sin ningún valor legal, solo serán anexados al expediente hasta que se haga entrega de los documentos originales.

- Los expedientes que se encuentren dados de alta previamente o se encuentren sin conclusión, deberán mantener la clasificación con la que anteriormente se hayan dado de alta.

A fin de que estos archivos sean guardados debidamente se contará con una estructura de roles bien definida dentro de la empresa. Para poder tener acceso al sistema habrá de darse de alta previamente y dependiendo de la influencia que éste tenga directa con los documentos podrá realizar acciones determinadas.

La estructura básica quedará de la siguiente manera:

- Administrador: Persona que se encargara de dar mantenimiento al sistema así como dar de alta a las personas que tendrán acceso al mismo.
- Operador: Persona encargada de crear la estructura del folio del expediente, quien subirá los archivos por lo que tendrá privilegios para modificar, crear y actualizar.
- Usuario: Persona final la cual solo tendrá privilegios para poder visualizar los documentos.

El departamento de contabilidad es el encargado de instrumentar y operar las políticas, normas o procedimientos necesarios que garanticen la exactitud y seguridad en el registro de todas las operaciones contables a efecto de suministrar la información necesaria que ayude a la toma de decisiones para la gestión de las operaciones financieras de la empresa, cuidando que dicha contabilización se realice con documentos comprobatorios y justificativos originales, y vigilando la debida observancia de las leyes, normas y reglamentos aplicables.

Así como cada empresa tiene su sistema de administración de documentos también cuentan con diferentes procedimientos para la destrucción de los mismos pero este sistema no intervendrá en ninguno de ellos ya que solo se encargara de mantener la integridad de la información.

En conclusión, hoy en día el manejo de archivos en grandes volúmenes dentro de las empresas actuales es de suma importancia ya que toda la información que manejan en la mayoría de los casos contiene datos relevantes de clientes o acciones realizadas por empleados. El éxito de un sistema se basa en elegir la metodología adecuada para cada uno, con esto podremos analizar el problema de forma teórica paso a paso hasta llegar a la implementación de mismo.

Cabe aclarar que se desarrollará un sistema que capture, gestione y administre toda la información documental que se va generando, día con día en sus oficinas.

Cualquier otro procedimiento como el escaneo, resguardo, etc. Quedará fuera del contexto de este trabajo.

CAPÍTULO II

MARCO TEÓRICO.

2.1 Características, ventajas y desventajas en las bases de datos relaciones, y metodología de Yourdon/De Marco.

2.1.1 Introducción.

Una base de datos es un almacén que nos permite guardar grandes cantidades de información de manera organizada para que posteriormente esta se pueda utilizar fácilmente.

Hoy en día las bases de datos son el método preferido para almacenar datos de manera estructurada, estas van desde grandes aplicaciones para multitud de usuarios, hasta los teléfonos móviles que utilizan esta tecnología para asegurar la integridad de los datos y facilitar el manejo de los mismos.

Una base de datos puede contener más de una tabla las cuales guardan un conjunto de datos. Cada tabla tiene una o más columnas.

Una tabla se asemeja a una matriz donde cada fila de ella representa una tupla, la columna almacena la información de una propiedad determinada de la tabla a la cual se le llama atributo.

Los Sistemas de Gestión de Base de Datos (en inglés Data Base Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El lenguaje ocupado en una base de datos relacional es SQL (Structure Query Language), éste es muy simple ya que solo cuenta con 16 comandos, pero en la práctica puede llegar a ser sumamente complejo. SQL es un lenguaje de alto nivel, se agrupa en tres tipos de sentencias con objetivos particulares, los cuales son:

- Lenguaje de Definición (DDL, Data Definition Language)
- Lenguaje de Manipulación de Datos (DML, Data Management Language)
- Lenguaje de Control de Datos (DCL, Data Control Language)

2.1.2 Características de las bases de datos relacionales.

Modelo de datos relacional.

El objetivo del modelo de datos relacional es el de facilitar que la base de datos sea vista por los usuarios empleando tablas las cuales contienen los datos de la información.

Filas y columnas.

El modelo de datos relacional es una tabla de dos dimensiones que consiste de un número de filas, las cuales contienen un conjunto de columnas. Dentro de una tabla, todas las filas tienen la misma estructura, aunque es posible que en algunas filas algunas columnas puedan no tener nada en ellas.

Tupla.

Una tupla se define como una función finita que asocia unívocamente los nombres de los atributos de una relación con los valores de una instancia de la misma. En términos simplistas, es una fila de una tabla relacional.

Álgebra relacional.

El álgebra relacional se puede describir como el conjunto de operaciones simples que pueden ser aplicadas sobre las tablas de una base de datos relacional, esto al principio puede parecer muy simple pero estas primeras operaciones definen a otras más complejas.

Para crear una relación nueva se usa una o dos relaciones que ya existan y también la nueva relación creada a partir de otras existentes puede ser usada para poder crear otra nueva. El álgebra relacional se apoya principalmente en dos tipos de operaciones,

las unarias y binarias. Las operaciones disponibles dentro del álgebra relacional son las siguientes:

- **Selección:** La operación de selección sirve para elegir un subconjunto de las tuplas de una relación que satisfacen una condición de selección, en pocas palabras esta operación actúa como un filtro que mantiene únicamente aquellas tuplas que satisfacen una condición de cualificación.
- **Proyección:** Es un caso específico de la selección. Si pensamos en una relación como una tabla, la operación selección selecciona algunas filas de la tabla y desecha las demás. La operación proyección, en cambio, selecciona ciertas columnas de la tabla y desecha las demás. Si solo nos interesan ciertos atributos de una relación, entonces usamos proyección para proyectar la relación sobre esos atributos exclusivamente.
- **Unión:** Operación que se encarga de recuperar tuplas de diferentes tablas las cuales se encuentran relacionadas entre sí a través de cláusulas de unión.
- **Diferencia:** Operación encargada de recuperar los registros que cumplan con pertenecer a una relación y que a su vez no pertenezcan a otra relación dentro del mismo esquema.
- **Asignación:** Operación que se encarga de asignar un valor a uno o varios elementos en un campo en específico dentro de una tabla.

Cálculo relacional.

El cálculo relacional es un lenguaje de consulta que describe la respuesta deseada sobre una base de datos pero éste no especifica cómo obtenerla, a diferencia del álgebra relacional.

Tanto el álgebra como el cálculo relacional proporcionan una base para el estudio de la parte manipulativa del modelo relacional. La diferencia entre ellas es que el álgebra proporciona un conjunto de operaciones explícitas que sirve para indicar la forma como construir una relación a partir de otras, y el cálculo solo ofrece una notación para definir la relación deseada en términos de otra relación.

El cálculo relacional se puede clasificar en:

- Cálculo relacional de tuplas.
- Cálculo relacional de dominios.

Ambos tipos son similares, y la diferencia fundamental radica en que las variables del primero representan tuplas y las del segundo representan dominios.

El cálculo relacional se apoya en las siguientes operaciones para obtener una respuesta sobre la base de datos:

- **Reunión:** Permite unir datos de varias relaciones.
- **Diferencia:** Identifica filas que están en una relación y no en otra.
- **Intersección:** Permite unir campos idénticos pertenecientes a las filas que son comunes en dos relaciones.
- **Producto:** Consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles combinaciones entre los registros de la primera y los registros de la segunda.
- Las combinaciones de fórmulas se generan a partir del uso de operadores como NOT, AND, OR.

Normalización.

El proceso de modelado de datos dentro de las tablas relacionales es conocido como normalización. Comúnmente se dice que hay 3 niveles de normalización los cuales se conocen como: primera, segunda y tercera forma normal. Existe un nivel más alto de normalización: cuarta y quinta forma normal las cuales están bien definidas pero no hay que preocuparse mucho de estas últimas formas normales. Es posible para una aplicación SQL trabajar con datos no normalizados aunque esto no suele ser muy eficiente ya que el lenguaje no está diseñado para hacer eso. En la mayoría de los casos, los datos que son almacenados en una base de datos relacional en la cual se

acceda con SQL (Structured Query Langague) debe ser normalizada cuando menos hasta la tercera forma normal.

Primer forma normal (1FN).

En la primera forma normal se deben remover los grupos repetidos o en pocas palabras todas las columnas de todas las tablas deben contener sólo valores atómicos, es decir, un único valor.

Segunda forma normal (2FN).

La tabla debe estar en 1NF y todas las columnas que no son llave deben ser completamente dependientes de la llave primaria. Todos los atributos que no forman parte de la llave deben depender de ella en conjunto y nunca de un subconjunto de la misma.

Tercera forma normal (3NF).

La tabla debe estar en 2FN y cada columna que no es llave depende directamente de la llave primaria. Todos los atributos que no forman parte de la clave primaria deben ser independientes entre sí. Cuando un atributo que no es llave se puede determinar con uno o más atributos que son llave, se dice que existe dependencia transitiva entre ambos, esto significa que una estructura de tipo tabla está en tercera forma normal si no existe dependencia transitiva entre los atributos que no son llaves. La tercera forma normal permite asegurar la eliminación de redundancia debida a las dependencias transitivas.

Cuarta forma normal (4NF).

La 4NF se asegura de que las dependencias multivaluadas independientes estén correcta y eficientemente representadas en un diseño de base de datos. La 4NF es el siguiente nivel de normalización después de la Tercera forma normal.

Una tabla está en 4NF si y solo si está en Tercera forma normal o en BCNF (Cualquiera de ambas) y no posee dependencias multivaluadas no triviales. La

definición de la 4NF confía en la noción de una dependencia multivaluada. Una tabla con una dependencia multivaluada es una donde la existencia de dos o más relaciones independientes muchos a muchos causa redundancia; y es esta redundancia la que es suprimida por la cuarta forma normal.

Quinta forma normal (5NF).

También conocida como forma normal de proyección-uni6n (PJ/NF), es un nivel de normalizaci6n de bases de datos designado para reducir redundancia en las bases de datos relacionales que guardan datos con multi-valores aislando semánticamente relaciones múltiples relacionadas. Una tabla se dice que está en 5NF si y sólo si está en 4NF y cada dependencia de uni6n (join) en ella es implicada por las claves candidatas.

Llave primaria y llave foránea.

Una de las principales características de normalizar los datos es que se debe hacer hincapié en el uso de llaves primarias y llaves foráneas.

Una llave primaria es un identificador único de una fila la cual también puede ser conocida como llave compuesta. Cada tabla debe tener definida una llave primaria. Esto es un requerimiento en el paradigma relacional aunque es posible definir tablas sin llave primaria aunque usualmente esto no es una buena idea ya que algunos RDMS no permiten eso.

Una llave foránea es una columna o uni6n de varias columnas las cuales pueden ser usadas para identificar una relaci6n de una fila con otra tabla. Una llave foránea de una tabla puede ser la llave primaria de otra tabla. Éste es el concepto básico de relaci6n muchos a uno. Una relaci6n muchos a uno es una conexi6n entre dos tablas, donde algunas filas en una tabla hacen referencia a una fila de otra, esto algunas veces puede tomar el nombre de relaci6n padre-hijo donde un padre puede tener muchos hijos.

Paradigma entidad-relación.

Una segunda característica a destacar de la normalización es el paradigma o modelo entidad – relación.

El desarrollo del modelo entidad-relación es el paso previo a la creación de una base de datos, este proceso abarca desde el análisis hasta la implementación, este proceso permite especificar la estructura estática de la aplicación, modela dónde se encontraran así como define la estructura de los datos. Este modelo debe cumplir con los siguientes requisitos:

- **Accesibilidad:** Si los datos no son de fácil acceso provocara que estos no sean utilizados.
- **Precisión:** Todo valor almacenado debe encontrarse dentro de un rango de precisión aceptable con respecto del valor real.
- **Consistencia:** Los datos deben representar de la mejor manera la realidad.
- **Disponibilidad:** Los datos deben poder ser accedidos por la aplicación en cualquier momento.

Este modelo nos permite describir la información que está involucrada en un sistema como un conjunto de entidades y la relación que tienen entre ellas.

Entidad.

Una entidad nos permite presentar la información que debe ser almacenada, la cual puede ser abstraída de cualquier cosa de la realidad tanto tangible como intangible.

En un modelo de Entidad-Relación las entidades y relaciones deben estar en Tercera Forma Normal.

Relación.

Una relación nos permite representar un conjunto de vínculos lógicos entre instancias de dos o más entidades. Una entidad puede participar en más de una relación las

cuales se pueden clasificar en diferentes tipos en base a las participaciones de las entidades.

Relación uno-a-uno

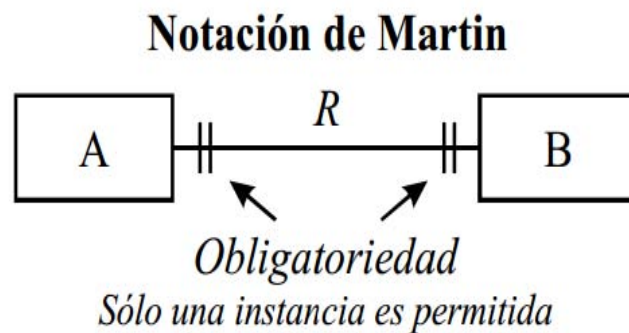


Figura 2.1.2.1. Relación uno a uno con notación de Martin.

En la figura 2.1.2.1 podemos observar lo siguiente:

- Cada ocurrencia de la Entidad A está relacionada a una y solo una ocurrencia de la entidad B y viceversa.

Relación uno-a-muchos.

Las relaciones de varias instancias de una entidad se presentan por medio del signo menor.

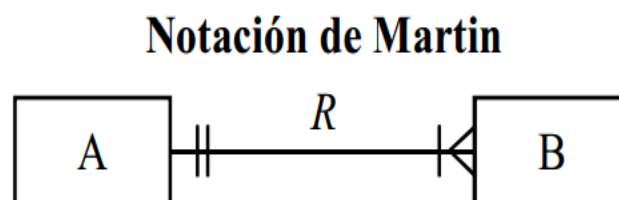


Figura 2.1.2.2 relación uno a mucho con la notación de Martin.

En la figura 2.1.2.2 se puede observar lo siguiente:

- Cada ocurrencia de la entidad A está relacionada a uno o varias ocurrencias de la entidad B.
- Cada ocurrencia de la entidad B está relacionada a uno y solamente a una ocurrencia de la entidad A.

Relación Muchos-a-Muchos.

Dos relaciones uno-a-muchos para ambos lados pueden existir entre entidades, ellas se convierten en una sola relación llamada muchos-a-muchos (figura 2.1.2.3).

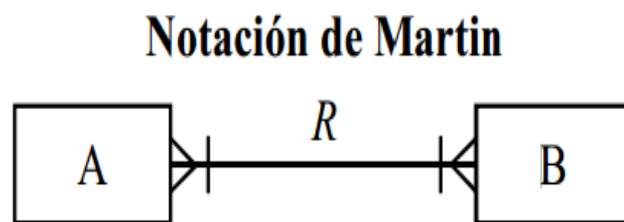


Figura 2.1.2.3 Relación muchos a muchos con notación de Martin.

En una relación muchos a muchos se debe cumplir lo siguiente:

- Cada ocurrencia de la entidad A está relacionada con una o varias ocurrencias de la entidad B y viceversa.
- Cada ocurrencia de la entidad B está relacionada con una o varias ocurrencias de la entidad A.

Ventajas de las base de datos relacionales.

- La búsqueda de información es rápida y ágil.
- Manejo sencillo de la información a través de estructuras llamadas tablas.
- Independencia de la estructura, lo que nos garantiza que al cambiar el Hardware no debe haber cambio en nuestra base de datos.

- Se hace uso de modelos los cuales nos permiten abstraer objetos tanto tangibles como intangibles de la realidad y proyectarlos en tablas.
- Hay independencia de datos, es decir un cambio de datos no implica un cambio en nuestro programa.
- Garantizan la integridad de los datos.
- Provee reglas de normalización para obtener modelos más eficientes.
- Aumento en la concurrencia, los SGBD se encargan de gestionar el acceso concurrente a la base de datos y garantizan que no ocurran problemas.
- Mejora en el mantenimiento.
- Poseen estándares para el formato de datos facilitando su documentación, actualización y reglas de acceso.

Desventajas de las bases de datos relacionales.

- Vulnerabilidad a fallos al estar todo centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos.
- Pueden ser de instalación compleja.
- Un mal diseño de la base de datos puede implicar un proceso de normalización complejo.

2.1.3 Metodologías de desarrollo de software.

En el desarrollo de sistemas los diagramas representan una parte fundamental, ya que se utilizan para definir de manera correcta sus especificaciones además de representar de manera gráfica el diseño en el desarrollo de un sistema.

SSADM.

SSADM (Structured System Analysis and Design Method), metodología de aproximación en cascada para el desarrollo de sistemas de información. Esta metodología estándar de desarrollo de proyectos del gobierno del Reino Unido.

Los aspectos claves de esta metodología son:

- Énfasis en los usuarios, sus requisitos y participación.
- Definición del proceso de producción, qué hacer, cuándo y cómo.
- Tres puntos de vista: datos, eventos y procesos.
- Máxima flexibilidad en herramientas y técnicas de implementación.

SSADM proporciona un conjunto de procedimientos para llevar a cabo el análisis y diseño, pero no cubre la planificación estratégica y tampoco está diseñada para realizar el mantenimiento de las aplicaciones, lo que hace es suministrar la documentación completa y precisa para poder mantener el sistema fácilmente.

SSADM tampoco cubre aspectos como la planificación estratégica ni entra en la construcción del código. Ver figura 2.1.3.1

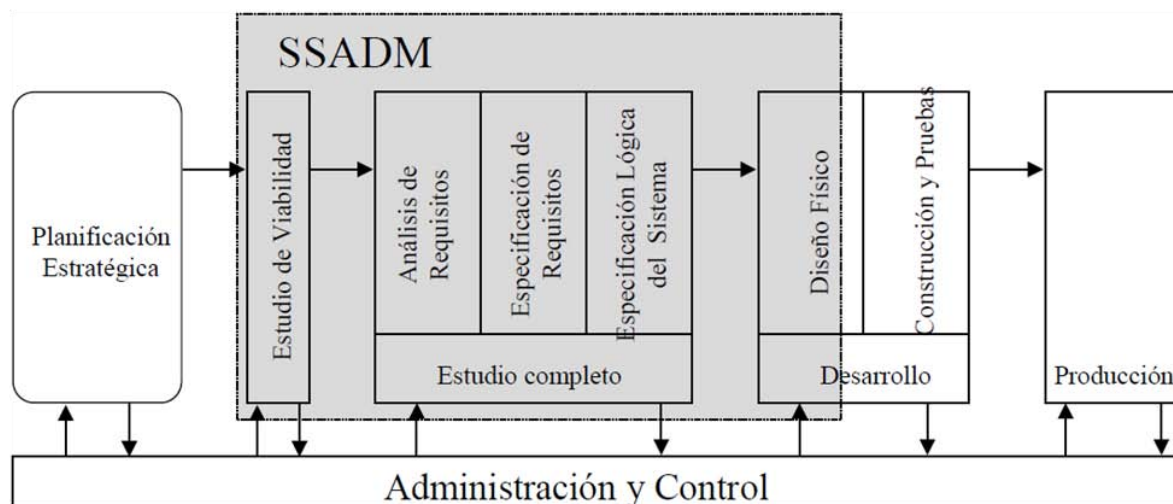


Figura 2.1.3.1 Metodología SSADM

Metodología Merise.

MERISE es una metodología de la Administración francesa, creada por iniciativa del Ministerio de Industria Francés y desarrollada por Tardieu, Rochfeld y Colletti. Para el Ministerio de Industria Francés.

La Metodología de Merise introduce dos ciclos complementarios: ciclo de abstracción y ciclo de decisión.

El ciclo de abstracción se basa en tres niveles:

- Nivel conceptual: Nivel en el cual se define los objetivos y limitaciones. Se realiza en esta fase un tratamiento de los datos según el modelo conceptual de datos y los procesos según el modelo conceptual de procesos.
- Nivel organizativo: Es donde se define la organización adecuada que hay que implantar para poder alcanzar los objetivos asignados y se realiza un tratamiento de los datos y el modelo organizativo de tratamientos para la realización de los procesos.
- Nivel físico: Se realiza la integración de los medios técnicos necesarios para el proyecto, utiliza el modelo físico de datos para los datos y el modelo operativo de tratamiento para los procesos.

Merise cubre las cuatro fases fundamentales en el ciclo de vida del desarrollo de software: estudio preliminar, análisis, diseño e implementación. Esta metodología no está diseñada para realizar el mantenimiento de las aplicaciones, solo suministra documentación completa para poder mantener el sistema. Ver figura 2.1.3.2.

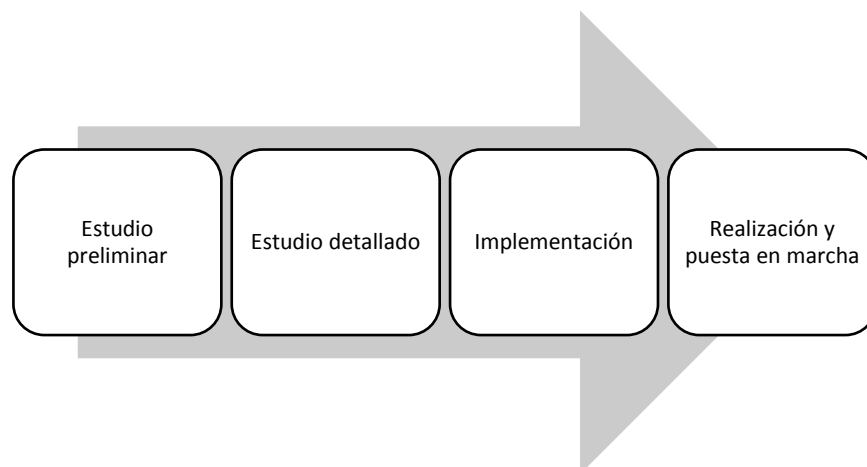


Figura 2.1.3.2 Fases del ciclo de vida del desarrollo de software según Merise.

Yourdon/De Marco

La Metodología de Yourdon nos proporciona un método para realizar el diseño paso a paso de manera detallada en el desarrollo de un sistema. Estos pasos involucran desde el análisis, desarrollo del diseño y posibles mejoras a futuro del mismo diseño.

Esta metodología consta de cinco fases las cuales se indican en la figura 2.1.3.3

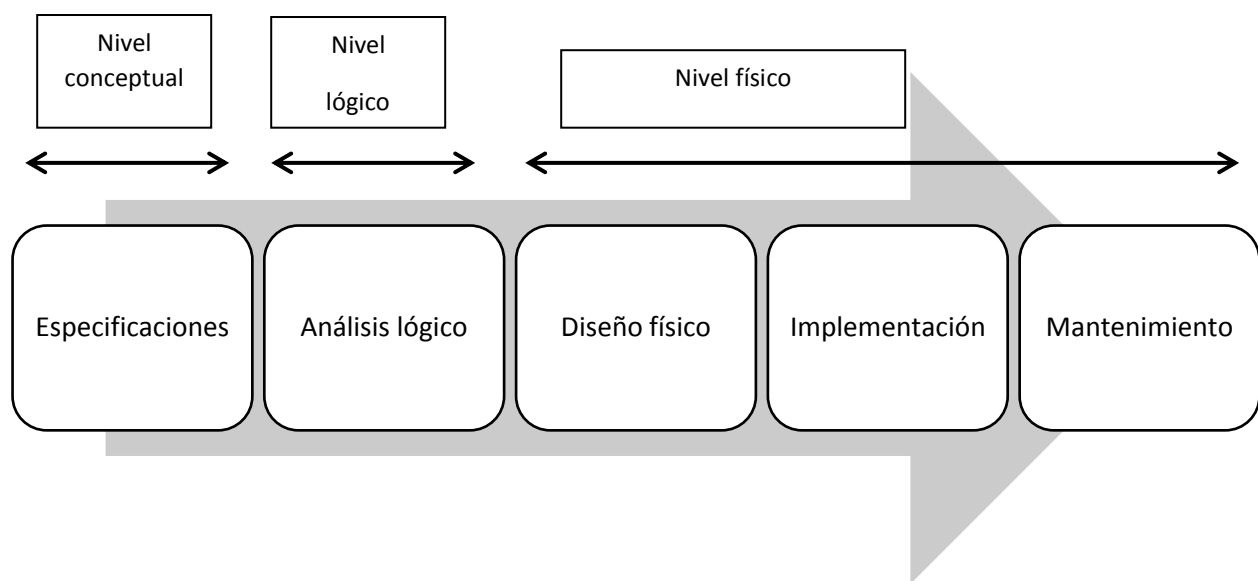


Figura 2.1.3.3 Fases del ciclo de vida de un sistema con base en la metodología Yourdon/De Marco

A lo largo de la metodología las actividades que deben ser cumplidas son:

Estudio de viabilidad: Este estudio tiene como finalidad comprobar si existe una solución real del proyecto la cual cumpla los objetivos definidos en el análisis inicial y determinar si es viable.

Análisis del sistema: En esta fase se representa mediante herramientas graficas todo el sistema a desarrollar.

Diseño: Tercer fase, se pasa del nivel conceptual descrito previamente a un nivel de representación lógica de los datos mediante el modelado de la base de datos y la estructuración de los procesos empleando diagramas para describirlos detalladamente.

Implementación: Esta fase comprende toda la generación de código necesaria para desarrollar cada módulo del sistema e integrarlos todos para obtener el sistema con funcionalidad completa.

Pruebas: Fase encargada de realizar las pruebas necesarias tanto de integración y funcionamiento del sistema.

Documentación: En este paso se deberá generar toda la documentación necesaria para llevar a cabo la instalación del sistema. Las especificaciones del sistema que se crearán en pasos anteriores y con las cuales los programadores desarrollaran los módulos del sistema.

Las principales herramientas en esta metodología son:

- Diagrama de procesos.
- Diagrama de flujo de datos.
- Modelo entidad relación.
- Diccionario de datos.
- Normalización.

2.2 Características, ventajas y desventajas de JavaScript, jQuery y XHTML.

2.2.1 Introducción de JavaScript

En los últimos años se ha tenido un crecimiento exponencial de Internet ya que diariamente hay más usuarios conectados y al paso del tiempo los servicios de Web se vuelven más elaborados y complejos.

Por ello los desarrolladores y analistas crean nuevas metodologías para difundir grandes cantidades de información, optimizando los recursos, incluso que permitan obtener una Web dinámica y mejoras en la interfaz de usuario.

Netscape es el origen de JavaScript debido a que fue el primer navegador por expandir la funcionalidad de un navegador Web. JavaScript fue diseñado para ser un lenguaje de elaboración de scripts que pudieran integrarse en archivos HTML que pudieran ser interpretados por el navegador sin la necesidad de compilarse previamente como es el caso de java. Inicialmente se llamó LiveScript posteriormente la empresa Sun Microsystems lo renombró como JavaScript. Vea figura 2.2.1.



Figura 2.2.1 Logotipo de JavaScript

Características de JavaScript.

- Es un código que es interpretado, no compilado.
- Es un lenguaje orientado a objetos que posee una gama de métodos.
- Maneja eventos, en otras palabras JavaScript podrá reaccionar ante cualquier número de eventos que se presenten (carga y descarga de páginas, arrastrar o apuntar con el mouse, envío de formulario, etc) de acuerdo a como el programador lo indique.
- Las referencias del objeto se verifican durante el tiempo de ejecución.

A continuación se mencionan algunas ventajas y desventajas del uso de JavaScript con respecto a otros lenguajes de programación.

Ventajas de JavaScript

- Es independiente de la plataforma.
- No requiere de declaración de variables.
- Las referencias del objeto se verifican durante el tiempo de ejecución.
- Al ser un lenguaje de programación orientado a objetos (POO), permite utilizar la herencia. Consiste en establecer relaciones entre diversas clases que conforman el sistema. Gracias a ello permite la reutilización de código.
- Al ser un lenguaje de programación orientado a objetos (POO), permite utilizar polimorfismo, es la capacidad de tener métodos del mismo nombre pero su implementación se lleva a cabo de manera diferente.
- Permite integrar Frameworks como lo es jQuery.

Desventajas de JavaScript:

- Limitado por la versión y el tipo de navegador en donde será ejecutado
- Requiere de una consola previamente instalada para poder ser ejecutado.
- Solo permite manipular objetos conocidos por el navegador.

- Los fragmentos de código de JavaScript una vez añadidos a las páginas web en los servidores, son descargados y ejecutados en el navegador del cliente permitiendo así que cierto código malicioso pueda ser ejecutado en la máquina del cliente con el objetivo de explotar alguna vulnerabilidad de seguridad.

2.2.2 Introducción de jQuery.

jQuery vea figura 2.2.2, es un framework JavaScript de carácter gratuito, que es uno de los más populares en la comunidad de desarrolladores por que combina su facilidad de uso y su gran potencia. Un framework es un conjunto de bibliotecas de código que contienen procesos o rutinas que garantizan un correcto funcionamiento reduciendo muchas líneas de código. Es ejecutado a nivel cliente y en su mayoría es implementado en aplicaciones Web para desplegar contenido dinámico.



Figura 2.2.2 Logotipo de jQuery

jQuery fue publicado por primera vez en Enero del 2006 en “BarCamp NYC” por John Resign.

Características de jQuery

- Es de carácter gratuito
- Posee una gran cantidad de objetos disponibles para su uso.
- Los objetos, documentos mostrados en la ventana, funciones y el manejo de eventos son ejecutados del lado del cliente haciendo uso de JavaScript.
- Simplifica la interacción con documentos HTML.

- Sus funcionalidades y desarrollo están basados en JavaScript.
- Compatible con la mayoría de los navegadores existentes.
- Interactividad y modificaciones del árbol DOM (Modelo de Objetos del Documento) e incluso cuenta con el soporte para CSS.

A continuación se mencionan algunas ventajas y desventajas del uso de la biblioteca de jQuery.

Ventajas de jQuery

- Permite al desarrollador concentrarse más en el diseño del sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores disponibles en el mercado.
- Permite manipular series de elementos y modificarlas con una simple línea de código.
- jQuery es muy fácil de expandir, ya que cuenta con gran cantidad de plug-ins que se pueden utilizar o hasta crear uno propio.
- Compatible con todos los navegadores modernos.
- Al ser un código abierto el desarrollador puede modificar el código fuente y ajustarlo a las necesidades del negocio.
- Permite desarrollar animaciones.
- Tiene una excelente comunidad de soporte.
- Puede ser integrado con código AJAX y con hojas de estilo CSS.

Desventajas de jQuery

- Si es implementado inapropiadamente como un Framework, el entorno de desarrollo se puede salir de control.
- Tiene una gran cantidad de versiones en un periodo de tiempo muy corto, por lo que se requiere de la instalación constante de actualizaciones. Por consecuencia esto aumenta la probabilidad de que haya problemas de incompatibilidad.

- Su código principal es difícil de entender.
- Su hoja de estilo - CSS es complejo a la hora de manipularlo.

2.2.3 Introducción de XHTML

HTML es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

XML es una especificación para diseñar lenguajes de marcado, que permite definir etiquetas personalizadas para descripción y organización de datos. Asegurando representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.

Debido al uso cada vez más generalizado de dispositivos de navegación, fue necesario crear herramientas que permitan el manejo de la información de una manera estandarizada. Por esta razón surge XHTML (eXtensible Hyper Text Markup Language o Lenguaje Extensible de Marcado de Hipertexto) vea figura 2.2.3, con un etiquetado más estricto que HTML, que permite una correcta interpretación de la información, independientemente del dispositivo con que se accede a ella.



Figura 2.2.3 Logotipo de XHTML

XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium (W3C), de lograr una Web homogénea, en la que la información y la forma de presentarla estén separadas claramente. XHTML serviría únicamente para transmitir la información contenida en un documento, dejando a las hojas de estilo y JavaScript el aspecto y el diseño para distintos medios.

Características de XHTML

- Independencia de interpretación sobre el dispositivo y plataforma.
- Creado para lograr una Web homogénea.
- Formado por etiquetas HTML y XML.
- Lenguaje no compilado sino interpretado.
- Permite incrustar código Javascript.
- Lenguaje de carácter libre.
- Amplia documentación por parte de los desarrolladores.
- Lenguaje fácil de implementar y usar.

A continuación se mencionan algunas ventajas y desventajas del combinar las etiquetas HTML bajo la sintaxis de XML.

Ventajas de XHTML

- La rigidez de XHTML ayuda a que el código sea coherente, bien estructurado, y etiquetas que se ajusten al estándar, lo que a la larga facilita la aplicación de formato mediante CSS.
- XHTML es un paso lógico en la transición de HTML a XML, usa las etiquetas tradicionales de HTML pero con la sintaxis moderna de XML.
- Insistencia en los estándares, lo que hace más probable que se soporte de forma coherente en varios navegadores y en varias plataformas en un futuro.

Desventajas de XHTML

- Hay pocos editores de ayuda que escriban correctamente XHTML.
- Si una página está hecha usando XHTML y contiene algún error, simplemente no se mostrará en el navegador.
- Algunos navegadores “antiguos” aún no son totalmente compatibles con los estándares.

2.3 Características, ventajas y desventajas de la arquitectura Web.

Introducción.

Debido al gran éxito que ha tenido la WWW (World Wide Web) el desarrollo de aplicaciones Web ha crecido de forma notable abarcando áreas como comercio electrónico, redes sociales, banca en línea, entretenimiento, etc. La mayoría de los grandes sistemas de información han tenido que ser trasladados a ambientes Web como parte de su evolución.

Las aplicaciones Web usan la infraestructura de la Web (protocolos, lenguajes, etc.) para su funcionamiento. Hoy en día las aplicaciones Web han crecido hasta convertirse en grandes sistemas distribuidos complejos y que pueden atender a millones de usuarios de forma simultánea.

Las aplicaciones Web utilizan tecnología basada en Web como son los navegadores y los servidores Web. Esto permite tener el acceso a un mundo de aplicaciones por medio de una sola interfaz que es el navegador.

Las aplicaciones Web han ganado terreno sobre las aplicaciones tradicionales (de escritorio) por las siguientes razones:

- Las aplicaciones Web son más accesibles al uso de protocolos estándar como HTTP, toda aplicación Web puede ser usada por toda la Web sin importar ubicación y plataforma.
- Las aplicaciones Web son más fáciles de mantener desde el momento en que una aplicación se despliega en el servidor, los clientes pueden acceder a ella, sin necesidad de una aplicación cliente. Esto permite que cualquier cambio sólo se haga en el servidor y no en el cliente ahorrando tiempos de desarrollo.

Características.

Existen cuatro tipos de aplicaciones Web:

- Estáticas este tipo de aplicaciones representan a la primera generación de páginas Web estáticas imágenes y texto, ejemplo páginas de consulta.
- Orientadas a servicios este tipo de aplicación ofrece una lógica de negocios acorde al servicio ofrecido.
- De datos, este tipo de aplicaciones está enfocado a proveer una interfaz para acceder a una gran cantidad de datos, catálogo en línea de las bibliotecas.
- Sistemas de información, combinan las aplicaciones orientadas a servicios de datos. Esta última será utilizada para el desarrollo.

Dentro de los desarrollos Web, existen varias arquitecturas, partiremos de uno genérico para aplicarlo en el sistema.

Esta arquitectura genérica contiene todos los componentes necesarios para el desarrollo de aplicaciones del dominio Web. A diferencia de las arquitecturas de aplicaciones locales, esta arquitectura debe contener componentes basados en Web que permiten aprovechar al máximo su infraestructura.

Requerimientos.

Ciente servidor.

En una aplicación Web sigue un modelo cliente-servidor. Esto se debe a que se requiere un componente que hospede a la aplicación y a todos sus componentes (servidor Web) y un componente cliente, en nuestro caso un navegador Web. Una vez que tenemos estos dos componentes necesitamos un sistema de archivos en donde se van a almacenar todos los recursos para la generación de contenido. El sistema de archivos puede almacenar una gran cantidad de archivos multimedia (fotos, videos, audio, etc.), archivos XML y HTML, necesarios para el funcionamiento de la aplicación. La primera propuesta de arquitectura para aplicaciones Web se puede observar.

Arquitectura para aplicaciones Web estáticas.

La arquitectura propuesta satisface los requerimientos de aplicaciones Web estáticas que basan su contenido en archivos HTML estáticos. Sin embargo, para que podamos modelar aplicaciones Web dinámicas necesitamos agregar componentes adicionales tales como un servidor de aplicaciones, vea figura 2.3.1

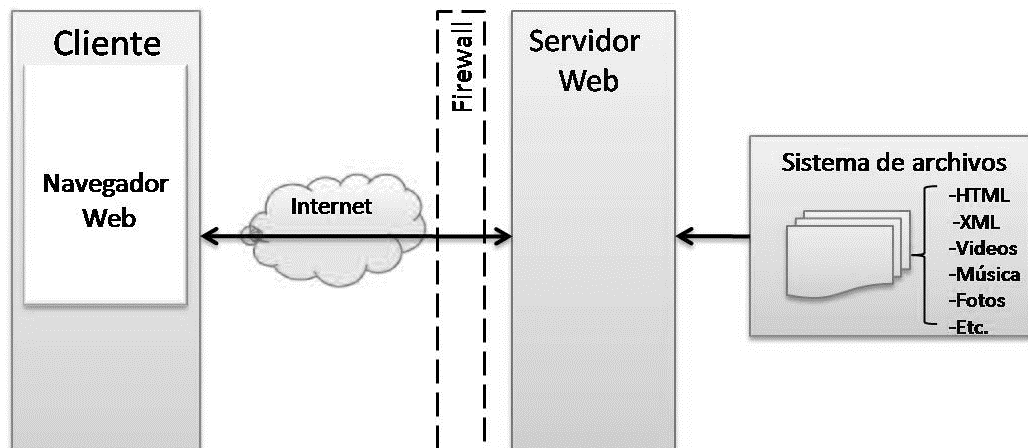


Figura 2.3.1 Arquitectura para Web estáticas

Permitir la generación de contenidos dinámicos tomando como base la arquitectura que estamos proponiendo. Al agregar este nuevo componente necesitamos considerar las interfaces que se necesitan incluir así como la relación con los componentes existentes, vea figura 2.3.2

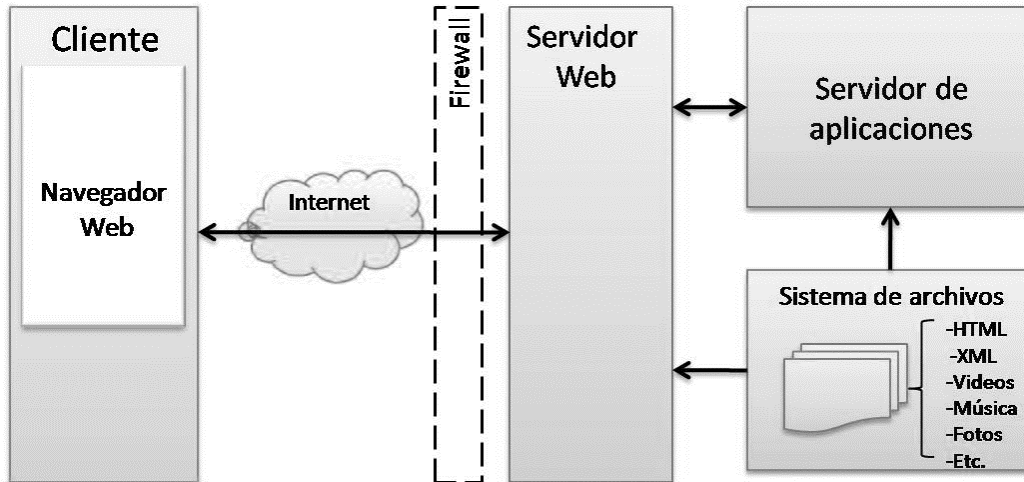


Figura 2.3.2 Arquitectura para aplicaciones Web dinámicas.

Ventajas de aplicaciones Web.

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil) porque se accede a través de una página Web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso de las aplicaciones Web a través de teléfonos móviles requiere sin embargo un diseño específico de los

ficheros CSS para no dificultar el acceso de estos usuarios.

- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas.

Desventajas de aplicaciones Web.

- Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.
- La disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

El servidor Web Apache.

Apache es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language), estáticas o dinámicas en la World Wide Web. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. Ver figura 2.3.3.



Figura 2.3.3 Logo servidor Web Apache

Ventajas.

- Es multiplataforma, se puede implementar en plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras que implementa el protocolo HTTP/1.1
- Es un servidor Web de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
- Posee una gran comunidad que ayuda para dar soporte al producto.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Desventajas.

- Puede resultar complejo al configurarlo incluso para tareas sencillas y por ello muchos novatos le escapan a su uso. Apache fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

- Al manejar formatos de configuración no estándar, esto dificulta un poco la automatización y el procesamiento de la configuración al no estar basada esta en formatos más soportados como el XML.
- Al ser un producto multiplataforma, el servidor no aprovecha al máximo las posibilidades que ofrece el sistema operativo.
- En la administración del servidor como la mayoría de los programas open source, uno depende de configurar los archivos a mano o tener que instalarse herramientas adicionales para las tareas de administración.

2.4 Características, ventajas y desventajas de la base de datos SQL-Server.

Introducción.

Microsoft SQL Server ver figura 2.4.1 es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional (SGBDR o RDBMS, Relational Database Management System) diseñado para trabajar con grandes cantidades de información y tiene la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios Web. Antecedentes de SQL Server.



Fig. 2.4.1 Logo SQL Server

SQL fue creado por IBM en los años 70's, basado en los conceptos formulados por el doctor Edgar Codd sobre bases de datos relacionales, pero es estandarizado por el ANSI American National Standards Institute (Instituto Nacional Americano de Estándares) hasta 1986, durante estos años se desarrolló en Sybase para sistemas UNIX, posteriormente la versión 4.21 de SQL Server fue lanzada con Windows NT 3.1, a partir de la versión 6.0 se han lanzado versiones desarrolladas independientemente de Sybase.

Cuando Microsoft lanza en 1998 la versión 7.0 implementa el modelo cliente-servidor dando así un paso adelante en rendimiento, escalabilidad y fiabilidad. La primera versión con una arquitectura a 64 bits fue la 2000. Más tarde, se puso en marcha Microsoft SQL Server 2005 o 9.0. Posteriormente Microsoft lanza la nueva versión SQL Server 2008 o versión 10.0 ofreciendo una plataforma de gestión de datos óptima, a la cual podemos acceder desde cualquier lugar y en cualquier momento. En el año 2010 es lanzado SQL Server 2008 R2, algunos la llaman SQL Server 2010, pero es solo una versión mejorada de la 2008.

SQL Server 2012, proporciona capacidades de centro de datos de tecnología avanzada completas con un rendimiento ultrarápido, virtualización ilimitada y Business Intelligence(Inteligencia de Negocios) integral, que habilita los mayores niveles de servicio para las cargas de trabajo de gran importancia y el acceso del usuario final a ideas claras de los datos.

El Sistema de Gestión de Bases de Datos utilizado para el desarrollo del Sistema de Administración Documental, es “SQL Server 2012, Enterprise Edition a 64 bits”.

Características.

Como ya lo hemos indicado una base de datos relacional es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, éstas a su vez están compuestas por filas y columnas. En el centro de SQL Server está el motor SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente las conexiones y sentencias SQL.

SQL Server incluye un conjunto de herramientas que facilitan la instalación y administración del servidor así como un conjunto de herramientas que facilitan el diseño e implementación de base de datos, entre ellos podemos mencionar:

- SQL Server Database Engine, diseñado para almacenar detalladamente los registros de las operaciones transaccionales, este motor es responsable de mantener la seguridad de los datos, proveen un adecuado nivel de tolerancia a fallos, optimiza las consultas, emplea adecuadamente los bloqueos de recursos para optimizar la concurrencia.
- SQL Server Analysis Services, provee herramientas para consultar información almacenada en datawarehouses y data marts, como por ejemplo cuando se desea obtener información totalizada acerca de los niveles de ventas mensuales por regiones de ventas.

SQL Server dispone de protección total para estos entornos, con medidas de seguridad que evitan problemas como tener varios usuarios intentando actualizar los mismos datos al mismo tiempo. SQL también asigna de manera muy eficaz los recursos disponibles, como memoria, ancho de banda de la red y E/S del disco, entre los distintos usuarios.

Las aplicaciones de SQL Server se pueden ejecutar en el mismo equipo que SQL Server. Esto permite utilizar SQL Server en sistemas pequeños en los que una aplicación debe almacenar los datos localmente.

El motor de la base de datos es el corazón del DBMS, responsable de la estructuración, almacenamiento y recuperación de los datos del disco. Acepta peticiones SQL de otros componentes del DBMS, tales como facilidades de formularios, generadores de informes o facilidades de consultas interactivas, de programas escritos por los usuarios e incluso de otros sistemas informáticos.

Otras características de SQL Server:

- Incluye un ambiente gráfico de administración. Permite el uso de comandos DDL (Lenguaje de Definición de Datos) y DML (Lenguaje de Manipulación de Datos) gráficamente.

- Manejo de triggers INSTEAD OF que sustituye comandos como insert, update o delete, y AFTER que se ejecuta una vez terminada la condición que lo originó.
- Conectividad. Integración con Internet mediante el soporte de XML.
- Lenguaje declarativo de alto nivel. También llamado de “no procedimientos”, se enfoca al manejo de conjuntos de registros, es decir una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel.
- Permite el uso de comandos LDD (Lenguaje de definición de datos) y LMD (Lenguaje de Manipulación de Datos).
 - El LDD de SQL proporciona comandos para la definición y modificación de esquemas de relación, así como borrado de relaciones, mediante sus operaciones básicas: CREATE, ALTER, DROP Y TRUNCATE.
 - El LMD de SQL es un lenguaje proporcionado por el sistema de gestión de bases de datos que permite la consulta y manipulación de datos.

Estas dos categorías de sentencias para manejo de datos se encuentran dentro del Lenguaje Transact-SQL (T-SQL) que es el principal medio de programación y administración de SQL, todas las aplicaciones que se comunican con SQL Server lo hacen enviando instrucciones Transact-SQL al servidor, independientemente de la interfaz de usuario de la aplicación.

- Soporte de transacciones. Es capaz de mantener la integridad de los datos, haciendo que los conjuntos de órdenes, llamados transacciones, no puedan finalizar en un estado intermedio. Cuando por algún motivo el sistema debe cancelar, comienza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial llamado punto de integridad, como si la orden de transacción nunca se hubiera realizado.
- Escalabilidad. Puede escalar de manera más efectiva el uso de recursos disponibles y la administración de las cargas de trabajo.
- Servicios de análisis. Funciones de procesamiento analítico en línea (OLAP), permitiendo al usuario crear y administrar estructuras multidimensionales de datos agregados de bases de datos relacionales y de minería de datos.

- Seguridad. Se pueden buscar datos encriptados a través de una búsqueda por rango o aleatoria, buscar datos seguros entre usuarios no autorizados y la posibilidad de encriptar datos sin la necesidad de realizar cambios en las aplicaciones existentes.
- Cliente-servidor. Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Generador de reportes. Genera fácilmente reportes con cualquier estructura, a través de un diagramador de reportes, además permite que los usuarios consuman reportes directamente desde las distintas aplicaciones de Microsoft Office.

Ventajas.

- Vinculación entre servidores. Es posible mediante el AMSM Application and Multi-Management Server (aplicación y manejo de varios servidores), que permite el funcionamiento entre múltiples servidores con una consulta.
- Soporte para datos estructurados y semi-estructurados. Incluye los formatos de medios digitales para imágenes, audio, video y otros medios multimedia. “Esto lo convierte en un motor de almacenamiento de datos para las diferentes variedades de datos.”
- Database mirroring (base de datos espejeada).- Permite hacer una réplica exacta de la base de datos, además de que se puede realizar un fail-over automático siempre y cuando se tenga instalado un tercer servidor el cual informa al servidor espejo que el servidor principal ha sufrido alguna anomalía y que la aplicación se va a reconectar a este servidor.
- Compresión del flujo de registros. Ofrece un rendimiento óptimo y reduce el ancho de banda de la red utilizada por el database mirroring de datos.
- Unicode compression. Reduce la cantidad de espacio en disco utilizado por los caracteres Unicode, se realiza automáticamente y es administrado por el motor de SQL Server, por lo que no son necesarios cambios de programación.
- Manejo de datos FILESTREAM. Permite almacenar BLOBs Binary Large Objects (objetos binarios grandes) en un sistema de ficheros NTFS New Technology File

System (Nueva Tecnología de Sistema de Archivos), es decir ficheros o documentos fuera de la base de datos, de forma transparente esto permite un acceso más rápido a dicha información, y adicionalmente, permite liberar de carga de memoria a SQL Server.

- Los datos null no consumen espacio físico. Lo cual supone una manera muy eficiente de administrar datos vacíos en una base de datos.
- Manejo de tablas particionadas. Las particiones permiten el manejo de tablas más grandes y crecientes de una manera más eficiente, ya que las fragmenta en bloques de datos administrables.
- Madurez de una solución que lleva muchos años de innovación y desarrollo dentro del área de los sistemas gestores de bases de datos. Cuenta con el soporte de una de las empresas más poderosas y con mayor alcance dentro del ámbito de las TI. Interactúa con las herramientas de desarrollo Microsoft de una manera fácil, amigable y transparente.

Desventajas.

- Está atado a la plataforma de sistema operativo Microsoft Windows, por lo que no puede instalarse, en servidores Linux.
- Los requerimientos del sistema para su instalación son bastante altos, al igual que los costos por licenciamiento comparadas con otros competidores.
- No se puede utilizar el almacenamiento FILESTREAM en una base de datos que esté utilizando database mirroring.
- Debido a las múltiples herramientas que ofrece, es frecuente que no haya disponibilidad de administradores íntegros de la base de datos y una sola persona termina responsabilizándose de una gran cantidad de bases de datos.
- El uso de índices acelera la instrucción de consulta, pero retrasa la actualización de los datos.
- Requiere de mantenimiento constante, ya que con el paso del tiempo los índices de la base de datos se fragmentan, lo que produce una degradación del rendimiento.

2.5 Características, ventajas y desventajas de la arquitectura cliente/servidor.

Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Normalmente los servidores son ordenadores potentes dedicados a gestionar unidades de disco (servidor de archivos), impresoras (servidor de impresión), tráfico de red (servidor de red), datos (servidor de base de datos) o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.

Esta arquitectura implica la existencia de una relación entre los procesos que solicitan servicios (clientes) y procesos que respondan a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos; esto es, cuando hacemos referencia a clientes y servidores, nos referimos a los procesos lógicos en vez de a computadoras físicas sobre las que se ejecutan.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes. El cliente (también llamado front-end) interactúa con el usuario usualmente en forma gráfica (hace interfaz entre el usuario y el resto de la aplicación) mientras que el servidor o back end interactúa con los recursos compartidos (bases de datos, impresoras, modem).

La arquitectura cliente servidor implica la realización de aplicaciones distribuidas. La principal ventaja de esta arquitectura es que permite separar las funciones según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución.

Ventajas de la arquitectura cliente/servidor.

- La parte cliente y servidor tienen diferentes necesidades de recursos a la hora de ejecutarse; velocidad de procesador, memoria, velocidad y capacidad en disco duro,

dispositivos de entrada/salida, etc. por lo que se puede aumentar la capacidad de uno u otro por separado.

- Cualquier elemento de la red puede ser aumentado o mejorado en cualquier momento o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- El entorno puede ser heterogéneo; es decir, el hardware y el sistema operativo del cliente y del servidor suelen diferir lo cual permite a las organizaciones ahorrar costos si alguna plataforma es económicamente más accesible y le arroja los mismos resultados.
- Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio
- Las redes de ordenadores permiten que múltiples procesadores puedan ejecutar partes distribuidas de una misma aplicación, logrando concurrencia de procesos.
- Existe la posibilidad de migrar aplicaciones de un procesador a otro con modificaciones mínimas en los programas.
- Se obtiene una escalabilidad de la aplicación. Permite la ampliación horizontal o vertical de las aplicaciones. La escalabilidad horizontal se refiere a la capacidad de añadir o suprimir estaciones de trabajo que hagan uso de la aplicación (clientes), sin que se afecte sustancialmente al rendimiento general. La escalabilidad vertical permite la migración hacia servidores de mayor velocidad o de un tipo diferente.

Desventajas de la arquitectura cliente/servidor.

- La congestión del tráfico es el problema más común en una arquitectura cliente servidor. Cuando una gran cantidad de clientes envía peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor) por lo que este último puede convertirse en un cuello de botella.
- Cualquier sistema distribuido tiende a fallar con más frecuencia que un sistema centralizado, ya que tiene más componentes que tienden a presentar imperfecciones de una manera independientemente.

- El software distribuido es más complejo que el no distribuido y también es más difícil probarlo y depurar errores; también la administración y la seguridad son más complejas.

Separación de funciones.

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles:

- Lógica de presentación. La presentación está relacionada con la presentación de la información al usuario y toda la interacción con él.
- Lógica de negocio (o aplicación). Los flujos de trabajo pueden cambiarse según las necesidades existentes de un procesador a otro.
- Lógica de datos. La gestión de los datos está relacionada con todas las operaciones sobre la base de datos la cual debe ser independiente para poder ser distribuida según las necesidades de la organización en cada momento.

Si un sistema distribuido se diseña correctamente, los tres niveles anteriores pueden distribuirse y redistribuirse independientemente sin afectar el funcionamiento de la aplicación.

Según como se distribuyan las tres funciones básicas de una aplicación (presentación, negocio y datos) entre el cliente y el servidor podemos contemplar:

- Presentación distribuida. El cliente solo mantiene la presentación, el resto de la aplicación se ejecuta remotamente. La presentación distribuida, en su forma más simple, es una interfaz gráfica de usuario a la que se pueden acoplar controles de validación de datos, para evitar la validación de los mismos en el servidor.
- Aplicación distribuida. Es el modelo que proporciona la máxima flexibilidad, puesto que permite tanto al servidor como al cliente mantener la lógica de negocio realizando cada uno las funciones que le sean más propias, bien por organización, o bien por mejora en el rendimiento del sistema.

- Datos distribuidos. Los datos son los que se distribuyen, por lo que la lógica de datos es lo que queda separada del resto de la aplicación. Se pueden dar de dos formas: archivos distribuidos o bases de datos distribuidas.

Capas en la arquitectura cliente servidor.

Arquitectura de una capa.

La arquitectura más simple dentro del concepto cliente servidor es la que se conoce de una capa, en la cual el usuario interactúa con una computadora la cual funge como servidor como se muestra en la figura 2.5.1, ya que en ella se mantienen todos los recursos informáticos y a su vez contiene la interfaz gráfica con la cual existe la interacción con el usuario.

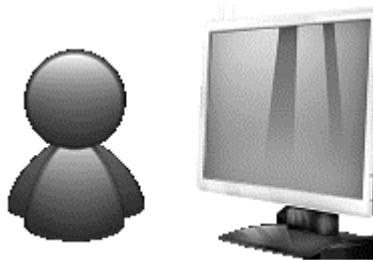


Figura 2.5.1 Arquitectura cliente-servidor de 1 capa.

Ventajas y desventajas de la arquitectura de una capa.

La ventaja de este esquema es que el usuario cuenta con todos los recursos que contiene el computador pero como desventaja es que si no administra bien los recursos estos se pueden degradar hasta que sea necesario aplicar un mantenimiento exhaustivo.

Arquitectura de dos capas.

En la arquitectura de dos capas una aplicación se organiza como servidor (o múltiples servidores idénticos) y un conjunto de clientes como se muestra en la figura 2.5.2. Dentro de esta arquitectura se presentan dos tipos o modelos:

- Modelo cliente ligero (thin-client). Es un modelo de cliente en el cual todo el procesamiento de las aplicaciones y la gestión de los datos se lleva a cabo en el servidor. El cliente simplemente es responsable de la capa de presentación del software.
- Modelo cliente pesado. En este modelo el servidor solo realiza la gestión de los datos por lo que el procesamiento de los mismos los realiza el cliente; es decir, el cliente a través del servidor va por los datos y al recibirlos los procesa para entregar al usuario.

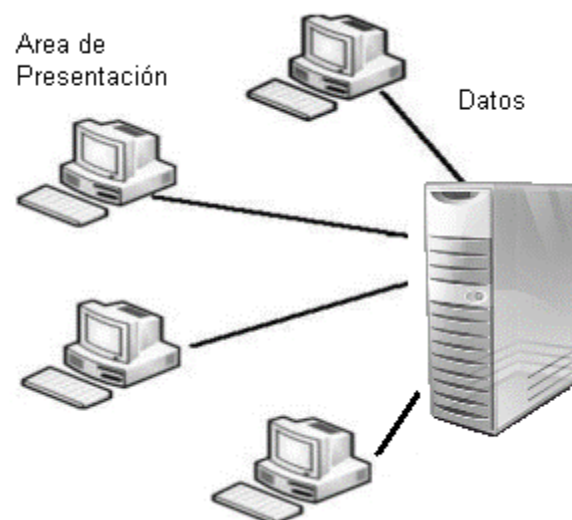


Figura 2.5.2 Arquitectura cliente-servidor de 2 capas

Ventajas en la arquitectura de dos capas.

- En el caso del modelo de cliente ligero no existe por lo regular preocupación alguna en mantener un cliente con gran cantidad de recursos ya que todo el trabajo prácticamente lo realiza el servidor.
- Muy adecuada para aplicaciones que requieren manejar una gran cantidad de datos (navegar y consultar) con poco o ningún procesamiento de la aplicación en el caso de clientes ligeros.

Desventajas en la arquitectura de dos capas.

- En el caso de falla de un servidor pesado todos los clientes se verán afectados.
- En caso de algún cambio en la aplicación; esta tiene que actualizarse en todos los clientes.

Arquitectura de tres capas.

Para evitar los problemas y desventajas de las arquitecturas de una y dos capas, se puede considerar la arquitectura de tres capas en la cual la presentación, el procesamiento de la aplicación y la gestión de los datos son procesos lógicamente separados que se ejecutan sobre procesadores diferentes.

Este tipo de arquitectura se puede encontrar por ejemplo en un sistema bancario en el cual la base de datos de clientes del banco (usualmente ubicada sobre un Mainframe) proporciona servicios de gestión de datos; un servidor web proporciona los servicios de aplicación tales como transferencia de efectivo, generación de estados de cuenta, pagos por servicios y así sucesivamente. La propia computadora del usuario con un navegador de Internet es el cliente.

Ventajas de la arquitectura de tres capas.

- Permite optimizar la transferencia de información entre el servidor Web y el servidor de la base de datos.
- Las comunicaciones entre estos sistemas pueden usar protocolos de comunicación de bajo nivel muy rápidos.
- Para la recuperación de la información de la base de datos se utiliza un middleware eficiente que soporte consultas a la base de datos en SQL.

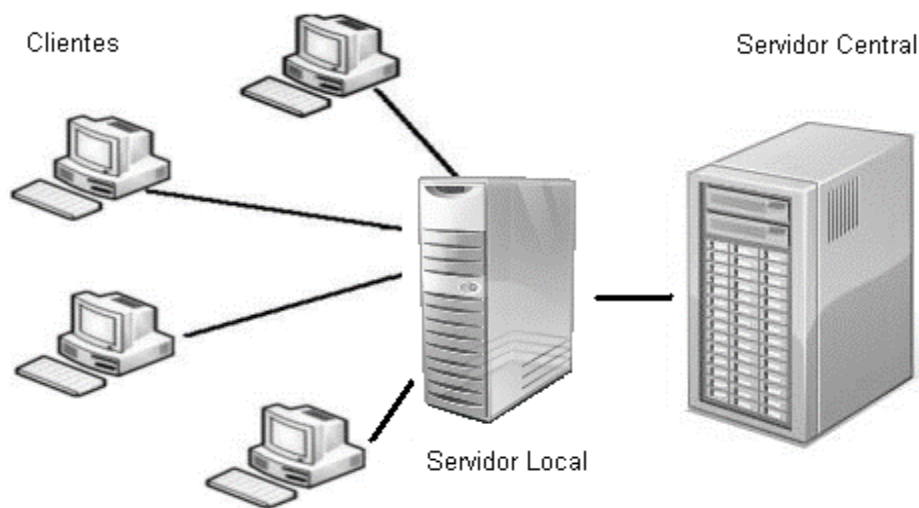


Figura 2.5.3 Arquitectura cliente-servidor de 3 capas.

Desventaja de la arquitectura de tres capas.

- Congestión del tráfico es el problema más común en una arquitectura cliente servidor de tres capas.

Arquitectura multicapa.

En algunos casos resulta adecuado extender el modelo cliente servidor de tres capas a una variante multicapa en la que se añaden al sistema servidores adicionales. Los

sistemas multicapa pueden usarse cuando las aplicaciones necesitan acceder y usar datos de diferentes bases de datos. El servidor de integración recoge los datos distribuidos y los presenta a la aplicación como si se obtuviesen desde una única base de datos.

Ventajas de la arquitectura multicapa.

- La arquitectura de tres capas y las variantes multicapa que distribuyen el procesamiento de la aplicación entre varios servidores son intrínsecamente más escalables que las de dos capas.
- El tráfico de la red se reduce al contrario que en la arquitectura de dos capas de cliente ligero.
- El procesamiento de la aplicación es la parte más volátil del sistema, y puede ser fácilmente actualizado debido a que está localizada centralmente.
- El procesamiento en algunos casos, puede distribuirse entre la lógica de la aplicación y los servidores de gestión de datos, en cuyo caso permite una respuesta más rápida a las peticiones de los clientes.
- Este tipo de arquitectura permite aplicaciones a gran escala con cientos o miles de clientes trabajando en paralelo.
- Integración de datos de múltiples fuentes.

Desventajas de la arquitectura multicapa.

- La detección de errores requiere de mucha especialización y por consiguiente puede ser muy costosa afectando al usuario o cliente final de la aplicación.
- Al permitirse una combinación de diferentes plataformas, el costo de licenciamiento, mantenimiento y soporte pueden influir con significancia en el presupuesto de una organización.

CAPÍTULO III

PLANTEAMIENTO DEL PROBLEMA.

3.1 Análisis del problema.

El área encargada de la administración en la organización debe proveer servicios administrativos y tecnológicos que coadyuven a soportar y optimizar sus funciones internas. Teniendo entre sus principales atribuciones, la gestión de recursos humanos, adquisiciones, finanzas, pagos a proveedores y el desarrollo tecnológico.

Como resultado de la operación diaria que tienen los usuarios responsables de estas actividades, continuamente se generan y actualizan numerosos expedientes documentales hacia el interior de las diversas áreas que conforman el área de administración.

Actualmente los usuarios de ésta deben crear expedientes físicos (en papel) de cada gestión o trámite que realizan para resguardar la documentación recibida y dar seguimiento a las actividades solicitadas por las diferentes áreas de la organización.

Se han identificado los siguientes procesos en general:

- Generación de expedientes.

De acuerdo a las funciones de cada área que realizan labores administrativas se generan expedientes una vez que se concreta un trámite o proceso, esquemáticamente. Vea la figura 3.1.1

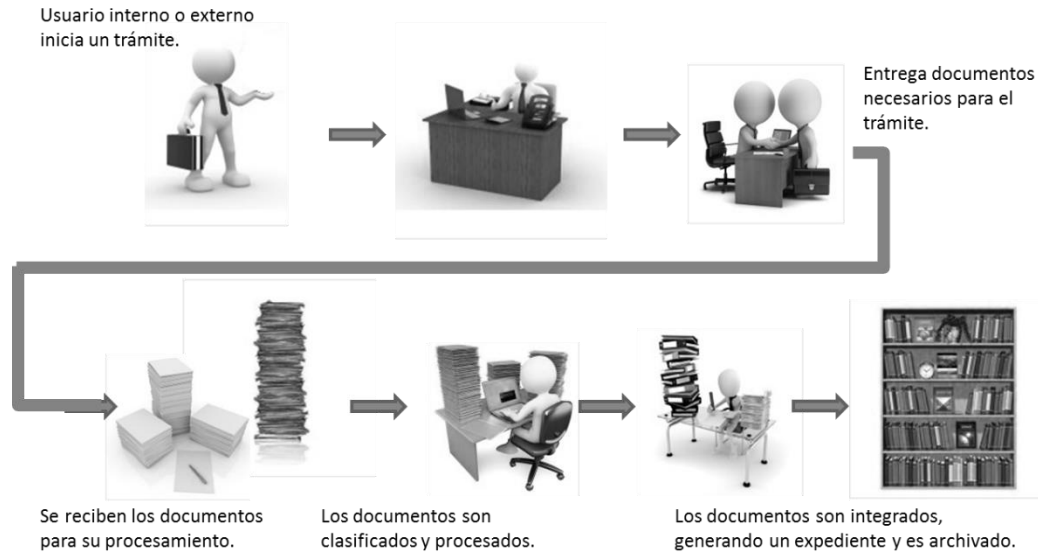


Figura 3.1.1 Generación de expedientes

Problemática:

- Los documentos son recibidos y procesados, se integran en una carpeta para generar un expediente. Existe la posibilidad de pérdida, extravío o de traspapelar un documento.
- Existe la posibilidad de que no se encuentre completamente integrado el expediente.
- Se desconoce el grado de integración de un expediente.
- El expediente se puede consultar hasta que es liberado por el área que lo integra.
- Existe atraso en la liberación de expedientes.

- Actualización de expedientes. Vea la figura 3.1.2:



Figura 3.1.2 Actualización de expedientes

Problemática:

- Durante la integración de los documentos recibidos al expediente respectivo, existe la posibilidad de pérdida, extravío o traspapelar un documento. Ya que los expedientes son integrados por diferentes personas en sus lugares de trabajo.
- Es posible que se integre erróneamente un documento a un expediente equivocado.
- El expediente no está disponible para su consulta, hasta que se termina de actualizar con los nuevos documentos.
- Las áreas retienen un expediente por periodos prolongados, lo que no permite su consulta por otros usuarios.

- Consulta de expedientes. Vea la figura 3.1.3:

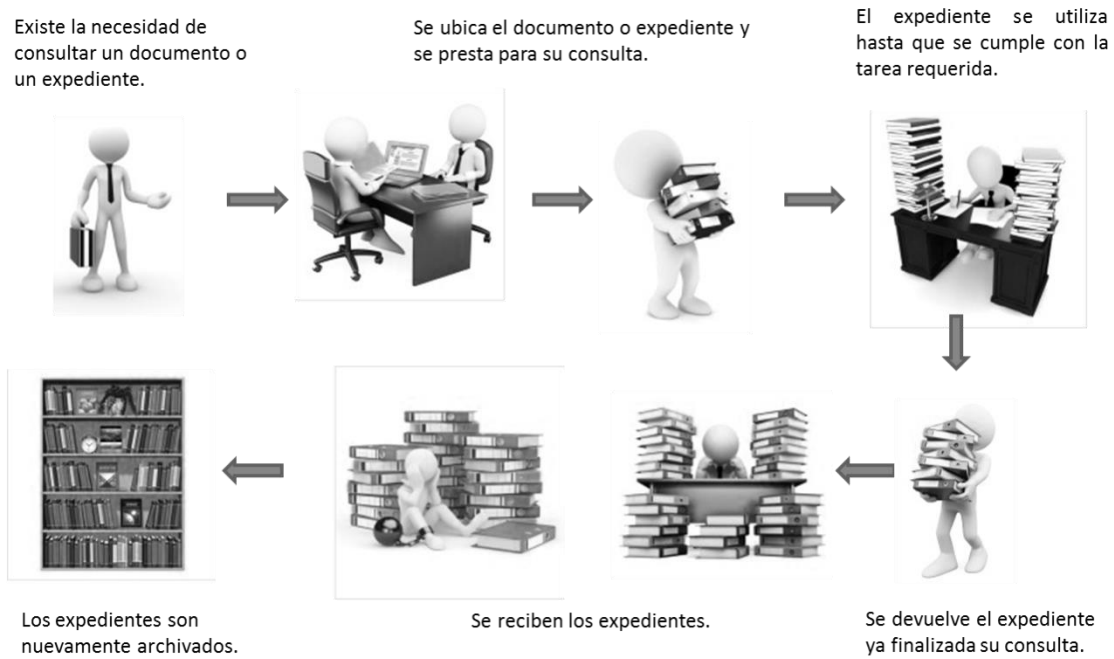


Figura 3.1.3 Consulta de expedientes

Problemática:

- Los expedientes son prestados a las áreas que solicitan su consulta. Por lo que existe la posibilidad de sustracción, alteración, pérdida, extravío o traspapelar un documento.
- Las áreas retienen un expediente por periodos prolongados de consulta, lo que no permite su uso por parte de otros usuarios. El expediente no está disponible para su consulta, hasta que es devuelto.
- Es posible que se deteriore el expediente o algún documento, derivado de la manipulación constante.
- Los expedientes pueden extraviarse.
- Los expedientes están expuestos a una contingencia durante su préstamo.
- Tiene que archivarse nuevamente.

Conclusión

El problema real de la gestión de expedientes físicos se reduce a:

- No se atiende oportunamente la gestión de tareas y actividades que realiza el área administrativa hacia las diferentes áreas internas de la organización.
- Necesidad de contar con un área cada vez mayor donde se resguarden los archivos en papel de todos los expedientes.
- Deterioro de los expedientes por el manejo durante su consulta y por causas como la humedad y tiempo de vida útil del papel.
- Posibilidad de extravíos de documentos físicos, derivado de no contar con un repositorio único de expedientes digitalizados.
- Búsquedas lentas y poco eficientes de documentos entre una gran cantidad de expedientes.
- No se cuenta con capacidad para generar y compartir expedientes entre diferentes departamentos y usuarios de la institución.
- El expediente en préstamo no está disponible para su consulta, hasta que es devuelto.
- No se posee un respaldo de seguridad para evitar pérdida documental.
- Consumo de grandes cantidades de papel, debido a que generalmente hay que sacar copias adicionales de expedientes.
- Los expedientes pueden extraviarse.

3.2 Requerimientos generales y particulares de la aplicación.

“Un requerimiento es una necesidad documentada que identifica atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario.”

Por definición los requerimientos en aplicaciones de software, son las características que debe tener el software instalado en una computadora para poder soportar y/o ejecutar una aplicación específica. Es una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema. Es decir, los requerimientos son lo que los clientes/usuarios esperan que haga el sistema.

Requerimientos generales.

- El sistema estará enfocado a personas con poca experiencia en computación y deberá contar con una interfaz amigable y con un diseño moderno donde la información esté muy bien organizada, lo cual motive al usuario a usar la aplicación.
- Las pantallas deberán tener un estilo en común, es decir, la posición de las opciones y los menús deben ser similares, así como el fondo y los colores.
- Las opciones deberán estar organizadas, evitando una saturación de elementos en la pantalla.
- Los textos que contengan información importante deberán estar en negritas y con mayúsculas, con un tamaño un poco más grande que las demás.
- Las imágenes o logos que se utilizarán deberán adecuarse a las pantallas de los usuarios, evitando que se distorsionen y no se puedan ver claramente.
- El sistema será desarrollado utilizando lenguajes de programación orientados a web, específicamente:
 - HTML: con el cual manejaremos la estructura de la página.

- PHP: que es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- Javascript: y jquery el cual puede afectar el comportamiento de navegadores web.
- El Sistema deberá operar en equipos que cuenten como mínimo con las siguientes especificaciones:
 - Servidor: 1.0 Ghz, RAM 2 GB, 400 MB de espacio en disco para la instalación.
 - Cliente: Pentium 500 MHz (Recomendado: Pentium 1GHz o superior), 1 GB RAM (Recomendado: 2 GB RAM o más), 300 MB de espacio en disco y monitor con resolución de 1024 x 720.
- Todos los datos serán almacenados en una BD utilizando el manejador SQL SERVER, ya que es rápido, confiable, seguro y se acopla al sistema que se desarrollará, además el usuario posee las licencias.
- Se deberá contar con una utilería de respaldos de la BD a utilizar, sobre todo donde se guardan la información de los documentos. También un plan de mantenimiento de la base de datos semestralmente, haciendo backup de las tablas necesarias y separando las que se utilizaran en el archivo histórico.
- El servidor deberá estar instalado en las computadoras con que cuente la institución, y éste dentro del mismo edificio o centro de cómputo, si es que cuentan con uno.
- El administrador del sistema será el único en poder acceder a toda la información almacenada en la base de datos, así como la actualización de los reportes correspondientes a entregar.
- Que no requiera de capacitación para el usuario
- Que se ejecute en un ambiente cliente – servidor.
- Que posea perfil de administrador, operador y usuario.

Requerimientos de seguridad.

- El sistema deberá tener un control de login, ya que es el control de inicio de sesión más simple y que soporte las situaciones más comunes; identificarse utilizando un nombre de usuario y una contraseña. El control incluirá los cuadros de texto usuario y contraseña y una casilla de aceptación.

Función de autenticación de usuario.

- Introducción: mediante la digitación del usuario o id y contraseña, los usuarios registrados y el administrador accederán a la aplicación y a sus características asociadas.
- Entradas: para el campo de usuario o id y contraseña se requieren caracteres alfanuméricos.
- Proceso: se realizará una comprobación de que el usuario y su contraseña coincidan dentro de la base de datos, si su identificación es correcta se mostrará la pantalla de inicio con las opciones del perfil de usuario habilitadas. En caso contrario aparecerá una pantalla con un aviso de que sus datos no son correctos.
- Salida: en caso de que la identificación sea correcta nos llevará a una pantalla con los menús necesarias para iniciar la captura y visualización de los documentos a archivar.

Funciones de visualización de la plantilla a capturar:

- Introducción: se selecciona la plantilla a usar.
- Proceso: el sistema cargará la información por partes y el usuario deberá usar los catálogos donde estén disponibles, sino no lo hace no podrá avanzar a la siguiente parte de la plantilla.

- Salida: en la pantalla aparecerá que su registro fue guardado satisfactoriamente y regresará al usuario para que continúe capturando la siguiente información. Al concluir con todos se le generará un número de folio, con lo cual se verifica que se terminó por completo la captura.

Requisitos particulares.

- Satisfacer las necesidades operativas de la dirección de archivo y documentación en el plan y programas de recuperación de información.
- Proveer apoyo a los ejecutivos para la adecuada asignación de los planes y programas, de acuerdo a las restricciones y políticas definidas por el negocio.
- Proporcionar información oportuna, íntegra y confiable de las gestiones realizadas por el personal administrativo.
- Brindar un servicio óptimo, con la finalidad de incrementar los beneficios y reducir los tiempos de respuesta, de acuerdo a los estándares operativos de la dirección.
- Realizar un proceso de carga inicial con la información ya digitalizada.
- Establecer la carga de histórico de los registros de los archivos de información confidencial.
- Que posea una búsqueda por filtro en todas las opciones.
- Que posea una bitácora de seguridad para evitar mal uso del sistema.
- Que se apegue a los estándares de la institución en cuanto a sistemas de cómputo.
- Que solo el personal autorizado pueda darle mantenimiento.

3.3 Análisis y posicionamiento de los lenguajes para el back-end y front-end.

Es fundamental llevar a cabo un análisis de las herramientas existentes en el mercado para poder elegir la mejor tecnología que permita cumplir las expectativas del usuario final y se ajuste a los requerimientos del sistema que se desea desarrollar. Algunos factores que se deben considerar son el tamaño y tipo de la aplicación, el número de usuarios concurrentes, el equipo y software disponible y el tiempo de entrega del sistema. En base a estos factores se analizan algunas opciones de solución y posteriormente se selecciona la más viable.

3.3.1 Elección del back-end (base de datos).

Para la elección de este componente se analizaron tres bases de datos: Son de carácter propietario.

SQLServer.

Sistema de base de datos relacional. Su uso es de carácter propietario.

Características principales.

- Solo permite conexiones desde el equipo local vía sockets de dominio.
- Permite cargar bloques de código para que se ejecuten en el servidor (Funciones). Estas son escritos en diferentes lenguajes, tales como Java, C, C++ y PL/pgSQL.
- Permite la alta concurrencia haciendo uso de 2 modos de control de concurrencia: “Pessimistic Concurrency” (uso de bloqueos exclusivos y compartidos) y “Optimistic Concurrency” (Permite la creación de una nueva fila cada vez que esta se actualiza, almacenando las versiones anteriores a la fila

de una tabla temporal junto con el ID de la transacción mediante la cual fueron actualizados.

- Permite migración de las bases de datos en la nube mediante la plataforma “Cloud-Ready”
- Cuenta con 3 formas diferentes de respaldo y recuperación de bases de datos: SQL Server Management Studio (herramienta grafica), Power Shell (Interfaz de consola avanzada) y Transact-SQL (lenguaje propio de SQL Server).
- Amplia variedad de tipos de datos permitiendo de manera adicional que los usuarios puedan crear sus propios tipos de datos.
- Uso de llaves foráneas, triggers, vistas, integridad transaccional entre otras.
- Incorpora un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Integra la tecnología “AlwaysOn” es una solución de alta disponibilidad y recuperación ante desastres integrada, flexible y eficiente en cuanto a costes de SQL Server 2012, proporciona redundancia dentro de centros de datos y entre ellos, y permite una rápida conmutación por error de las aplicaciones, ofreciendo la máxima disponibilidad y protección de datos a las aplicaciones críticas.

Oracle.

Es un sistema de gestión de base de datos objeto-relacional (ORDBMS), multihilo y multiusuario. Es el mas usado a nivel mundial y considerado el sistema de bases de datos más completo por tener soporte de transacciones, estabilidad, escalabilidad y ser multiplataforma.

Características principales.

- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.

- Cuenta con la herramienta Oracle Designer que soporta las técnicas de modelado como los diagramas entidad-relación, ingeniería de información y diseño de objetos.
- Soporte a multiplataforma.
- Procedimientos almacenados, triggers y cursores.
- Oracle repository permite la configuración de los objetos de bases de datos, formularios, clases Java, archivos XML, y otro tipo de archivos.
- Soporta todas las funciones que se esperan de un servidor, un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Posee asistentes que ayudan a los usuarios finales a visualizar los datos como gráficos.
- En seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos.
- Cuenta con la característica "Oracle Data Guard" que permite tener una alta disponibilidad, realizando una copia de la base de datos en sistemas por separados.

Sybase.

La empresa Sybase Inc fue fundada en Berkeley California en 1984 y fue hasta mayo del 2010 que SAP la adquirió por la cantidad de 4567 millones de euros.

Uno de sus productos es el Adaptive Server Enterprise (ASE) que es un motor de bases de datos. Fue la primera base de datos con arquitectura cliente – servidor, entre sus principales características son que es altamente escalable, tiene un alto rendimiento, procesa grandes volúmenes de información, transacciones y usuarios y a un costo bajo. Dicho manejador permite almacenar datos de manera segura, acceso y procesamiento de datos de manera inteligente y movilización de datos.

ASE cuenta con 5 versiones diferentes que cuentan con distintas características se adaptan a diversos negocios permitiendo tener un software que se adapte más a las necesidades del cliente. Su licenciamiento depende de la versión que se elija que puede ser desde gratuito hasta pagar una cuota anual. Las 5 versiones son:

- ASE Enterprise Edition.
- ASE Small Business Edition.
- ASE Developer's Edition.
- ASE Express Edition.
- ASE Cluster Edition.

Características principales.

- Técnicas de particionamiento semántico de tablas que aumentan la velocidad de acceso a los datos.
- Columnas cifradas para mayor seguridad de los datos.
- Mejoras al lenguaje Transact-SQL, para mayor productividad.
- Columnas computadas "virtuales" y materializadas, e índices funcionales, que brindan mayor rendimiento.
- Mejoras a los servicios de Java y XML en la base de datos.
- Herramientas mejoradas para la administración y el monitoreo.
- Copias de respaldo en línea y de alto rendimiento.
- Programador de tareas.
- Soporte a múltiples herramientas de desarrollo y lenguajes de programación, como PowerBuilder, Visual Basic, Java, C, PHP, etc.
- Soporte a múltiples protocolos de conectividad, como Open Client (propio de Sybase), ODBC, OLE DB, ADO.NET y JDBC.
- Multiplataforma, incluyendo IBM AIX, HP-UX, Silicon Graphics IRIX.

La tabla 3.3.1.1 muestra la comparación de algunas características de las tres bases de datos anteriormente mencionadas.

| Características | SQL Server 2012 | Oracle 12.1 | Sybase – ASE 15 Enterprise Edition |
|---|---|---|---|
| Licenciamiento | Privativa | Privativa | Gratuita y privativa. |
| Plataformas | Windows | Larga lista de plataformas | Larga lista de plataformas |
| Soporte de 64 bits | Si | Si | Si |
| Limite por base de datos | 524 PB | 4PB | Ilimitada |
| Limite por tabla | Limitado por el número de objetos de la base de datos | 16 TB en Solaris | Ilimitada |
| Número de índices por tabla | Limitados | Depende de las tablas existentes y espacio en el disco duro | Ilimitada |
| Limites numéricos | Enteros 64 bits | Enteros 64 bits | Enteros 64 bits |
| Límites de cadenas (VARCHAR) | 2048 MB | 32767 bytes | 32KB |
| Soporte de unicode | Si (UTF-16) | Si (UTF-16) | Si (UTF-16) |
| Transacciones | Si, conformidad ACID | Si, conformidad ACID | Si, conformidad ACID |
| Candados a nivel renglón de tabla (conurrencia) | Modelo MVCC | Modelo MVCC | Modelo MVCC |
| Funcionalidad de TIMEOUT | No soportada | Soportada | Soportada |
| Soporte de stores | Soportada | Soportada | Soportada |

| Características | SQL Server 2012 | Oracle 12.1 | Sybase – ASE 15 Enterprise Edition |
|--------------------------------------|--|--------------------|---|
| procedures | | | |
| Soporte de triggers | Soportada | Soportada | Soportada |
| Escalabilidad/soporte SMP | Soportada | Soportada | Soportada |
| Balanceo de cargas | No soportado | Soporte | Soporte |
| Replicación | Soportado | Soportado | Soportado |
| Respaldo en línea (Online Backup) | Soportada con la tecnología “AlwaysOn” | Soportado | Soportado |

Tabla 3.3.1.1 Software de base de datos.

3.3.2 Elección del front-end.

Para el desarrollo y construcción del front-end que es la parte que sirve de interfaz al usuario y operador, esta interfaz tiene que ser amigable y en un ambiente totalmente gráfico para la representación de resultados, debe ser operablemente sencilla capaz de realizar las acciones con la base de datos de una manera eficiente, para ello se han tomado los siguientes productos, PHP, JavaScript y jQuery. Ver tabla 3.3.2.1.

PHP.

Es un lenguaje de programación de uso general de script, ejecutado en el servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha

evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP es independiente de la plataforma y puede ser usado sin ningún costo.

Características de PHP.

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.
- La principal ventaja se basa en ser un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo los que ingresan los usuarios desde formularios con código HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una muy buena documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de programación orientada a objetos (clases y herencia.).
- Muy sencillo de aprender.
- La sintaxis es similar a lenguajes como C, Java y PERL.
- Manejo de excepciones.
- Biblioteca nativa de funciones.
- Fácil integración con varias bibliotecas externas, incluso puede generar documentos PDF y analizar código XML.

JavaScript.

JavaScript es un lenguaje de programación orientado a objetos, basado en prototipos, imperativo y dinámico.

Características de JavaScript.

- Creación de entidades más genéricas(objetos) que permitan la reutilización del código.
- Orientado a objetos, se refiere a un método de programación y al diseño del lenguaje. Diseñar el software de forma que haya independencia entre los módulos desarrollados.
- La independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de dispositivo.

- La licencia de Java de Sun insiste que todas las implementaciones sean compatibles, sin importar el fabricante.
- Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad.
- La portabilidad, es de hecho posible ya que al escribir programas para la plataforma Java actuarían de forma correcta en múltiples plataformas de distinta arquitectura.
- En Java el problema de las fugas de memoria se evita en gran medida gracias a la recolección de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos, el recolector de basura de Java permite una fácil creación y eliminación de objetos y mayor seguridad.
- La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos.
- Es seguro y fiable. Por lo que puede ser filtrado para el mismo Javascript. La seguridad es casi total.
- El código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido.
- Reducción de código.
- Código más conciso y de fácil lectura.

jQuery.

jQuery es un Framework JavaScript de carácter gratuito, combina su facilidad de uso y su gran potencia, en otras palabras es un conjunto de bibliotecas de código que contienen procesos o rutinas que garantizan un correcto funcionamiento reduciendo muchas líneas de código.

Características de jQuery.

- Es de carácter gratuito.
- Permite al desarrollador concentrarse más en el diseño del sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores disponibles en el mercado.
- Permite manipular series de elementos y modificarlas con una simple línea de código.
- jQuery es muy fácil de expandir, ya que cuenta con gran cantidad de plug-ins que se pueden utilizar o hasta crear uno propio.
- Compatible con todos los navegadores modernos.
- Al ser un código abierto el desarrollador puede modificar el código fuente y ajustarlo a las necesidades del negocio.
- Permite desarrollar animaciones.
- Tiene una excelente comunidad de soporte.

XHTML.

XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades pero cumple las especificaciones más estrictas de XML.

Características de XHTML.

- Permite una correcta interpretación de la información, independientemente del dispositivo con que se accede a ella.
- Permite el manejo de la información en la WWW, de una manera estandarizada.
- Es posible usar las etiquetas tradicionales de HTML pero con la sintaxis moderna de XML.
- Esta diseñado para mantener una independencia entre contenido y el diseño.

La tabla 3.3.2.1 muestra la comparación de los lenguajes anteriormente mencionados.

| CARACTERÍSTICAS | PHP | JavaScript y jQuery | XHTML |
|-----------------------------------|----------------------------------|---|---|
| Marca | Libre | Sun | Libre |
| Programación | Orientada a objetos. | Orientada a objetos | Estructurada |
| Conectividad con base de datos. | Todas | Todas | No |
| Herramienta de generación de SQL. | Sí | Sí | No |
| Plataforma | Cualquiera | Cualquiera | Cualquiera |
| Gestión de excepciones. | Sí | Sí | No |
| Ventajas | Multiplataforma, software libre. | Multiplataforma y portabilidad. Amplía sus funcionalidades integrando el Framework jQuery. | Multiplataforma, manejo de información en la Web Estandarizada. Independencia entre contenido-diseño. |

Tabla 3.3.2.1 Software front-end

La elección de las herramientas para el desarrollo del sistema se realizó tomando en cuenta las siguientes consideraciones:

- Los conocimientos con los que cuentan los integrantes del equipo desarrollador.

- La portabilidad del software.
- El Software con el que cuenta el usuario

Las herramientas elegidas para el desarrollo del sistema son las siguientes que veremos en la tabla 3.3.2.2:

| Componente | Producto | Característica Principal |
|---|----------------------------|--|
| Manejador de base de datos | SQLServer | Alta concurrencia. Instalación ilimitada. Soporte. Ahorros considerables en costos de operación. Estabilidad y confiabilidad. Extensible Diseñado para ambientes de alto volumen. Herramientas gráficas de diseño y administración de BD. |
| Lenguaje de programación para páginas dinámicas, acceso y actualización de base de datos, interfaz. | JavaScript, jQuery y XHTML | Código escrito en Java y HTML. Independencia de plataforma. Portabilidad. Facilidad de instalación configuración y utilización. jQuery cuenta con una gama de objetos ya programados que optimizan el desarrollo de las aplicaciones. |

Tabla 3.3.2.2 Software front-end.

3.4 Separación de los posibles módulos según la metodología de Yourdon/De Marco.

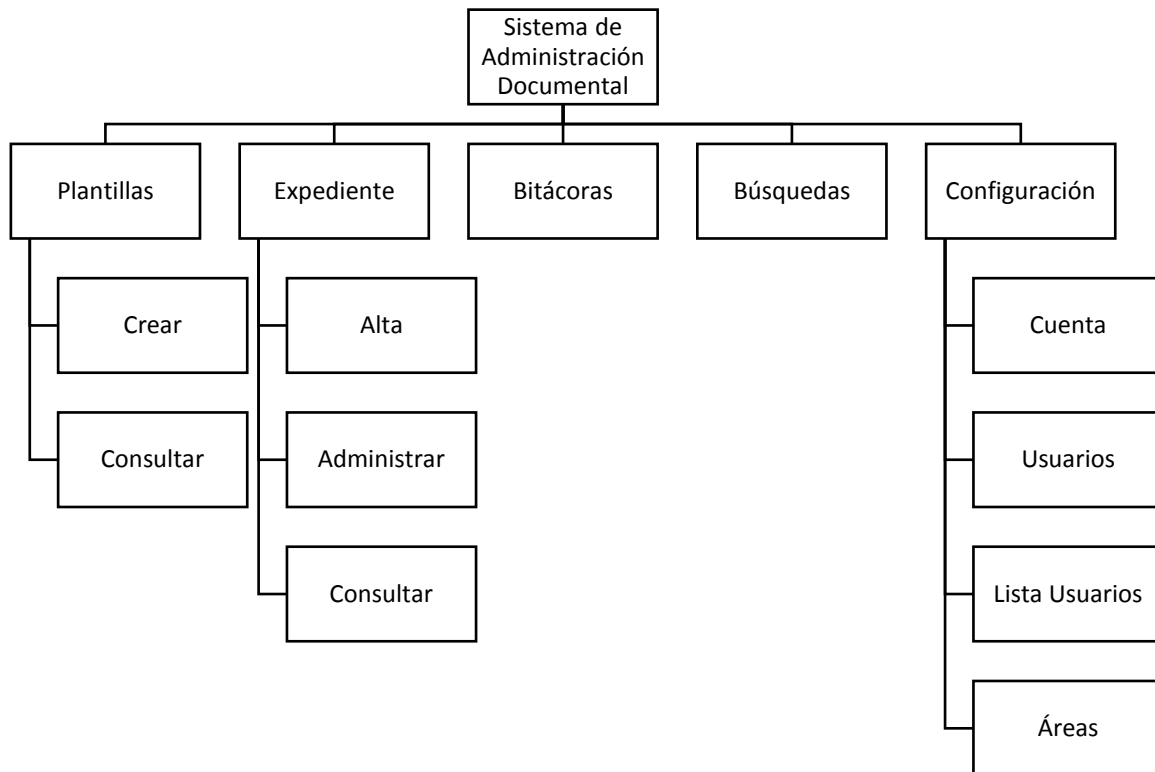


Figura 3.4.1 Diagrama de bloques del sistema.

En la figura 3.4.1 podemos observar el diagrama de bloques general de nuestro sistema.

Separación de los módulos para el Sistema de Administración Documental.

En esta sección se tratará de explicar de manera general lo que debe hacer cada módulo ya que la explicación a detalle será abordada en el Capítulo IV.

Autenticación.

En este módulo el personal de la empresa deberá de introducir un usuario con su contraseña el cual sea válido dentro del sistema para poder acceder a él, la empresa se encargará de definir su rol.

Por lo anterior el módulo deberá tener un formulario con los siguientes datos:

- Usuario.
- Contraseña.

Plantillas.

En esta parte los usuarios podrán crear el esqueleto general del expediente, indicando el número que carpetas necesarias para poder subir los archivos correspondientes de un expediente en particular.

Expediente.

En esta sección el usuario será capaz de administrar todo lo relacionado a un expediente, desde crearlo, modificarlo o simplemente buscarlo.

Bitácoras.

Módulo que se encargará de registrar todos los movimientos realizados por los usuarios desde que inician sesión hasta que salen del sistema.

Búsquedas.

En este módulo el usuario podrá realizar búsquedas para encontrar rápidamente documentos, apartados o expedientes que ya se encuentren alojados dentro del sistema.

Configuración.

En este módulo los usuarios podrán realizar cambios a su cuenta aunque también los usuarios que cuenten con el privilegio de administrador podrán realizar actividades relacionadas con el mantenimiento del sistema.

- Cuenta: en este apartado el usuario podrá cambiar su contraseña.
- Usuarios: dentro de este apartado se desplegará la lista de usuarios registrados en el sistema además de mostrarnos el tipo de usuario que es y a qué área está asignado; por ultimo también se puede agregar usuarios nuevos al sistema.
- Tipos de usuarios: en éste se podrá visualizar los diferentes tipos de usuarios disponibles en el sistema.
- Área: del mismo modo que el apartado de Tipos de usuarios éste se encarga de mostrarnos en pantalla las diferentes áreas registradas dentro del sistema.

3.5 Diseño a nivel de bloques de la interfaz del usuario.



Figura 3.5.1 Página de sign-on al sistema

Página de sign-on, ver figura 3.5.1

- Logo de la empresa – contiene los logos de la empresa, título de la misma y nombre del sistema.
- Dirección de la página – contiene de manera informativa (no es un link) la página en la cual se puede acceder esta aplicación.
- Sign-on al sistema – permite introducir el usuario y clave para acceder al sistema.

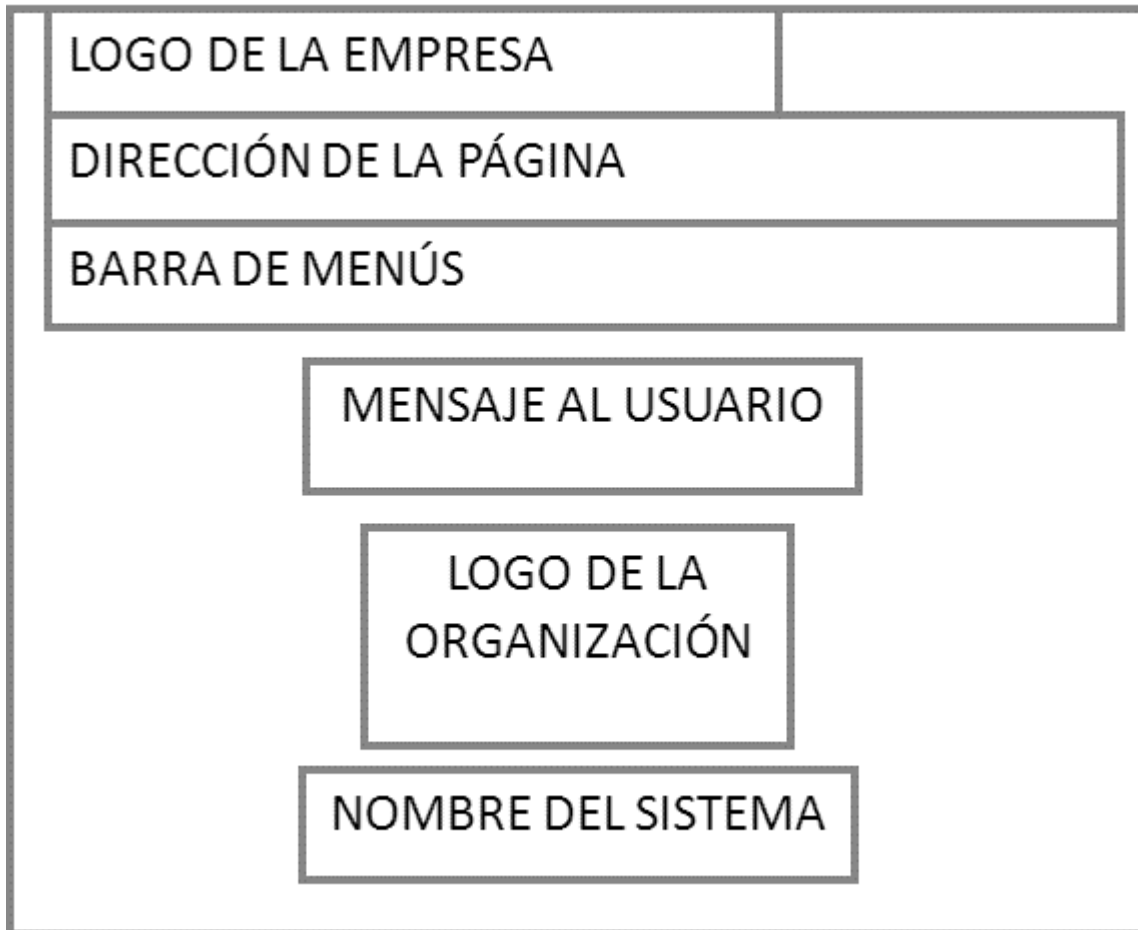


Figura 3.5.2 Pantalla principal del Sistema de Administración Documental

Pantalla principal del sistema, ver figura 3.5.2

- Logo de la empresa – contiene los logos de la empresa, título de la misma y nombre del sistema.
- Dirección de la página – contiene de manera informativa (no es un link) la página en la cual se puede acceder esta aplicación.
- Barra de menús – contiene todas las opciones principales del menú que son: Inicio, Plantillas, Expedientes, Bitácora, Búsqueda, Configuración y Salir. Algunas de estas ofrecen otras opciones a su vez.
- Mensaje al usuario – en el menú principal esta zona es para mensajes al usuario como ejemplo el de bienvenida.

- Logo de la organización– logotipo de la organización que mantiene el sistema.
- Nombre del sistema – nombre del sistema “Sistema de Administración Documental”.

En las pantallas subsecuentes del sistema los tres primeros bloques de logo, barra de dirección y de menú se mantienen.

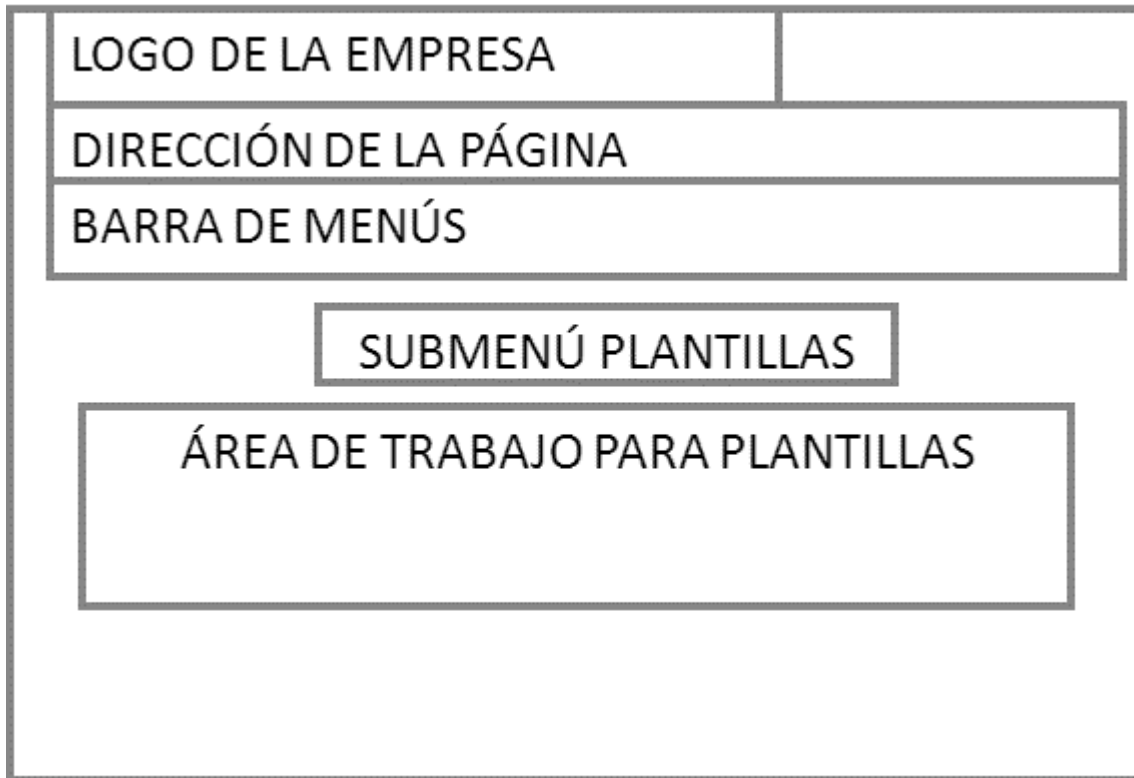


Figura 3.5.3 Pantalla para operaciones sobre plantillas.

Pantalla para creación y consulta de plantillas, ver figura 3.5.3

- Submenú plantillas – aquí se muestra la actividad a realizar en cuanto a creación de plantillas así como consulta de las mismas.
- Área de trabajo para plantilla – zona en la cual el usuario crea las plantillas las cuales se conforman de carpetas y subcarpetas que contendrán los documentos en los diferentes formatos. Para el caso de consulta al usuario se le permite navegar entre todas las carpetas para ver el contenido de cada una de ellas.

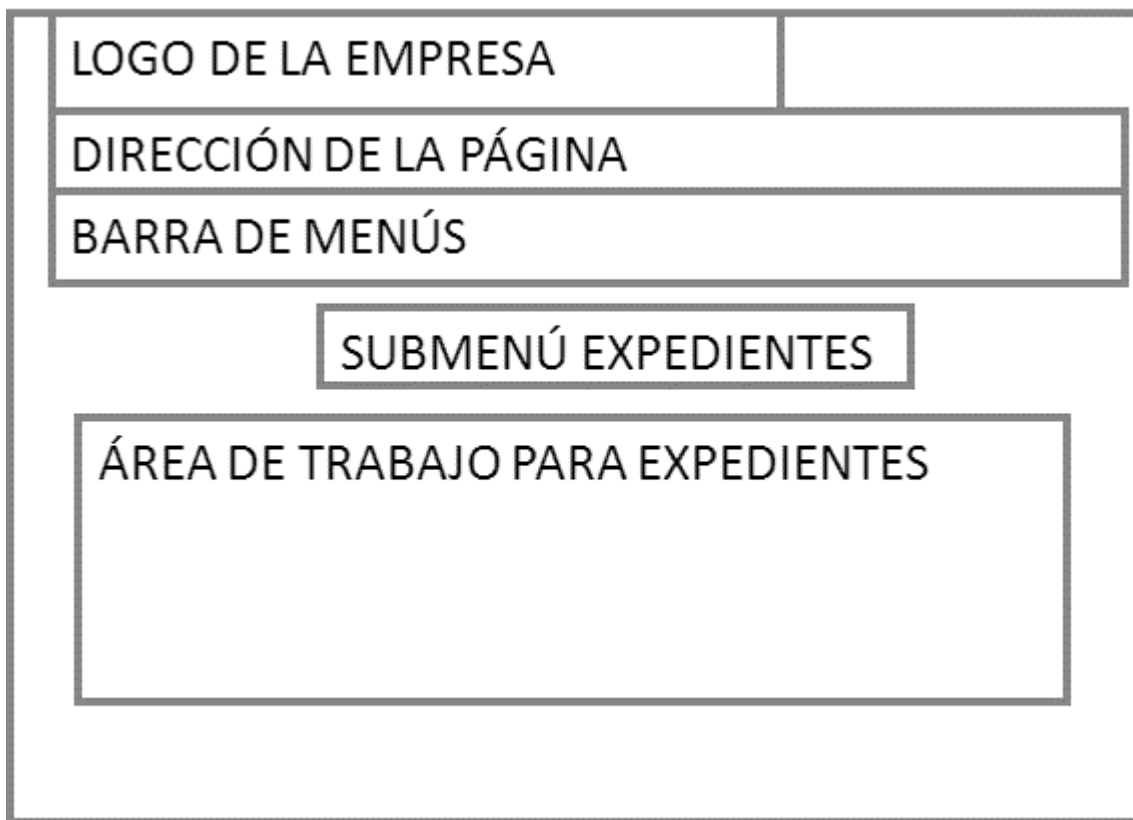


Figura 3.5.4 Pantalla para operaciones sobre expedientes.

Pantalla para alta, administración y consulta de expedientes, ver figura 3.5.4

- Submenú Expedientes – aquí se muestra la actividad a realizar en cuanto a las diferentes operaciones sobre los expedientes que en sí son los documentos que se alojan en las plantillas.
- Área de trabajo para Expedientes – zona en la cual se realizan las tres operaciones sobre expedientes: alta de expedientes, administrar expedientes y consultar expedientes. Para cada expediente se especifica su clave, área a la cual pertenece y el tipo de template a utilizar. En nombre de expedientes es acorde a la dependencia de la organización en cuanto a los estándares que cada una establece.

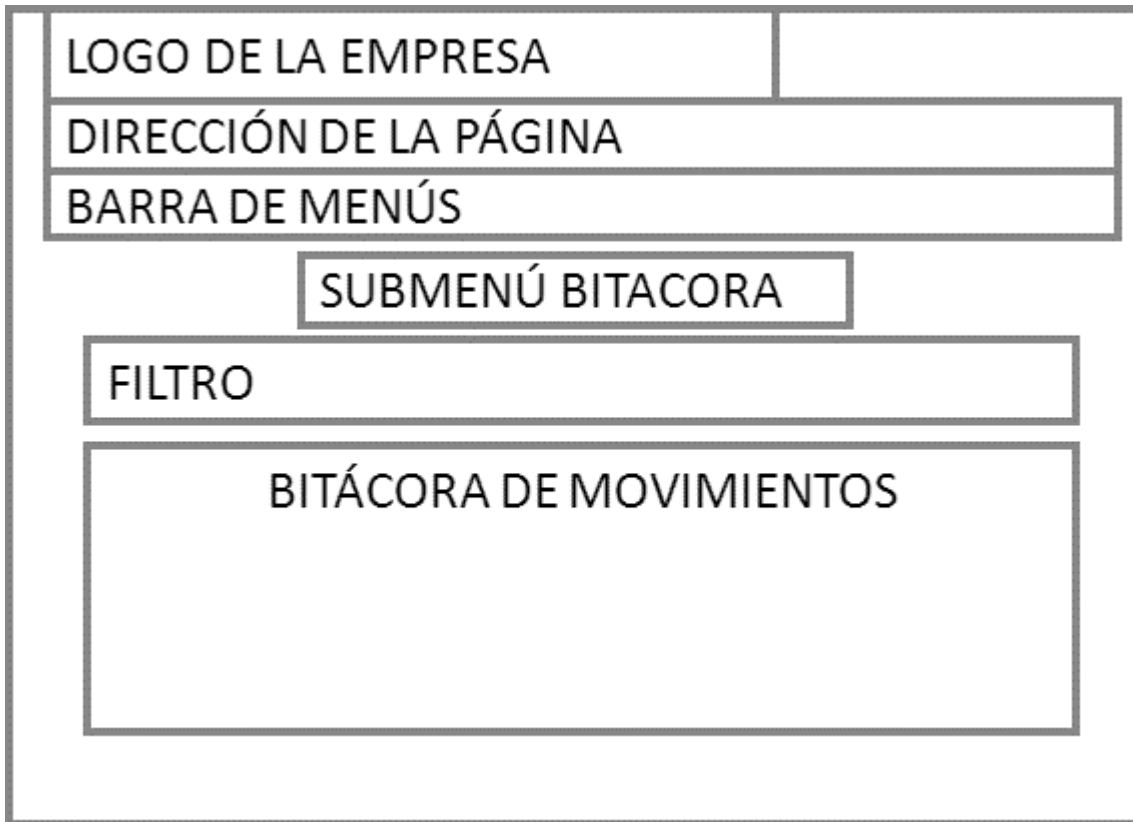


Figura 3.5.5 Pantalla relativa a la Bitácora del SAD

Pantalla para consulta de la bitácora del sistema, ver figura 3.5.5

- Filtro – en esta sección se puede indicar el filtro para consultar la bitácora permitiéndose especificar tipo de documento, fecha y hora.
- Bitácora de movimientos – en esta zona se despliegan todas las operaciones que los usuarios realizan en el SAD mostrando el día y la hora de cada una.

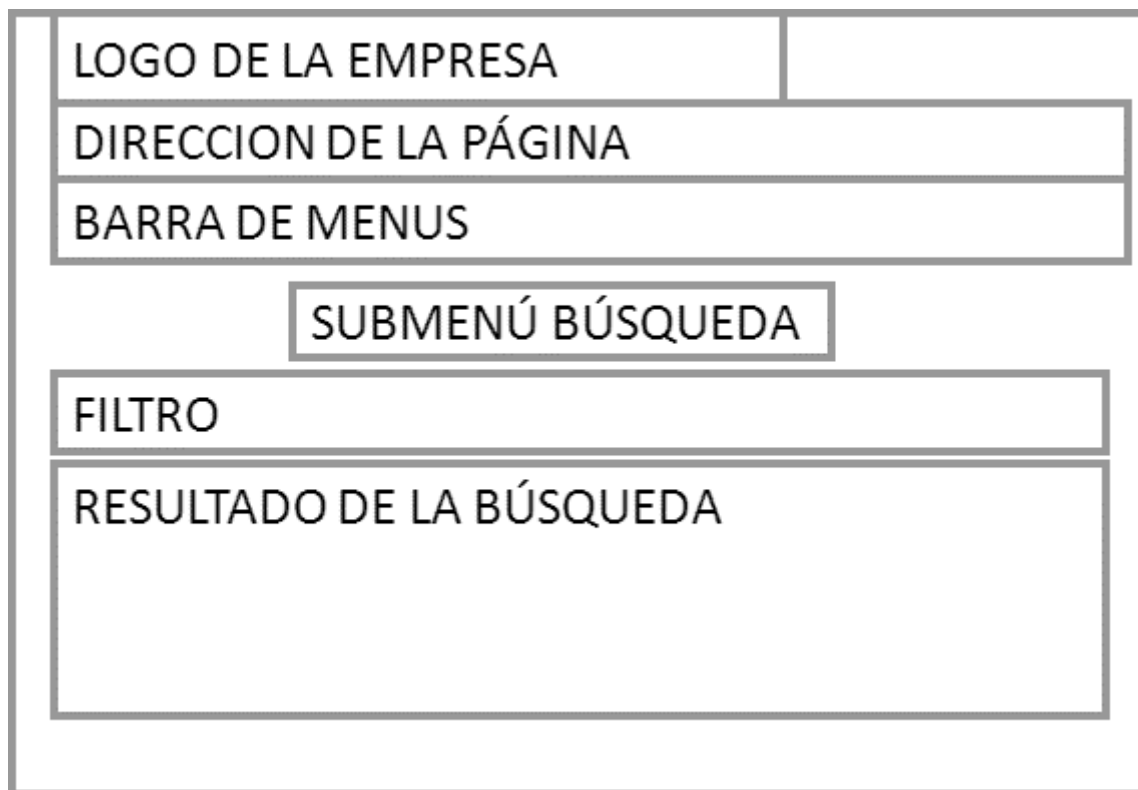


Figura 3.5.6 Pantalla de búsqueda

Pantalla para búsqueda de archivos, apartados y expedientes, ver figura

3.5.6

- Filtro – en esta sección se puede indicar el componente a buscar en el sistema.
- Resultado de la búsqueda – zona en la cual se muestra el resultado de la búsqueda.

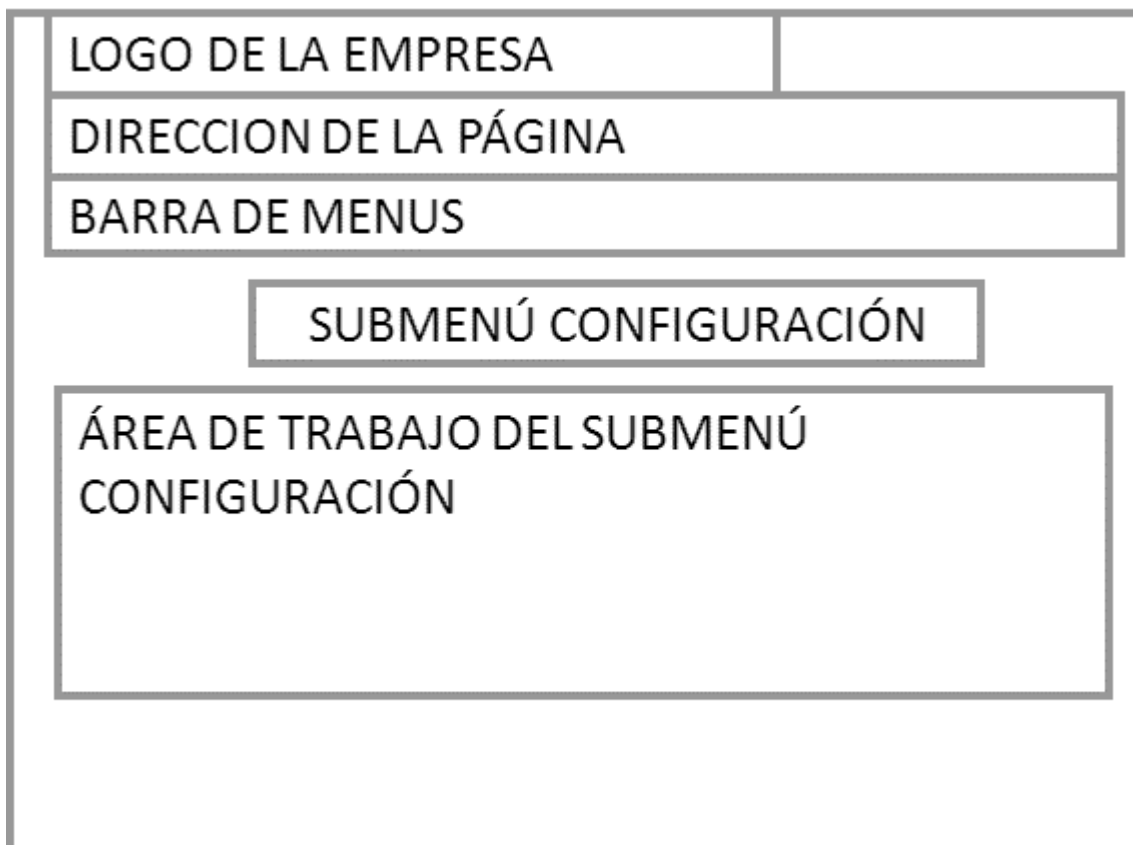


Figura 3.5.7 Pantalla de Configuración

Pantalla de configuración de cuentas, usuarios, tipos de usuarios y áreas, ver figura 3.5.7

- Submenú configuración – indica la modalidad en la cual se encuentra el usuario (Configuración de su cuenta, de usuarios, tipos de usuarios y áreas).
- Área de trabajo del submenú configuración – aquí se permite al usuario modificación de su contraseña, elegir el tipo de usuario en el sistema (Administrador, consulta, estándar), definir los tipos de usuario y las áreas dentro de la organización.

CAPÍTULO IV

ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN.

4.1 Introducción.

Una metodología aplicada en el desarrollo de software, es el conjunto de procedimientos, técnicas, herramientas que sirven para asegurar el éxito en la construcción de un sistema.

En el Capítulo II se describieron tres metodologías de las cuales se debe elegir una de ellas para el desarrollo de este trabajo, para poder tomar una decisión y ver cuál de ellas se apega más a las necesidades de nuestro proyecto, a continuación se presenta una breve síntesis de las metodologías de MERISE, SSADM y Yordoun/DeMarco.

Metodología Merisse.

Está orientada a procesos, se basa en cuatro fases ver figura 4.1.1, abarca desde el análisis y el diseño hasta la implementación y el mantenimiento, no distingue claramente las divisiones entre las etapas de desarrollo.

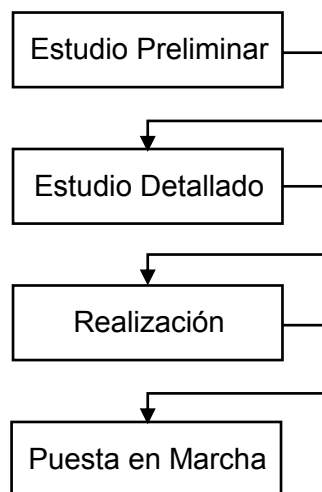


Figura 4.1.1 Metodología de Merisse.

Metodología SSADAM.

Está orientada al análisis y diseño, se basa en tres fases ver figura 4.1.2, se ocupa únicamente del análisis y diseño, no define la etapa de la construcción del proyecto ni de su implementación.

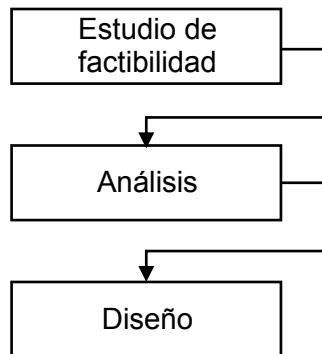


Figura 4.1.2 Metodología SSADM.

Metodología de Yourdon/De Marco.

Proporciona diversas técnicas especializadas en modelar aspectos específicos de un sistema, brinda un panorama amplio y permite la obtención de documentación concreta facilitando el control en el desarrollo del proyecto; esta metodología se basa en cinco fases; ver figura 4.1.3. Abarca desde el análisis y el diseño, hasta la implementación y mantenimiento.

Yourdon/De Marco propuso la metodología de análisis conocida como análisis estructurado, la cual permite modelar los proyectos, en base a la información que se intercambia en los procesos, a diferencia del enfoque tradicional que se centraba en los procesos propiamente dichos.

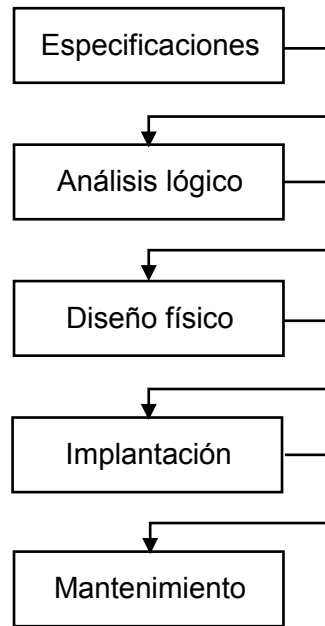


Figura 4.1.3 Metodología Yourdon/De Marco.

Conclusión.

De las tres metodologías descritas se eligió aplicar la técnica Yourdon/De Marco para atender a las necesidades de los usuarios finales de la organización y así reflejar el resultado de las especificaciones o requerimientos que aseguren el éxito del sistema, derivado de que dicha metodología está basada en funciones o procesos.

Así mismo determina los elementos internos y externos que intervienen en un sistema. Es decir, los elementos que el sistema recibe del exterior y la información que se genera a través del sistema o respuesta esperada.

La metodología de Yourdon/De Marco incluye una serie de eventos, que son una descripción o narrativa de los diferentes sucesos o estímulos que ocurren en el ambiente exterior y a los que debe dar una respuesta el sistema diseñado.

La utilización de esta metodología permitirá que el modelado del sistema integre las siguientes cualidades:

- **Gráfica**, para asegurar su comprensión.
- **Concisa**, plasmando la importancia de la brevedad.
- **Top – down**, desarrollo secuencial de arriba a abajo.
- **Particionada**, fragmentación de las piezas tan pequeñas como se requiera y tan independientes como lo permita.
- **No redundante**, cada componente de información será refinada y definida una vez y solo una vez, permitiendo asegurar la consistencia.
- **Esencial**, que requiere hacer el sistema, no como se configurara.

Yourdon y De Marco utilizaron los Diagramas de Flujo de Datos (DFD) en los que se representa las funciones y sus interfaces entre los componentes, así como la situación desde el punto de vista de cualquier persona, empleado para ello:

- Diagrama de contexto.
- Modelo lógico, donde se utilizarán Diagramas de Flujo de Datos DFD para presentarlo, al igual que Diagramas de Flujo de Procesos.
- Elaboración del diagrama entidad/relación, donde se modela lógicamente las estructuras de datos del sistema relacionado.
 - Entidades.
 - Relaciones.
- Recopilación del diccionario de datos.
- Normalización.

4.1.2 Diagrama de contexto.

Para desarrollar el sistema de una forma profesional es indispensable seguir una metodología con el fin de obtener un software de calidad y que sea una buena solución al problema. El modelado es importante porque con él podemos explicar el comportamiento del sistema, además de tener un mayor control sobre el mismo. Podemos tener modelos que proporcionan información desde la más general hasta las más detalladas.

El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema.

El diagrama de contexto muestra a través de flujos de datos las interacciones existentes entre los agentes externos y el sistema, sin describir en ningún momento la estructura del sistema de información.

Teniendo en cuenta que este diagrama debe de ser comprensible, no es posible representar todos los flujos de datos del sistema en él, sino más bien debe representarse en él una visión general del sistema desde la perspectiva de los propietarios del sistema siguiendo dos lineamientos básicos:

- Representar únicamente los flujos de datos que tengan algo que ver con el objetivo principal del sistema.
- Utilizar flujos de datos compuestos que representen a aquellos que sean similares.

Dentro de éste diagrama se enfatizan varias características importantes del sistema:

- Las personas, organizaciones y sistemas con los que se comunica el sistema. Son conocidos como terminadores.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos producidos por el sistema y que se enviarán al exterior.
- Los almacenes de datos que el sistema comparte con los terminadores.
- La frontera entre el sistema y el resto del mundo.

Para hacer este diagrama utilizaremos las convenciones de modelado del Diagrama de Yourdon/De Marco para construir el modelo lógico que son conformadas por los elementos de la figura 4.1.2.1 utilizados a un nivel de

generalidad muy alto, viendo a éste como una burbuja de un solo proceso, el Sistema de Administración Documental que interactúa con los actores o entidades externas y los medios de almacenamiento o repositorio de datos.

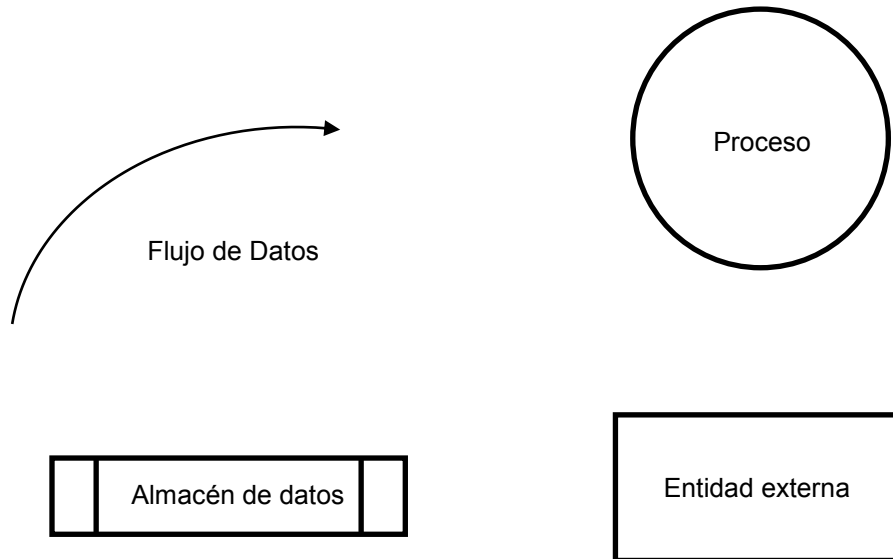


Figura 4.1.2.1 Elementos del diagrama de contexto

El Diagrama de contexto de nuestro sistema se muestra en la Figura 4.1.2.2

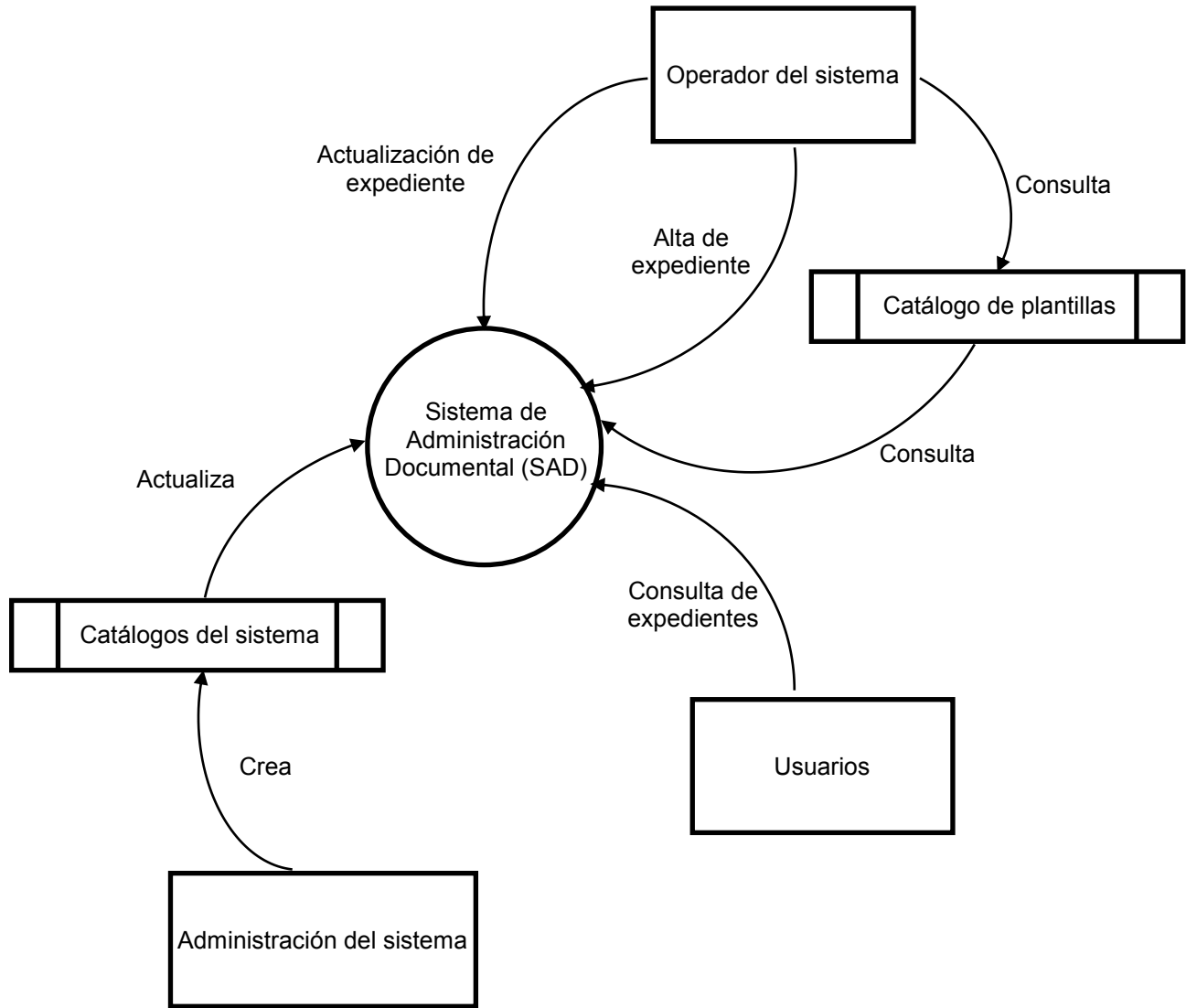


Figura. 4.1.2.2 Diagrama de contexto del Sistema de Administración Documental (SAD).

En la figura 4.1.2.2 se muestra de forma general el proceso del sistema como una burbuja que involucra todas sus funciones indispensables para su óptima operación.

También se muestran los actores o almacenes de datos que interactúan directamente con el sistema, a continuación describiremos de manera general el tipo de interacción que realizan:

- **Operador del sistema**, representa al personal encargado de crear y actualizar los expedientes de un área determinada.
- **Catálogo de plantillas**, almacén de datos donde se tendrá la información necesaria para llevar a cabo la interacción con el sistema para consultar los tipos de plantillas y dar de alta expedientes en un área determinada.
- **Catálogos del sistema**, almacén de datos donde se tendrá la información necesaria para llevar a cabo la interacción con el sistema para consultar usuarios y áreas.
- **Usuarios**, entidad que lleva a cabo la interacción con el sistema para consultar los expedientes, realizar búsquedas e imprimir la integración de expedientes.
- **Administración del sistema**, entidad que lleva a cabo la interacción con el sistema crear usuarios, áreas y plantillas, para establecer los privilegios de los usuarios, definir los tipos de expedientes y validar la bitácora del sistema.

4.1.3 Diagrama de flujo de datos.

El diagrama de flujo de datos (DFD) es una representación de forma gráfica de los algoritmos o procesos de un sistema de información. Un diagrama de flujo representa las actividades de trabajo paso por paso de un proceso y operacionales de los componentes de un sistema.

Los diagramas de flujo se datos fueron inventados por Larry Constantine. Para la representación de estos diagramas se utilizan símbolos con significados definidos que representan los pasos del algoritmo y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso.

La principal ventaja de hacer uso de los diagramas de flujo de datos es la de facilitar la comprensión de los procesos involucrados reduciendo considerablemente el número de páginas de texto. Además de poder identificar las áreas de oportunidad del proceso, y constituyen información muy recurrida para una explicación sintetizada de un proceso.

Los diagramas de flujo se deben considerar los siguientes puntos:

- Existe un único inicio del proceso.
- Existe un camino que permite llegar a una solución.
- Existe un único punto final para el proceso, a excepción del rombo que indica una comparación y da como resultado dos caminos posibles.

La figura 4.1.3.1 muestra el DFD general de la construcción del sistema.

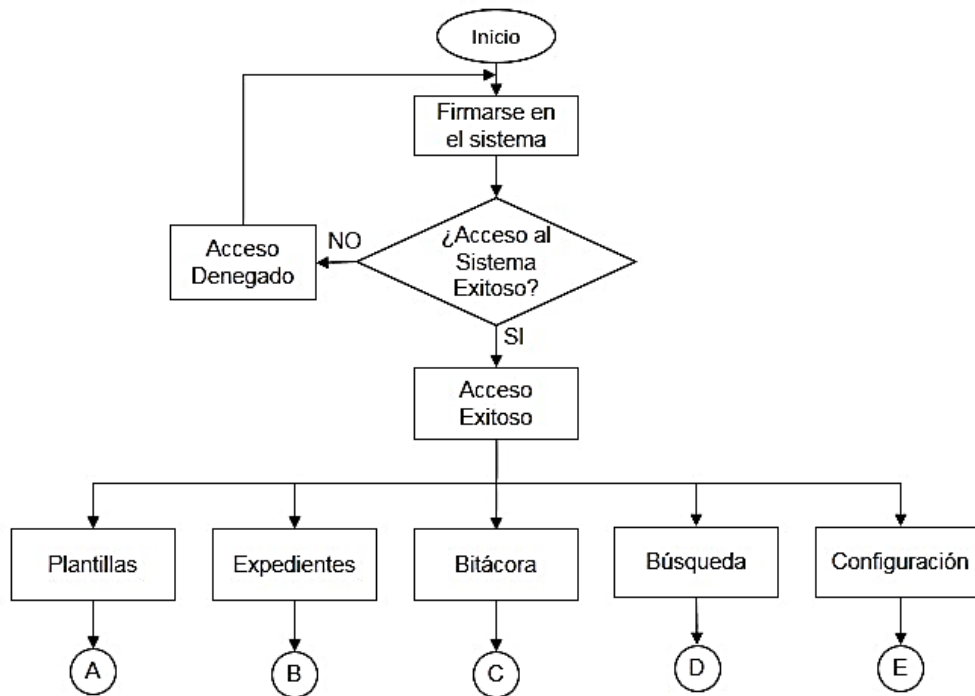


Figura 4.1.3.1 Diagrama de flujo general

Plantillas.

Este módulo permitirá gestionar las plantillas que servirán para crear una estructura tipo árbol para posteriormente asociarlas con expedientes ingresados por los usuarios. Ver figura 4.1.3.2

- **Alta de plantillas.-** Después de especificar el nombre de la plantilla, el usuario puede crear los apartados según sus necesidades, creando una estructura de árbol.
- **Consulta de plantillas.-** Realiza una consulta de las plantillas y apartados que hayan sido creados en el sistema por el usuario. El esquema de visualización es de una estructura de árbol. La consulta es de un modo gráfico, suele ser más efectivo para un manejo fácil.

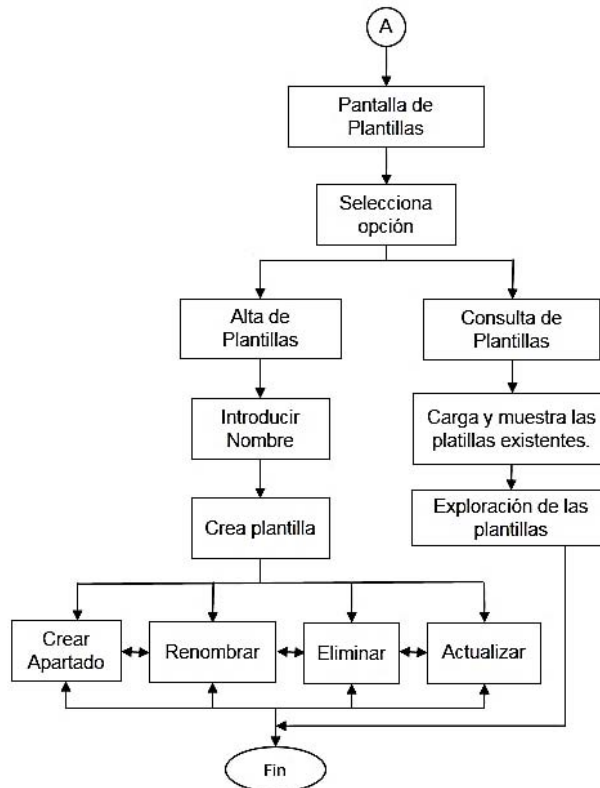


Figura 4.1.3.2 Diagrama de flujo de plantillas

Expedientes.

Es uno de los módulos más importantes de todo el sistema, ya que en él se concentrará todos los documentos y expedientes. En él se permite el alta, administración y consulta de los expedientes. Ver figura 4.1.3.3

- **Alta de expedientes.**- permite crear un expediente nuevo en el sistema, se debe ingresar el nombre, área administrativa a la que pertenece y ligarlo con una plantilla.
- **Administración de expedientes.**- principalmente permite agregar la documentación a los expedientes así como eliminarlos o renombrarlos.
- **Consulta de expedientes.**- Realiza una consulta de los expedientes y sus respectivos documentos que se encuentran en el sistema.

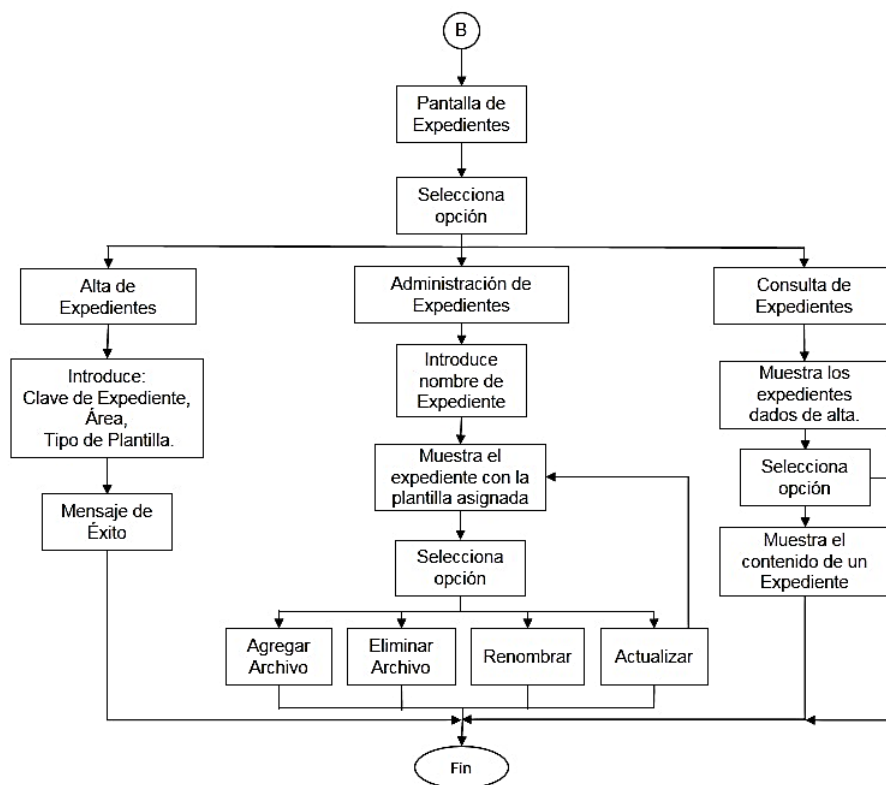


Figura 4.1.3.3 Diagrama de flujo de expedientes

Bitácora.

Es el modulo encargado de registrar todos los cambios que se hayan realizado en el sistema, tiene como finalidad poder visualizar los movimientos de expedientes, plantillas, usuarios, áreas, archivos y cambio de atributos de los usuarios. La

consulta mostrará el usuario, tipo de operación, fecha y hora del cambio realizado.
Ver figura 4.1.3.4

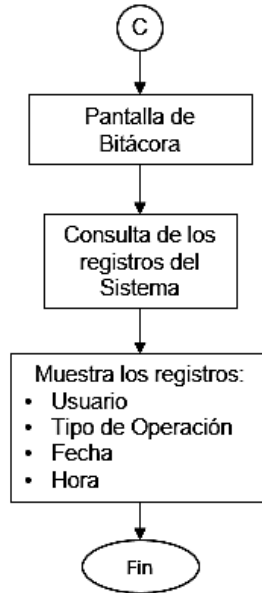


Figura 4.1.3.4 Diagrama de flujo de la bitácora

Búsquedas.

Es el modulo ayudará al usuario a encontrar los expedientes, archivos y apartados de una manera más sencilla, el usuario únicamente deberá ingresar el nombre de lo que desea encontrar y el sistema mostrará todas las coincidencias. Ver figura 4.1.3.5

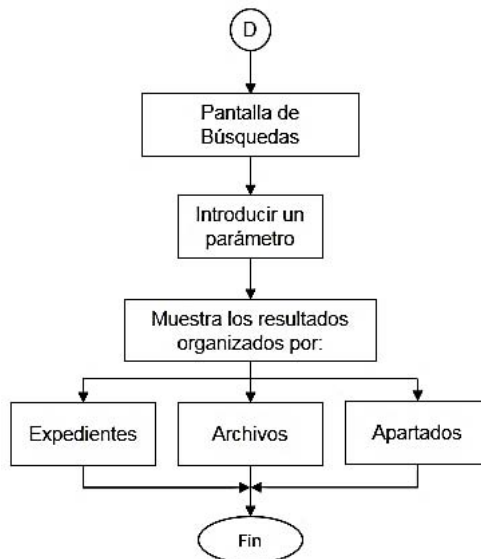


Figura 4.1.3.5 Diagrama de flujo de las búsquedas.

Configuraciones.

Por último en este módulo se encargará de administrar las cuentas de usuarios en el sistema, además de poder ajustar parámetros como lo son: cambios de área administrativa, permisos de acceso, cambios de contraseña, entre otros. Ver figura 4.1.3.6.

- **Cambio de contraseña.-** el usuario podrá cambiar su contraseña, solo debe validarse la contraseña anterior y si es correcta el sistema podrá efectuar el cambio solicitado.
- **Usuarios.-** el administrador podrá cambiar de estado (activo o inactivo) a un usuario dado de alta, además de dar de alta, eliminarlo o incluso cambiarlo de área administrativa.
- **Tipos de usuario.-** el administrador podrá asignar los permisos de acceso al sistema enlazando a un usuario con los permisos que están catalogados como: administrador, estándar o consulta.
- **Área.-** aquí se podrán dar de alta las diferentes áreas administrativas existentes en la institución gubernamental.

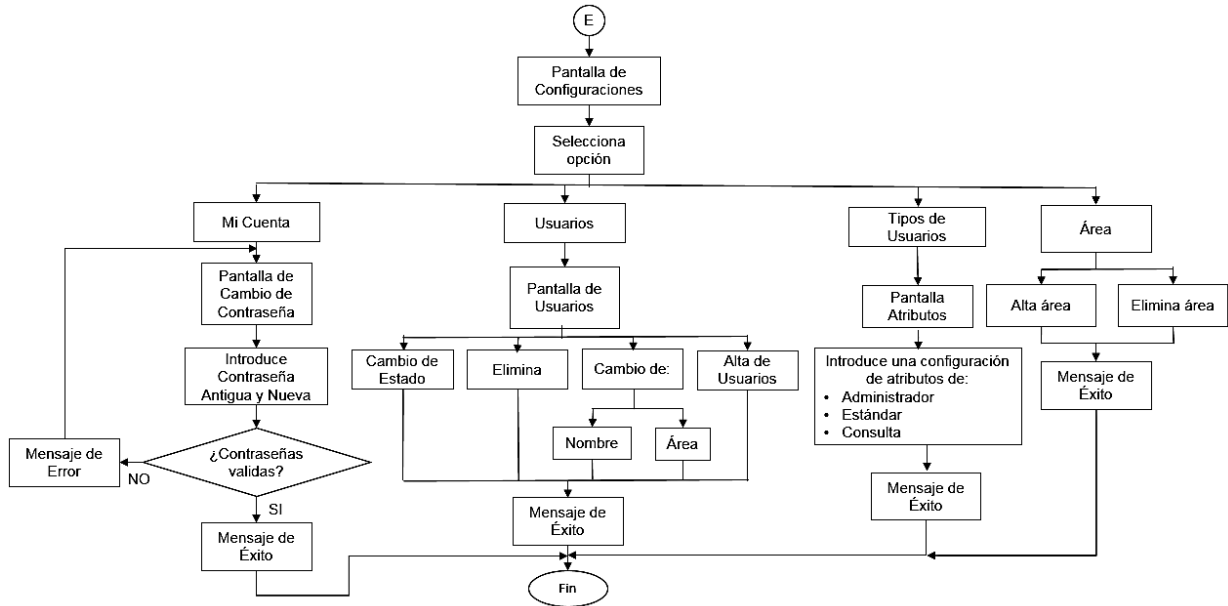


Figura 4.1.3.6 Diagrama de flujo de configuraciones

4.1.4 Diagrama de procesos.

El diagrama de procesos nos permitirá identificar a las personas o entidades que se comunican con el sistema y recibirá datos del medio ambiente que serán sus entradas necesarias para producir datos y se identificarán como salida. El sistema debe reconocer cada flujo de entrada para identificar que ha ocurrido un evento y cada evento debe generar salidas inmediatas como respuesta o bien almacenar datos, que posteriormente se convertirán en salida o un indicador de un cambio dentro del sistema. La figura 4.1.4.1 muestra el diagrama de contexto.

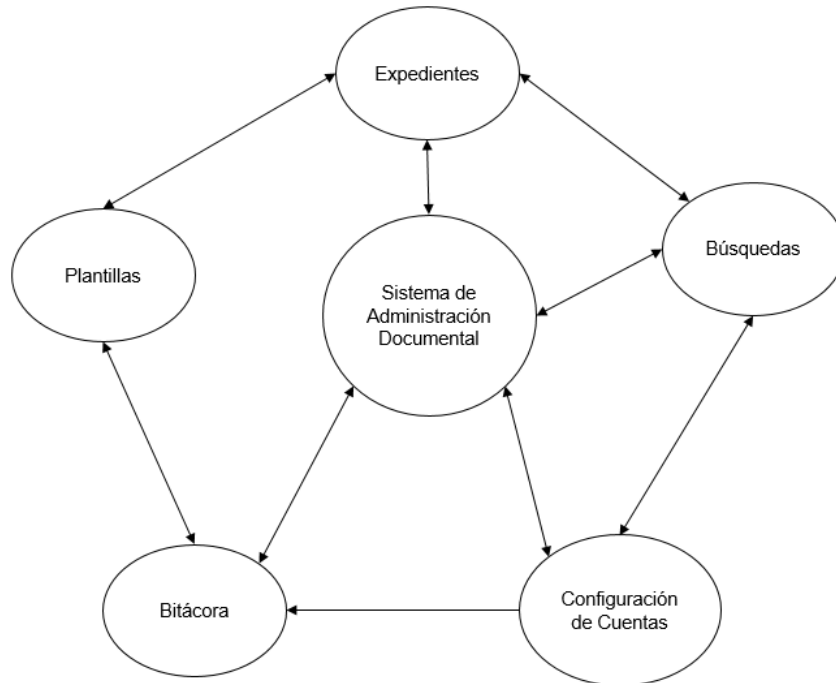


Figura 4.1.4.1 Diagrama de contexto del sistema en Nivel 0

Los procesos identificados para la resolución del problema, son los siguientes:

- Proceso de gestión de plantillas
- Proceso de gestión de expedientes
- Proceso de consulta de la bitácora del sistema
- Proceso de búsqueda de archivos, apartados y expedientes
- Proceso de gestión de usuarios y áreas

Diagrama de procesos general.

En el diagrama de procesos general muestra la interacción entre los módulos que componen nuestro sistema. Ver figura 4.1.4.2.

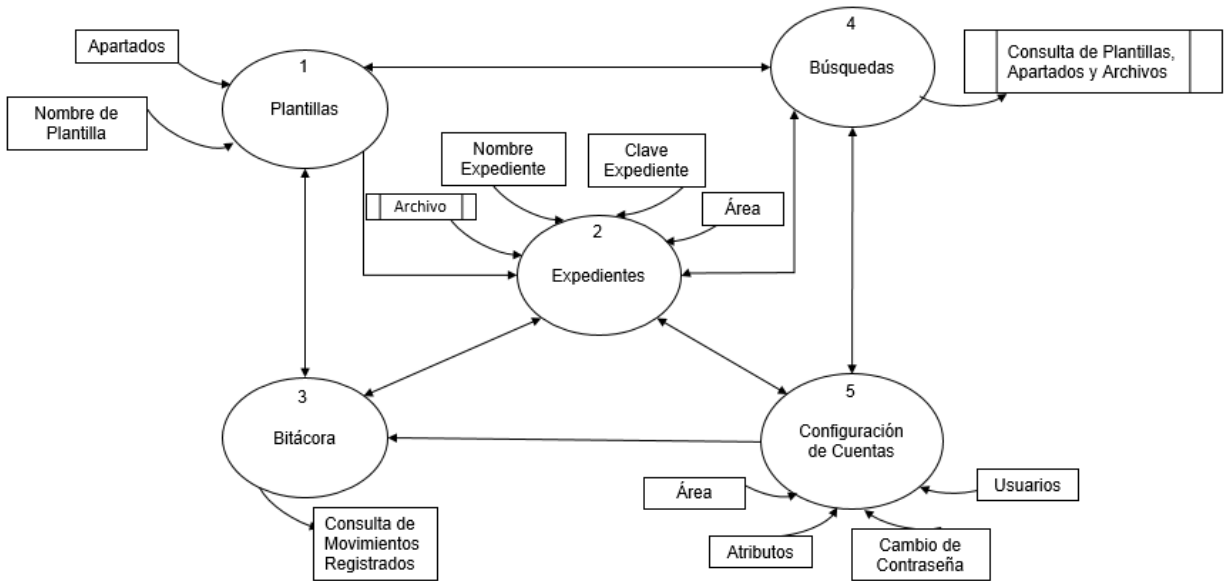


Figura 4.1.4.2 Diagrama de procesos general (Nivel I)

A continuación veremos algunos diagramas de procesos de los módulos de nuestro sistema.

El modulo de Plantillas es fundamental en este proyecto, debido a que proporcionará los apartados de los expedientes y ayudará a mantener en orden nuestra documentación. Ver figura 4.1.4.3.

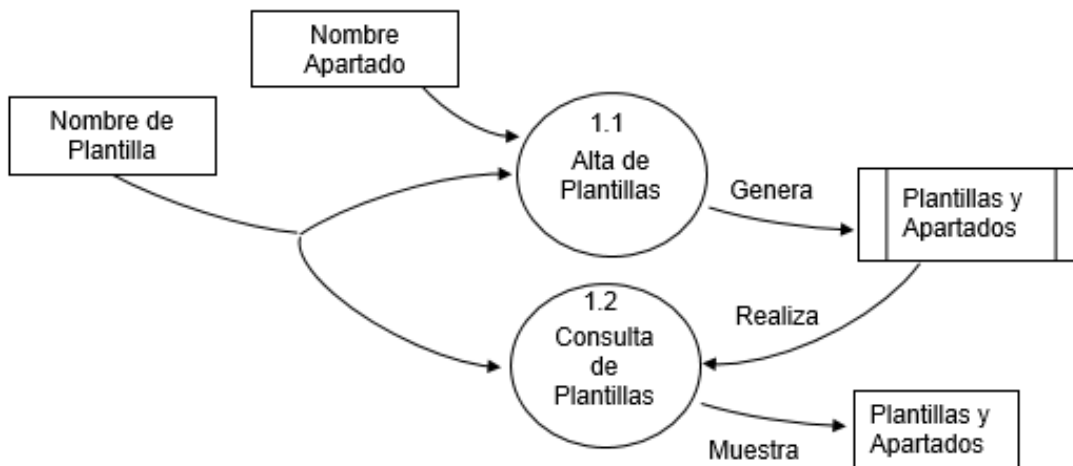


Figura 4.1.4.3 Diagrama de procesos de plantillas (Nivel II)

Especificación del proceso 1.1 Alta de plantillas.

Recibe el nombre de la plantilla, la crea y posteriormente se permite crear los apartados que requiera el usuario creando una especie de árbol, dicha estructura va a permitir organizar la documentación de una forma más ordenada. Por último el usuario puede renombrar y eliminar la plantilla y/o los apartados las veces que lo requiera.

Especificación del Proceso 1.2 Consulta de plantillas.

Realiza una consulta de las plantillas y apartados que hayan sido creados en el sistema por el usuario. El esquema de visualización es de una estructura de árbol. La consulta es de un modo gráfico, suele ser más efectivo para un manejo fácil.

El modulo de Expedientes es la razón de ser del proyecto ya que en esta parte se gestiona toda la documentación a resguardar de la institución. Ver figura 4.1.4.4

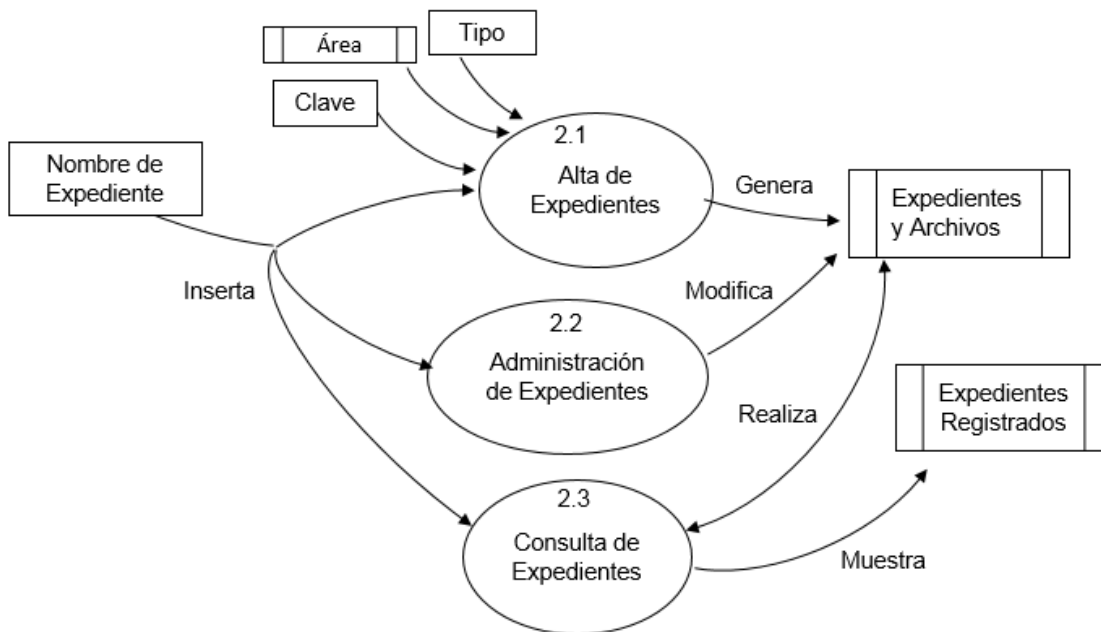


Figura 4.1.4.4 Diagrama de procesos de expedientes (Nivel II)

Vale la pena señalar hasta en un tercer nivel el proceso 2.1 que corresponde al alta de expedientes y la inserción de datos en la base de datos. Ver figura 4.1.4.5

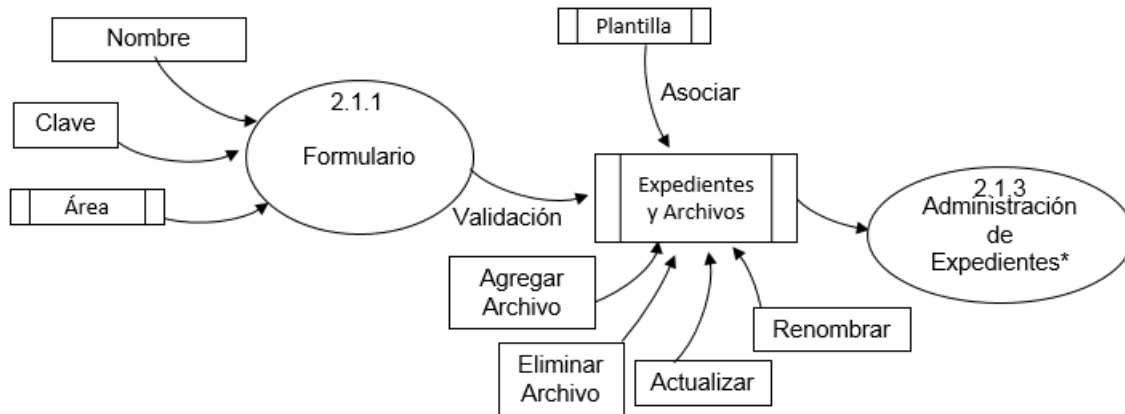


Figura 4.1.4.5 Diagrama de procesos de expedientes (Nivel III)

Especificación del proceso 2.1 Alta de expedientes.

El proceso permite crear expedientes, estos funcionarán como contenedores de los archivos que posteriormente el usuario insertará. Los parámetros que se reciben para crear un expediente son: la clave de expediente, área y el tipo de plantilla. En la primera el usuario debe insertar una cadena de caracteres alfanuméricos, después se debe especificar alguna de las áreas y el tipo de plantilla que están dadas de alta en el sistema. Por último se validan los parámetros insertados y si todo es correcto mostrará un mensaje de éxito.

Especificación del proceso 2.2 Administración de expedientes.

Este proceso es el encargado de la gestión de los expedientes ya creados en el sistema. Inicialmente el usuario debe especificar el nombre del expediente que quiere administrar y el sistema devolverá el mismo agregándole la plantilla asignada. Las opciones que el usuario podrá ejecutar son el de agregar y/o eliminar documentos o incluso renombrarlos.

Especificación del proceso 2.3 Consulta de expedientes.

Es el proceso encargado únicamente de consultar los expedientes dados de alta. Está orientado principalmente para que un grupo específico de usuarios que tenga el atributo de consultas, así solamente podrá ver la documentación contenida en los expedientes.

El modulo de la Bitácora ayuda al administrador y dueño de la información quienes son los usuarios que han modificado la información contenida en el sistema. Ver figura 4.1.4.6

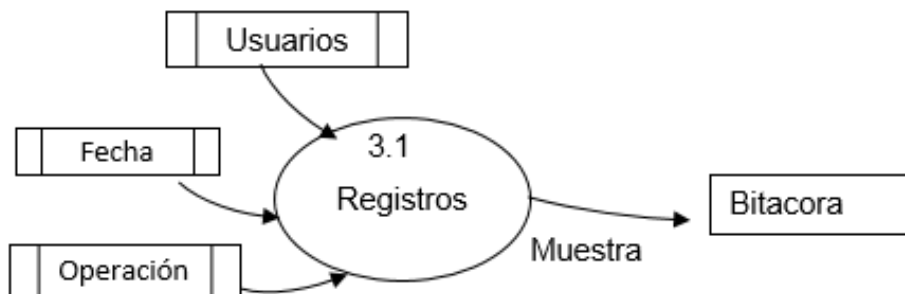


Figura 4.1.4.6 Diagrama de procesos de la bitácora (Nivel II)

Especificación del Proceso 3.1 Consulta de la bitácora del sistema.

El tema de la seguridad es muy importante en este sistema, por ello es indispensable que exista una bitácora donde se registren todos los cambios realizados por los usuarios, especificando que tipo de acción realizó, fecha y hora. Estos registros no pueden ser eliminados, únicamente son de consulta incluso para los administradores.

El modulo de las búsquedas, ayuda al usuario a una rapida localización de expedientes, archivos y apartados. Ver figura 4.1.4.7

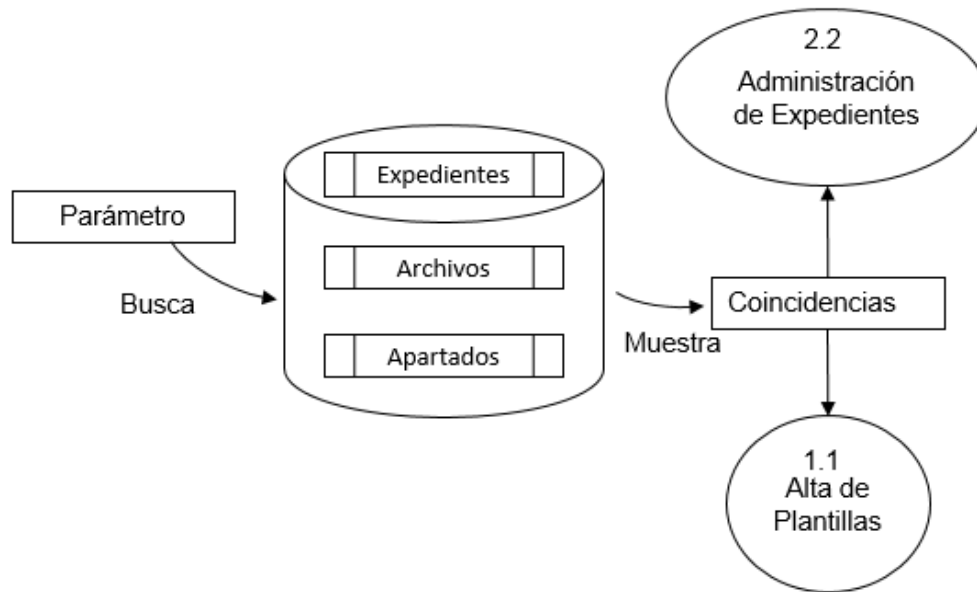


Figura 4.1.4.7 Diagrama de procesos de las búsquedas (Nivel II)

Especificación del proceso 4.1 Consultas de las búsquedas de expedientes, archivos y apartados.

Este proceso recibe una cadena de caracteres alfanuméricos que son ingresados por el usuario y el sistema devuelve una consulta de los expedientes, archivos y apartados que se encuentren existentes en el sistema. Es de gran importancia este proceso ya que al paso del tiempo el volumen de información va incrementando.

El modulo de la Configuración de cuentas o usuarios, el administrador podrá gestionar todas las configuraciones de una cuenta de usuario, así como sus atributos. Ver figura 4.1.4.8

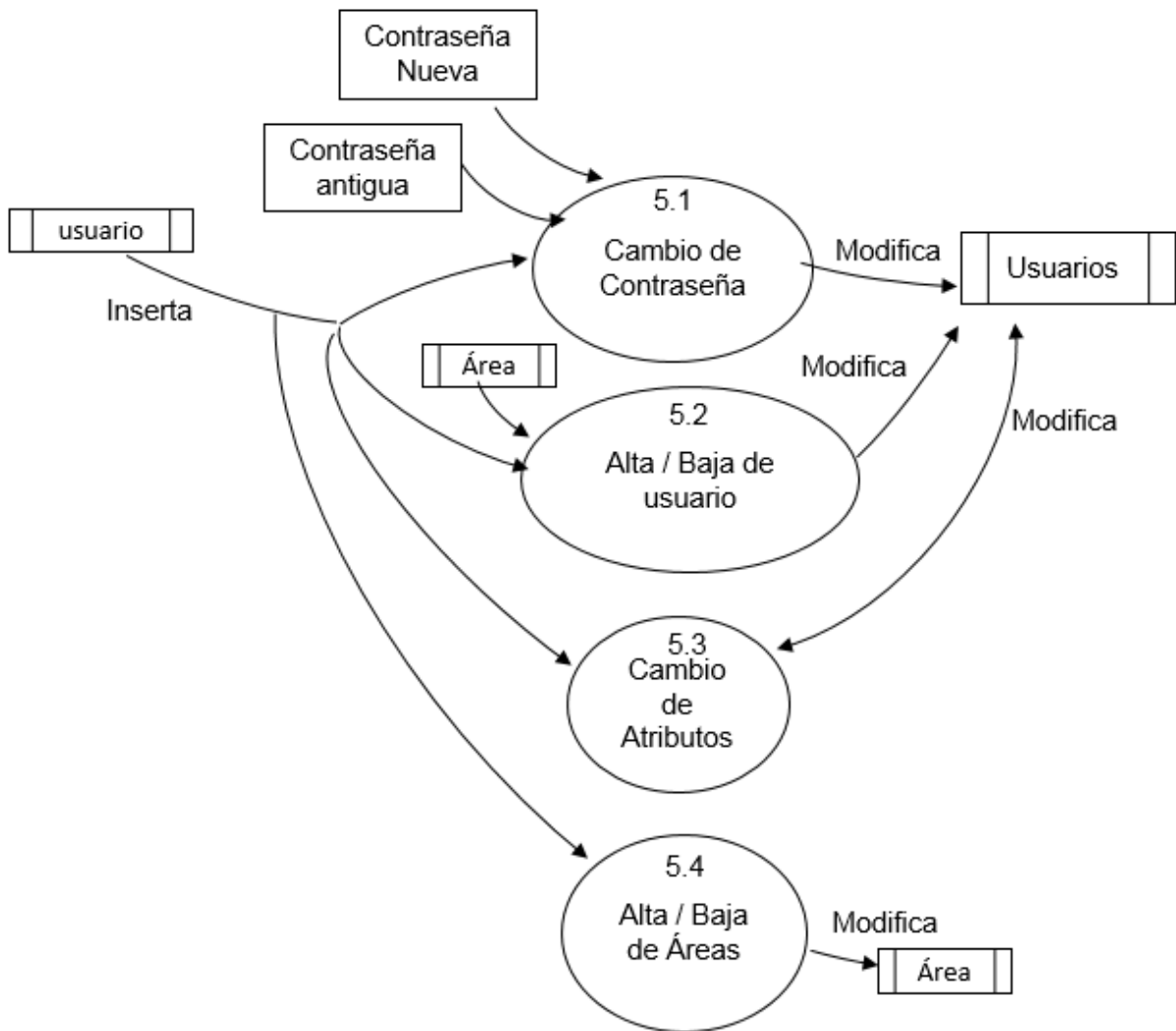


Figura 4.1.4.8 Diagrama de procesos de configuración de cuentas (Nivel II)

Especificación del proceso 5.1 Configuración de la cuenta.

Este proceso de configuración de la cuenta del usuario, permite cambiar de contraseña, por medidas de seguridad debe ser insertada la contraseña anterior y la nueva contraseña debe ser escrita dos veces para descartar un error de dedo. Si los parámetros anteriores son válidos el sistema actualiza la contraseña y arroja un mensaje de éxito al usuario.

Especificación del proceso 5.2 Configuración de los usuarios.

El proceso de configuración de usuarios está diseñado para que un usuario con privilegios de administrador pueda gestionar las altas y/o bajas de otros usuarios, cambios de nombre, áreas incluso un cambio de estado, este último va orientado a habilitar o deshabilitar temporalmente a un usuario. Una vez concluido un cambio, el sistema muestra al usuario un mensaje de éxito.

Especificación del proceso 5.3 Configuración de los tipos de usuarios.

Debido a que no todos los usuarios pueden ser administradores este proceso es el encargado de la gestión de privilegios. Los 3 atributos disponibles son administrador que es el que tiene control absoluto del sistema, el estándar que puede crear apartados, plantillas subir y/o bajar información, por último el de consulta, este únicamente puede ver información.

Especificación del proceso 5.4 Configuración del área de trabajo.

Este proceso es el encargado de dar de alta y/o baja las áreas administrativas que pertenecen a la Institución. Una vez que el usuario realiza un cambio el sistema le devuelve un mensaje de éxito.

4.1.5 Diagrama entidad-relación del Sistema de Administración Documental (SAD).

El modelo o diagrama entidad-relación es la herramienta que nos permite modelar los datos mediante entidades y las relaciones entre ellas así como sus propiedades en una base de datos para un sistema de información. Los objetivos de un diagrama entidad-relación son:

- Representar las necesidades de información para un sistema en modo gráfico.
- Ayudar a prevenir errores.
- Es la base en el diseño de la base de datos.

Los principales componentes en un diagrama entidad-relación son:

- **Entidades:** lo podemos definir como cualquier objeto, real o abstracto, que existe en un contexto determinado o pudiera llegar a existir y del cual se desea guardar información.
 - **Atributos:** es una característica de una Entidad la cual puede identificar, relacionar o describir.
- **Relaciones:** es la conexión que puede haber entre los atributos de entidades.
- **Cardinalidad:** expresa el número de ocurrencias que existen entre un conjunto de entidades.
- **Llave primaria:** conjunto de uno o más atributos de una entidad que se encargaran de identificar los registros de una entidad como único.
- **Llave foránea:** es uno o más campos de una entidad que hacen referencia a uno o un conjunto de campos de atributos que son llave primaria en otra entidad.

El modelo entidad-relación del Sistema de Administración Documental está representado en la figura 4.1.5.1 nos muestra las diferentes entidades así como se interrelacionan entre ellas.

Diagrama entidad relación del Sistema de Administración Documental.

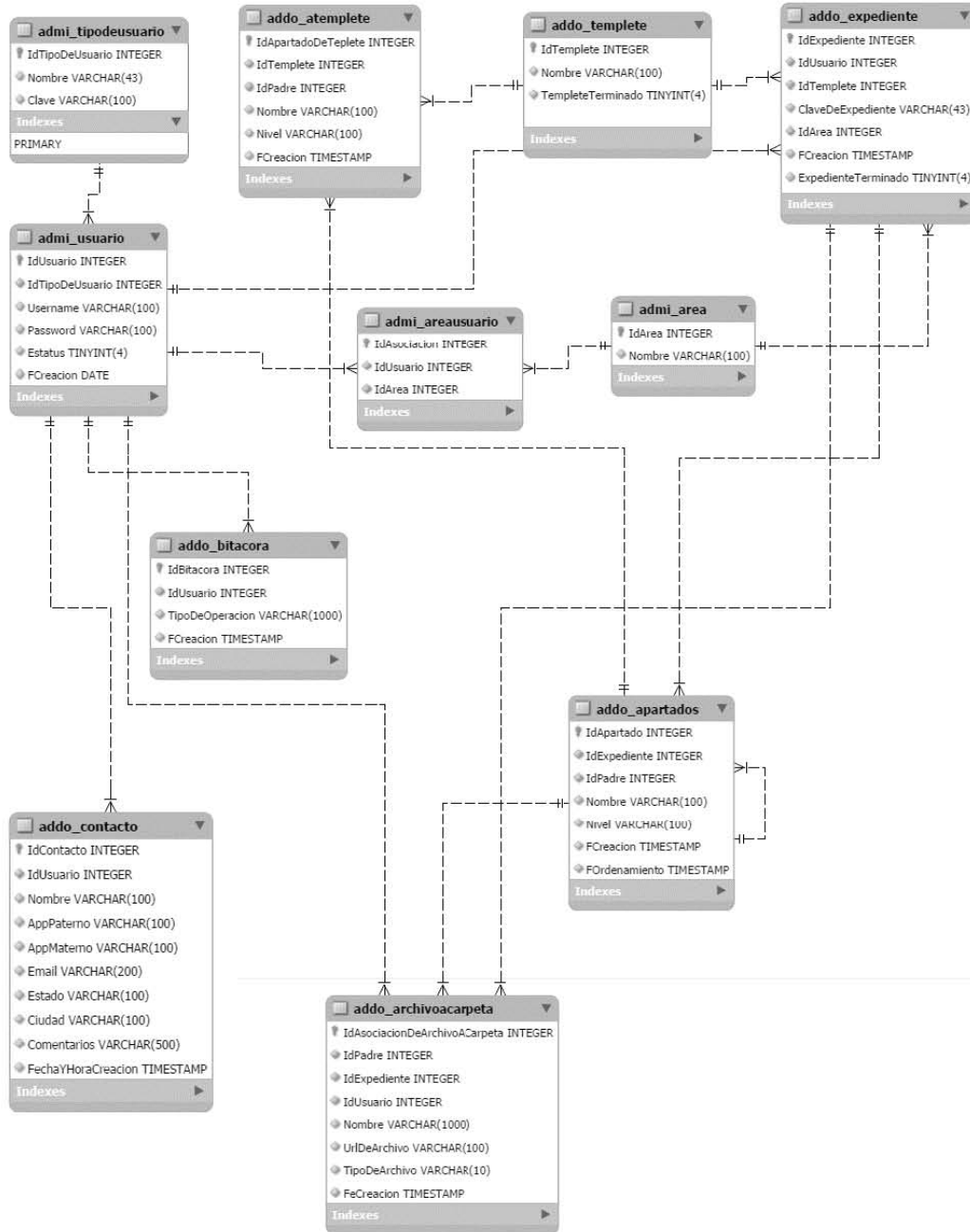


Figura 4.1.5.1 Diagrama entidad relación.

4.1.6 Diccionario de datos del Sistema de Administración Documental (SAD).

El diccionario de datos se puede definir como el conjunto de tablas que verifican, registran y proporcionan información, que describen la base de datos.

El diccionario es muy importante ya que contiene todos los objetos de la base de datos con su respectivo nombre y características de sus atributos, en resumen contiene metadatos y es usado frecuentemente para los nombres de los objetos; en los siguiente cuadros se mostrara el diccionario de la solución propuesta del Sistema de Administración Documental.

Estos son utilizados para describir el propósito de la base de datos, describir de manera detallada cada tabla dentro de la base de datos, documentar la estructura de cada tabla incluyendo sus campos y los tipos de datos con comentario.

La tabla `admi_usuario` se encargara de almacenar la información necesaria para realizar el login al sistema, guardando el usuario, contraseña y el estado del usuario dentro del sistema, figura 4.1.6.1

Tabla `admi_usuario`.

| nombre | acronimo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-----------------------|-----------------|-----------|--------|-------|------|--------------------------|----------------|
| Usuario ID | IdUsuario | INTEGER | 11 | PK | No | Identificador de usuario | |
| Nombre de usuario | Username | VARCHAR | 100 | | NO | Username de usuario | |
| Contraseña de usuario | Password | VARCHAR | 100 | | NO | Password de usuario | |
| Estado | Estatus | TINYINT | 4 | | NO | Estado del usuario | |
| Fecha de vigencia | FechaDeVigencia | TIMESTAMP | | | NO | Fecha de vigencia | |

Figura 4.1.6.1 Tabla `admin_usuario`.

La tabla `admi_areausuario` se encarga de mantener la relación del usuario con respecto a un área o áreas en específico a la que se encuentre asignado dentro

de la empresa, esta tabla está asociada con las tablas usuario y área, figura 4.1.6.2.

Tabla admi_areausuario.

| Nombre | Acrónimo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|---------------|--------------|---------|--------|-------|------|-----------------------------|----------------|
| Asociación ID | IdAsociacion | INTEGER | 11 | PK | NO | Identificador de asociación | |
| Usuario ID | IdUsuario | INTEGER | 11 | FK | NO | Llave foránea | admi_usuario |
| Área ID | IdArea | INTEGER | 11 | FK | NO | Llave foránea | admi_area |

Figura 4.1.6.2 Tabla admi_areausuario.

Tabla admi_area se encargara de almacenar el nombre de las diferentes áreas involucradas en el sistema; ver figura 4.1.6.3.

Tabla admi_area.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-------------|--------|---------|--------|-------|------|------------------------|----------------|
| Área ID | IdArea | INTEGER | 11 | PK | NO | Identificador del área | |
| Nombre área | Nombre | VARCHAR | 100 | | NO | Nombre del área | |

Figura 4.1.6.3 Tabla admi_area.

Tabla admi_apartado almacena los apartados de las plantillas a utilizarse para guardar los expediente. Tablas asociadas admi_expediente y admi_apartado. Ver figura 4.1.6.4

Tabla admi_apartado.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|---------------|--------------|---------|--------|-------|------|----------------------------|-----------------|
| Apartado ID | IdApartado | INTEGER | 11 | PK | NO | Identificador de apartados | |
| Expediente ID | IdExpediente | INTEGER | 11 | FK | NO | Llave foránea | admi_expediente |

Capítulo IV Análisis, diseño y construcción de la aplicación

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-----------------------|---------------|-----------|--------|----------|------|---------------------------------------|-------------------|
| Padre ID | IdPadre | INTEGER | 11 | FK | NO | Llave foránea | admi_apartado |
| Nombre apartado | Nombre | VARCHAR | 100 | | NO | Nombre apartado | |
| Nivel del apartado | Nivel | VARCHAR | 100 | | NO | Nivel en que se encuentra el apartado | |
| Fecha de Creación | FCreacion | TIMESTAMP | | | NO | Fecha de creación del apartado | |
| Fecha de ordenamiento | Fordenamiento | TIMESTAMP | | | NO | Fecha de ordenamiento | |

Figura 4.1.6.4 Tabla admi_apartado.

La tabla admi_contacto será la encargada de mantener almacenada la información personal de los usuarios, esta tabla tendrá relacionada con la tabla usuario mediante el id del usuario. Ver figura 4.1.6.5

Tabla admi_contacto.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|--------------------------|------------|---------|--------|----------|------|------------------------------|-------------------|
| Contacto ID | IdContacto | INTEGER | 11 | PK | NO | Identificador de contacto | |
| Usuario ID | IdUsuario | INTEGER | 11 | FK | NO | Llave foránea | admi_usuario |
| Nombre usuario | Nombre | VARCHAR | 100 | | NO | Nombre usuario | |
| Apellido paterno usuario | AppPaterno | VARCHAR | 100 | | NO | Apellido paterno del usuario | |
| Apellido materno usuario | AppMaterno | VARCHAR | 100 | | NO | Apellido materno del usuario | |
| Email usuario | Email | VARCHAR | 200 | | NO | Email del usuario | |
| Estado usuario | Estado | VARCHAR | 100 | | YES | Estado usuario | |
| Ciudad del usuario | Ciudad | VARCHAR | 100 | | YES | Ciudad usuario | |

Capítulo IV Análisis, diseño y construcción de la aplicación

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-------------------------|-------------|-----------|--------|----------|------|---------------------|----------------|
| Comentarios del usuario | Comentarios | VARCHAR | 500 | | NO | Comentarios usuario | |
| Fecha de creación | FCreacion | TIMESTAMP | | | NO | Fecha de creación | |

Figura 4.1.6.5 Tabla adm_i_contacto.

Tabla en la cual se almacena toda la información necesaria de los expedientes, quien lo creo y a qué área estaba asignado, cuando lo crearon y que plantilla está usando para guardar la información. Ver figura 4.1.6.6

Tabla adm_i_expediente.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|---------------------------|---------------------|-----------|--------|----------|------|-------------------------------------|----------------|
| Expediente ID | IdExpediente | INTEGER | 11 | PK | NO | Identificador expediente | |
| Usuario ID | IdUsuario | INTEGER | 11 | FK | NO | Llave foránea | adm_i_usuario |
| Template ID | IdTemplate | INTEGER | 11 | FK | NO | Llave foránea | adm_i_template |
| Área ID | IdArea | INTEGER | 11 | FK | NO | Llave foránea | adm_i_area |
| Clave del expediente | ClaveDeExpediente | VARCHAR | 43 | | NO | Clave del expediente | |
| Fecha creación expediente | FechaDeCreacion | TIMESTAMP | | | NO | Fecha de la creación del expediente | |
| Estado expediente | ExpedienteTerminado | TINYNT | 4 | | NO | Estado del expediente | |

Figura 4.1.6.6 Tabla adm_i_expediente.

Tabla que guarda solamente el nombre de la plantilla y su estado. Ver figura 4.1.6.7.

Tabla adm_i_template.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-----------------|-------------------|---------|--------|----------|------|------------------------|----------------|
| Template ID | IdTemplate | INTEGER | 11 | PK | NO | Identificador template | |
| Nombre template | Nombre | VARCHAR | 100 | | NO | Nombre template | |
| Estado Template | TemplateTerminado | TINYINT | 4 | | NO | Estatus de template | |

Figura 4.1.6.7 Tabla adm_i_template.

Tablas relacionadas: adm_i_template y adm_i_apartado. figura 4.1.6.8

Tabla atemplate.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|-----------------------------|----------------------|-----------|--------|----------|------|-------------------------------------|----------------|
| Apartado Template ID | IdApartadoDeTemplate | INTEGER | 11 | PK | NO | Identificador de apartado template | |
| Template ID | IdTemplate | INTEGER | 11 | FK | NO | Llave foránea | adm_i_template |
| Padre ID | IdPadre | INTEGER | 11 | FK | NO | Llave foránea | adm_i_apartado |
| Nombre apartado template | Nombre | VARCHAR | 100 | | NO | Nombre de apartado template | |
| Nivel del apartado template | Nivel | VARCHAR | 100 | | NO | Nivel de apartado template | |
| Fecha de Creacion | FCreacion | TIMESTAMP | | | NO | Fecha de creación apartado template | |

Figura 4.1.6.8 Tabla adm_i_atemplate.

Tabla adm_i_archivocarpeta encargada de guardar todas las asociaciones necesarias para identificar en que carpeta del expediente fueron alojados los documentos así como registrar a que usuario pertenece, la fecha en la que fueron subidos al sistema y la dirección en la que se almaceno. Ver figura 4.1.6.9

Tabla admi_archivocarpeta.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|----------------------------|------------------------------|-----------|--------|----------|------|--|-----------------|
| Asociación Archivo ID | IdAsociacionDeArchivoCarpeta | INTEGER | 11 | PK | NO | Identificador de asociación de archivo a una carpeta | |
| Padre ID | IdPadre | INTEGER | 11 | FK | NO | Llave foránea | admi_apartado |
| Expediente ID | IdExpediente | INTEGER | 11 | FK | NO | Llave foránea | admi_expediente |
| Usuario ID | IdUsuario | INTEGER | 11 | FK | NO | Llave foránea | admi_usuario |
| Asociación archivo carpeta | Nombre | VARCHAR | 100 | | NO | Nombre de asociación | |
| URL del archivo | UrlDeArchivo | VARCHAR | 100 | | NO | Dirección de almacenamiento del archivo | |
| Tipo de archivo | TipoDeArchivo | VARCHAR | 10 | | NO | Tipo de archivo | |
| Fecha de creación | FCreacion | TIMESTAMP | | | NO | Fecha de la creación | |

Figura 4.1.6.9 Tabla admi_archivocarpeta.

Tabla que guardara todo el historial de movimientos realizados por los usuarios mientras este dentro del sistema. Ver figura 4.1.6.10

Tabla addo_bitacora.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|---------------------|---------------|-----------|--------|----------|------|---|----------------|
| Bitácora ID | IdBitacora | INTEGER | 11 | PK | NO | Identificador en la bitácora | |
| Usuario ID | IdUsuario | INTEGER | 11 | FK | NO | Llave foránea | admi_usuario |
| Operación realizada | TipoOperacion | VARCHAR | 100 | | NO | Descripción del tipo de movimiento realizado por los usuarios | |
| Fecha de creación | FCreacion | TIMESTAMP | | | NO | Fecha en la que se realizaron | |

Figura 4.1.6.10 Tabla addo_bitacora.

Tabla en la que se definirán los tipos de usuarios que tiene el sistema. Figura 4.1.6.11

Tabla adm_i_tipousuario.

| Nombre | Campo | Tipo | Tamaño | PK FK | NULL | Descripción | Tabla asociada |
|----------------------------|---------------|---------|--------|----------|------|------------------------------------|----------------|
| Tipo de usuario ID | IdTipoUsuario | INTEGER | 11 | PK | NO | Identificador del tipo de usuario | |
| Nombre del tipo de usuario | Nombre | VARCHAR | 43 | | NO | Nombre del rol del tipo de usuario | |
| Clave del usuario | Clave | VARCHAR | 100 | | NO | Clave del tipo de usuario. | |

Figura 4.1.6.11 Tabla adm_i_tipousuario.

4.1.7 Normalización.

Como habíamos explicado en capítulos anteriores existen cinco formas normales pero en la práctica solo se usa en la mayoría de los casos hasta la Tercer Forma Normal ya que después de esta las bases de datos quedan completamente normalizadas.

Como ejemplo de normalización de nuestro sistema tomaremos una tabla no normalizada con nombres de usuarios (figura 4.1.7.1) la cual guarda información de nombre completo de usuarios, detalles de su cuenta, información del área e información general de los mismos, en este ejemplo usaremos como id la columna de nombre Clave Usuario.

Datos de ejemplo.

| Nombre | Clave | Cuenta | Contacto | Estatus | Área | Estado |
|------------------------|-------|-------------|-------------------|---------|--------------------|------------------------|
| Luis Murguía Rojas | 123 | LMR, LMR123 | LMR@correo.com | ACTIVO | Informática,123 | Guadalajara, Jalisco |
| Ignacio Cruz Cruz | 456 | ICC, ICC123 | ICC@correo.com | ACTIVO | Archivo, 456 | Distrito Federal, D.F. |
| Eliza Ortiz Ortiz | 789 | E00, E00123 | E00123@Correo.com | ACTIVO | Archivo, 456 | Puebla, Puebla |
| Luis Murguía Rojas | 12 | LMR, LMR123 | LMR123@correo.com | ACTIVO | Archivo, 456 | Guadalajara, Jalisco |
| Javier Morales Morales | 456 | JMM, JMM123 | JMM123@correo.com | ACTIVO | Adquisiciones, 789 | Veracruz, Xalapa |
| Ignacio Cruz Cruz | 890 | ICC, ICC123 | ICC123@correo.com | ACTIVO | Finanzas, 345 | Distrito Federa, D.F. |

Figura 4.1.7.1 Tabla usuario no normalizada.

Si guardáramos datos en una tabla como esta tendríamos demasiadas anomalías, ya que un usuario puede estar asignado en más de un área se tendría que agregar otro registro por cada área a la que se encuentre asignado y con esto estaríamos duplicando valores en nuestra tabla, además si nos encontramos con algún error en el nombre de algún usuario no se podría actualizar de una manera eficiente cada registro ya que no podríamos saber el número exacto de registro con errores y esto es aplicable para cualquier registro de la tabla. Estos son solo unos de los problemas a los que nos presentaríamos si trabajáramos con una tabla no normalizada.

Con la primera forma normal podemos remover los grupos repetidos transformando todos nuestros valores a atómicos. Ver figura 4.1.7.2

Tabla después de usar la primera forma normal.

| área | Clave área | Clave usuario | User | Pass | status | Estado | Ciudad | Contacto | Nombre | App Pat | App Mat |
|---------------|------------|---------------|------|--------|--------|------------------|---------|----------------|---------|---------|---------|
| Informática | 123 | 123 | LMR | LMR123 | Activo | Guadalajara | Jalisco | LMR@correo.com | Luis | Murguía | Rojas |
| Archivo | 456 | 456 | ICC | ICC123 | Activo | Distrito Federal | D.F. | ICC@correo.com | Ignacio | Cruz | Cruz |
| Archivo | 456 | 456 | E00 | E00123 | Activo | Puebla | Puebla | E00@correo.com | Eliza | Ortiz | Ortiz |
| Adquisiciones | 789 | 789 | JMM | JMM123 | Activo | Veracruz | Xalapa | JMM@correo.com | Javier | Morales | Morales |
| Archivo | 456 | 456 | LMR | LMR123 | Activo | Guadalajara | Jalisco | LMR@correo.com | Luis | Murguía | Rojas |
| Finanzas | 345 | 345 | ICC | ICC123 | Activo | Distrito Federal | D.F. | ICC@correo.com | Ignacio | Cruz | Cruz |

Figura 4.1.7.2 Tabla inicial después de aplicar la 1 FN.

Hasta este paso aún no hemos acabado con el procedimiento de normalización ya que todavía no queda completamente normalizada nuestra tabla inicial.

La segunda forma normal remueve las columnas de una tabla que no dependan de la llave primaria, en nuestro ejemplo serían las columnas con datos del área con las cuales formaríamos una tabla independiente llamada área (figura 4.1.7.3) por ultimo una que contenga los datos del usuario(figura 4.1.7.4). Aunque también podemos anexar nuevas columnas a cada tabla y esto no afectará en nada a las demás tablas.

Tabla área formada a partir de la segunda forma normal.

| Id_Area | Nombre |
|----------------|----------------------|
| 123 | Informática |
| 456 | Archivo |
| 789 | Adquisiciones |
| 345 | Finanzas |

Figura 4.1.7.3 Tabla área.

Tabla usuario después de la segunda forma normal.

| User | Pass | status | Estado | Ciudad | Contacto | Nombre | App Pat | App Mat |
|------|--------|--------|------------------|---------|----------------|---------|---------|---------|
| LMR | LMR123 | Activo | Guadalajara | Jalisco | LMR@correo.com | Luis | Murguía | Rojas |
| ICC | ICC123 | Activo | Distrito Federal | D.F. | ICC@correo.com | Ignacio | Cruz | Cruz |
| EOO | EOO123 | Activo | Puebla | Puebla | EOO@correo.com | Eliza | Ortiz | Ortiz |
| JMM | JMM123 | Activo | Veracruz | Xalapa | JMM@correo.com | Javier | Morales | Morales |
| LMR | LMR123 | Activo | Guadalajara | Jalisco | LMR@correo.com | Luis | Murguía | Rojas |
| ICC | ICC123 | Activo | Distrito Federal | D.F. | ICC@correo.com | Ignacio | Cruz | Cruz |

Figura 4.1.7.4 Tabla usuario.

Después de aplicar las dos primeras formas normales podemos observar que todavía encontramos anomalías en la tabla usuarios por lo que al aplicar la tercera forma normal dentro de la tabla usuarios nos quedaran dos tablas(figura 4.1.7.5 y figura 4.1.7.6) además de que al realizar una tabla nueva(figura 4.1.7.7) en la cual se verá las asociaciones que tengan los usuarios con las diferentes áreas dentro de la empresa ya que nuestra relación Usuario-Área es de muchos a muchos, un usuario puede estar asociado a más de un área pero también un área puede tener más de un usuario y por último la tabla área que se queda igual que en el paso donde aplicamos la segunda forma normal(figura 4.1.7.8).

Tabla contacto.

| Id Contacto | Nombre | App Pat | App Mat | Correo | Estado | Ciudad |
|--------------------|----------------|----------------|----------------|-----------------------|-------------------------|----------------|
| 987 | Luis | Murguía | Rojas | LMR@correo.com | Guadalajara | Jalisco |
| 654 | Ignacio | Cruz | Cruz | ICC@correo.com | Distrito Federal | DF |
| 321 | Eliza | Ortiz | Ortiz | E00@correo.com | Puebla | Puebla |
| 543 | Javier | Morales | Morales | JMM@correo.com | Veracruz | Xalapa |

Figura 4.1.7.5 Tabla contacto.

Tabla usuario.

| Id usuario | Username | Password | status | Id contacto | Id asociación |
|-------------------|-----------------|-----------------|---------------|--------------------|----------------------|
| 123 | LMR | LMR123 | Activo | 987 | 1 |
| 456 | ICC | ICC123 | Activo | 654 | 2 |
| 789 | E00 | E00123 | Activo | 321 | 3 |
| 345 | JMM | JMM123 | Activo | 543 | 4 |

Figura 4.1.7.6 Tabla usuario.

Tabla de asociación usuario área.

| Id asociación | Id usuario | Id area |
|----------------------|-------------------|----------------|
| 1 | 123 | 123 |
| 2 | 456 | 456 |
| 3 | 789 | 456 |

Figura 4.1.7.7 Tabla usuario área.

Tabla área.

| Id_Area | Nombre |
|----------------|----------------------|
| 123 | Informática |
| 456 | Archivo |
| 789 | Adquisiciones |
| 345 | Finanzas |

Figura 4.1.7.8 Tabla Área.

Con lo que después de aplicar las tres formas normales pasamos de una tabla con demasiados datos no normalizados (Figura 4.1.7.9) a tener cuatro tablas para almacenar de una mejor manera los datos en nuestra base de datos, en la figura 4.1.7.10 podemos observar el diagrama que se realiza entre las tablas después de aplicar las reglas de normalización.

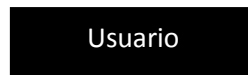


Figura 4.1.7.9 Tabla inicial no normalizada.

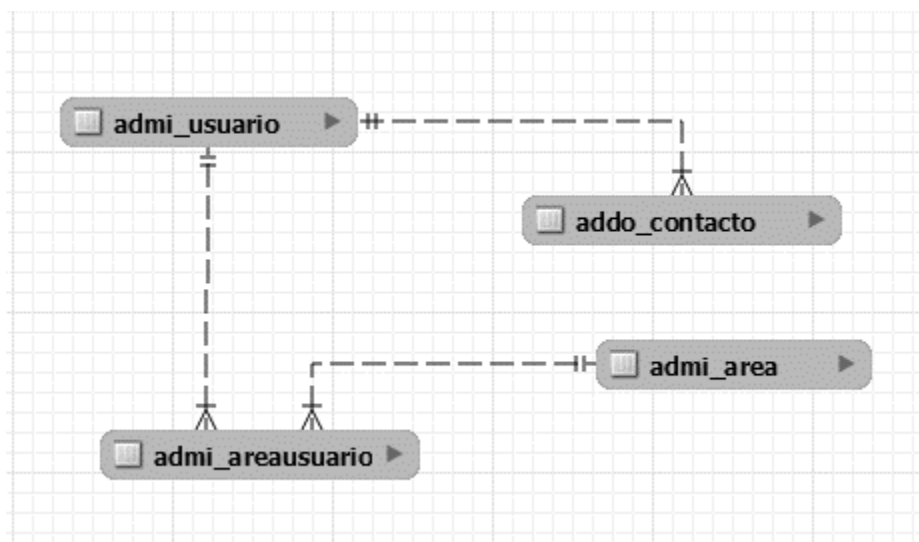


Figura 4.1.7.10 Diagrama resultado de aplicar las reglas de normalización.

4.2 Creación de la base de datos e implementación del Sistema de Administración Documental (SAD).

Instalación del manejador base de datos SQL Server 2012

Para iniciar con la instalación de SQL Server 2012, requerimos del archivo SETUP.exe previamente descargado de la página oficial de Microsoft, éste es el instalador con el cual solo debemos de seguir las indicaciones que aparecen en pantalla, debemos de escoger la opción de instalar y dar clic en la opción “Nueva instalación independiente del SQL server o agregar características a una instalación existente”. Ver figura 4.2.1.

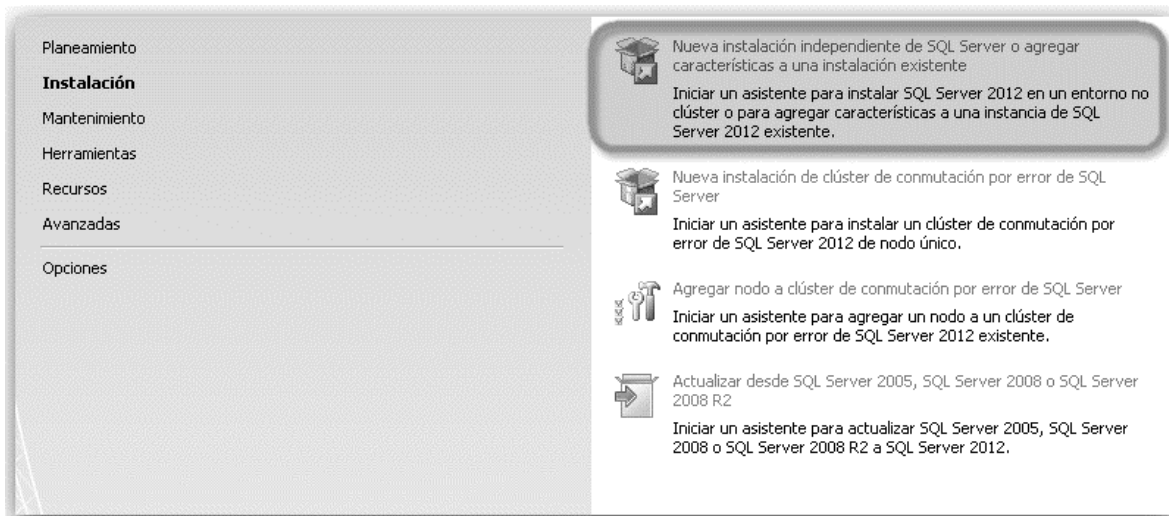


Figura 4.2.1 Inicio de la instalación.

Si el procedimiento llegase a marcar el error de la figura 4.2.2, realizar el siguiente procedimiento:

Lo primero de debemos hacer es buscar el Panel de control, damos clic sobre el icono de Región y se abrirá el panel de control, dentro del panel buscamos el icono de Región y damos clic sobre él para abrir las opciones de configuración regional. Dentro de las opciones de Región hacemos clic en Formato para poder visualizar los diferentes idiomas en una lista, dentro de la lista buscaremos la opción Español (España) y la seleccionamos.

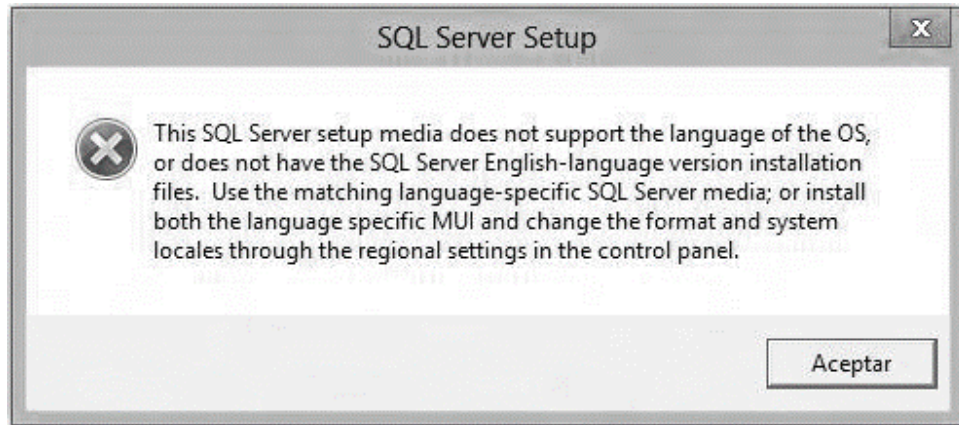


Figura 4.2.2 Error instalación SQL Server 2012.

Al finalizar la instalación, si todo ha salido correcto, nos aparecerá una ventana “Operación completada”, indicándonos que la instalación de SQL Server 2012 se completó de manera satisfactoria. Ver figura 4.2.3

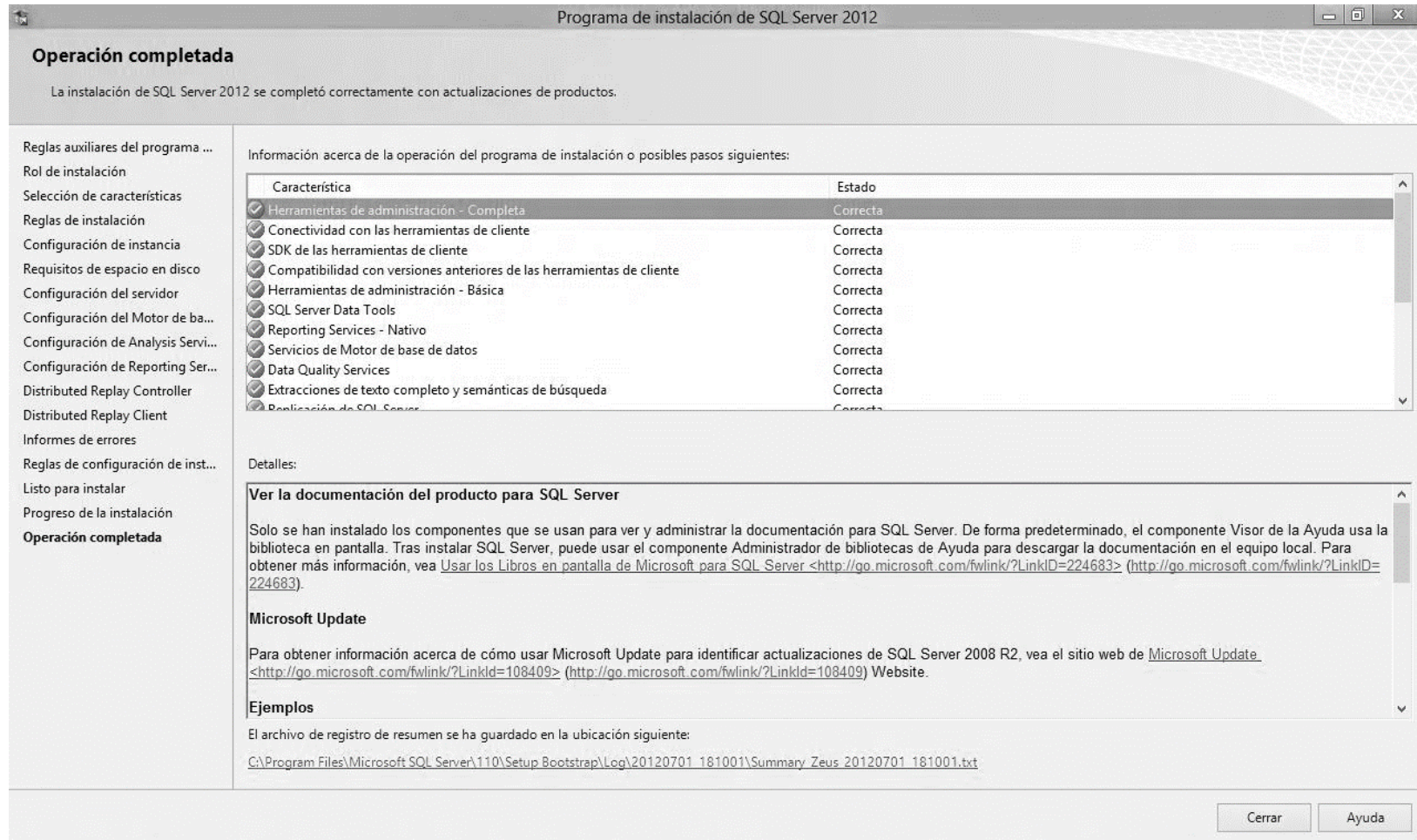


Figura 4.2.3 Instalación completa.

Creación de la base de datos.

Primero abrimos SQL Management Studio, al momento de conectar al servidor, debemos de autenticarnos dependiendo de la configuración que hayamos realizado en la instalación. Véase figura 4.2.4.



Figura 4.2.4 Inicio a SQL Management Studio mediante autenticación de Windows.

Para realizar la base de datos, se puede hacer de diferentes formas, usando la interfaz de usuario o mediante comandos SQL.

Para llevar a cabo la creación con comandos SQL debemos de pulsar la opción “Nueva Consulta” la cual desplegara una sección donde se podrán introducir los comandos. La consulta SQL lleva la siguiente estructura.

```

USE MASTER
GO
IF EXISTS(SELECT NAME FROM SYS.DATABASES WHERE NAME =
'gestion_documental')
BEGIN
DROP DATABASE gestion_documental
END
CREATE DATABASE gestion_documental
GO
PRINT 'La base de datos: gestion_documental ha sido creada con exito.'
GO
USE gestion_documental
GO

```

Con este script nos aseguraremos primero que no haya otra base de datos con el mismo nombre y si existe una, esta será borrada para después crear una nueva con el nombre de `gestion_documental` tal y como se muestra en la figura 4.2.5.



```

SQLQuery1.sql - AL...aro-PC\Alvaro (53)* X
USE MASTER
GO
IF EXISTS(SELECT NAME FROM SYS.DATABASES WHERE NAME = 'gestion_documental')
BEGIN
DROP DATABASE gestion_documental
END
CREATE DATABASE gestion_documental
GO
PRINT 'La base de datos: gestion_documental ha sido creada con exito.'
GO
USE gestion_documental
GO

```

Mensajes

La base de datos: gestion_documental ha sido creada con exito.

Consulta ejecutada correctamente. ALVARO-PC\SQLSERVER (11)

Figura 4.2.5 Creación de la base de datos `gestion_documental` con comandos SQL.

Una segunda forma sería con la interfaz de usuario, desde la cual seleccionamos el apartado Bases de datos, le damos clic derecho con lo que desplegaremos todas las opciones disponibles dentro veremos la opción “Nueva Base de datos”

tal y como se puede observar en la figura 4.2.6. Escogiendo esta opción nos abrirá una nueva ventana desde la cual podremos asignar el nombre a la base de datos, tipo de archivo que tendrá, el grupo al que debe pertenecer y por último el tamaño inicial de dicho archivo, después de asignar los respectivos valores a cada campo solo se deberá de dar clic al botón de Aceptar para terminar la creación de la base de datos tal y como se puede apreciar en la figura 4.2.7.

Una vez creada la base de datos se puede ver en el árbol de objetos independientemente del método que se haya usado para su creación.

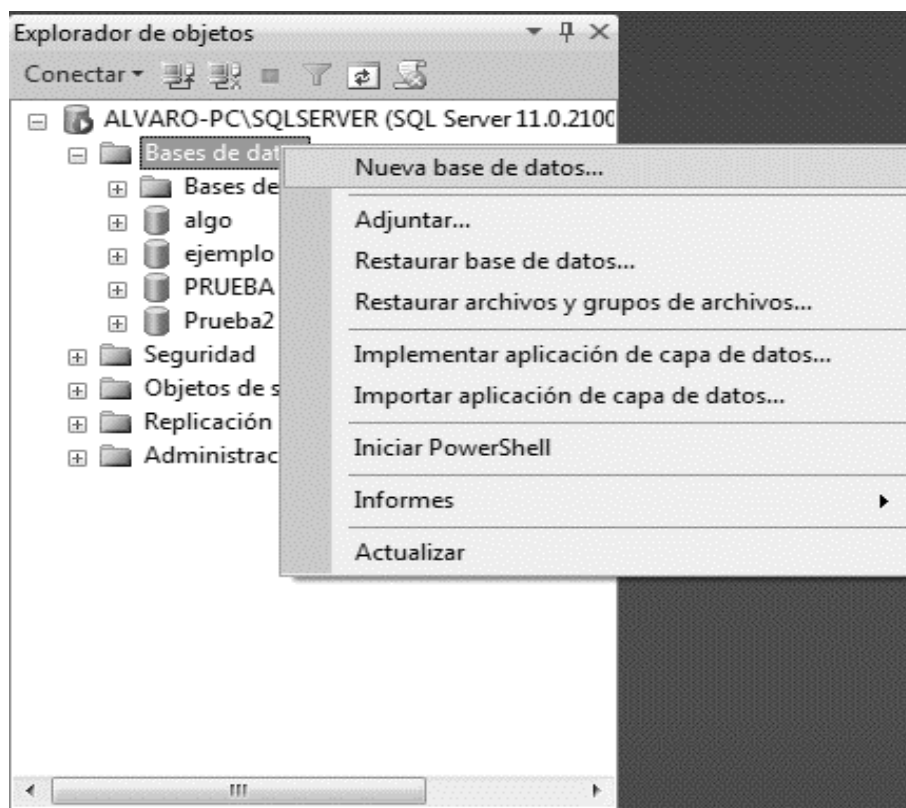


Figura 4.2.6 Creación de la base de datos desde la interfaz de usuario.

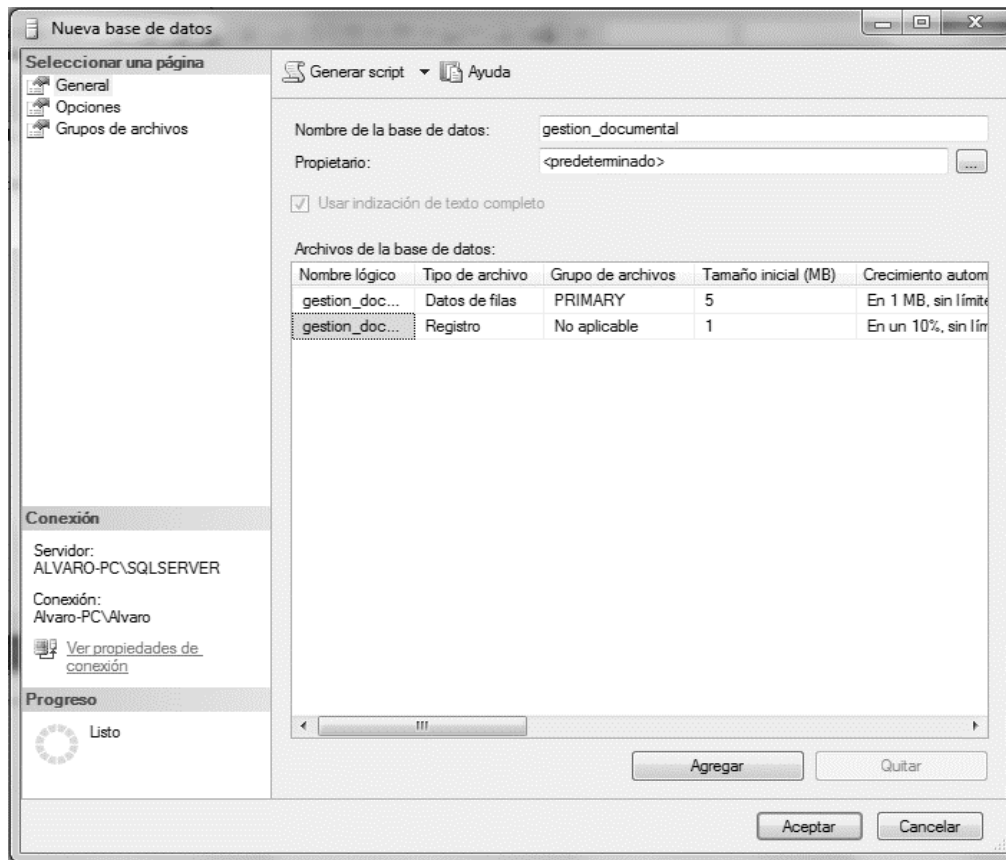


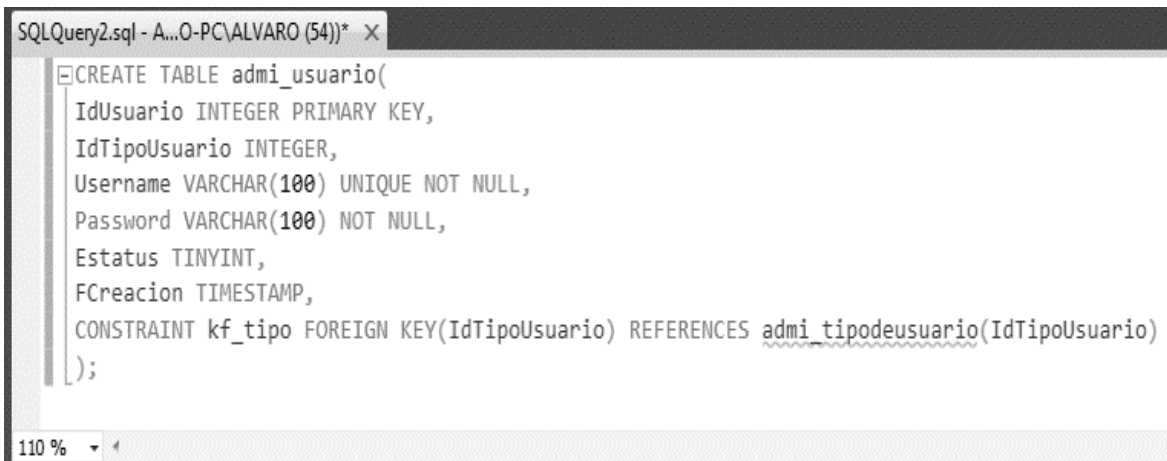
Figura 4.2.7 Asignación de valores a los campos de la base de datos.

Creación de las tablas del Sistema de Gestión Documental.

Una tabla se puede crear de dos diferentes formas, mediante el asistente y usando código SQL.

La sintaxis general para la creación de una tabla mediante SQL es la siguiente.

```
CREATE TABLE "nombre de la tabla"(  
nombre_variable tipo_de_dato [(longitud) ],  
nombre_variable2 tipo_de_dato [(longitud) ],  
);
```



```
SQLQuery2.sql - A...O-PC\ALVARO (54)* x
CREATE TABLE admi_usuario(
  IdUsuario INTEGER PRIMARY KEY,
  IdTipoUsuario INTEGER,
  Username VARCHAR(100) UNIQUE NOT NULL,
  Password VARCHAR(100) NOT NULL,
  Estatus TINYINT,
  FCreacion TIMESTAMP,
  CONSTRAINT kf_tipo FOREIGN KEY(IdTipoUsuario) REFERENCES admi_tipodeusuario(IdTipoUsuario)
);
```

110 %

Figura 4.2.8 Creación de una Tabla mediante SQL.

Para llevar a cabo la tabla desde el asistente debemos de seleccionar del árbol de bases de datos que acabamos de crear, con clic izquierdo se deberá de desplegar todo los apartados de los diferentes objetos de nuestra base de datos, dentro de estos debemos de buscar el apartado de Tablas y darle clic derecho para poder observar la opción “Nueva Tabla” como se puede observar en la figura 4.2.9. Dentro de esta opción podremos anexar el nombre de cada campo, su tipo de dato como longitud del mismo además de los diferentes constraints que deba llevar, esto se puede observar en la figura 4.2.10.

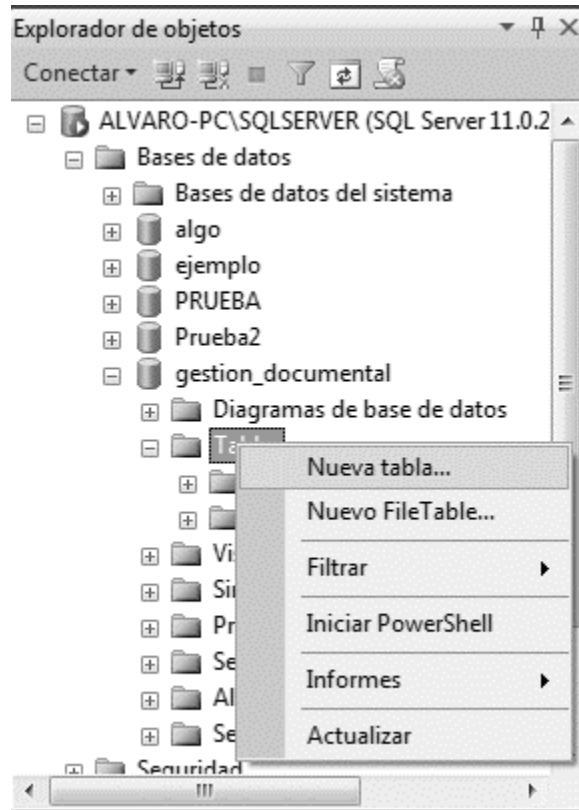


Figura 4.2.9 Opción para la creación de una tabla desde el asistente.

A screenshot of the 'Nueva tabla' wizard in SQL Server Enterprise Manager. The wizard is titled 'ALVARO-PC\SQLEXP...eba - dbo.Table_1*' and shows a table with the following columns:

| Nombre de columna | Tipo de datos | Permitir val... |
|---------------------|---------------|--------------------------|
| IdArchivoCarpeta | int | <input type="checkbox"/> |
| IdPadre | int | <input type="checkbox"/> |
| IdExpediente | int | <input type="checkbox"/> |
| IdUsuario | int | <input type="checkbox"/> |
| Nombre | varchar(100) | <input type="checkbox"/> |
| urlDeArchivoCarpeta | varchar(100) | <input type="checkbox"/> |
| FCreacion | timestamp | <input type="checkbox"/> |

Figura 4.2.10 Creación de la tabla desde el asistente.

En la figura 4.2.11 podemos observar todas las tablas que se necesitó crear para poder cumplir con los requisitos del Sistema de Administración Documental.

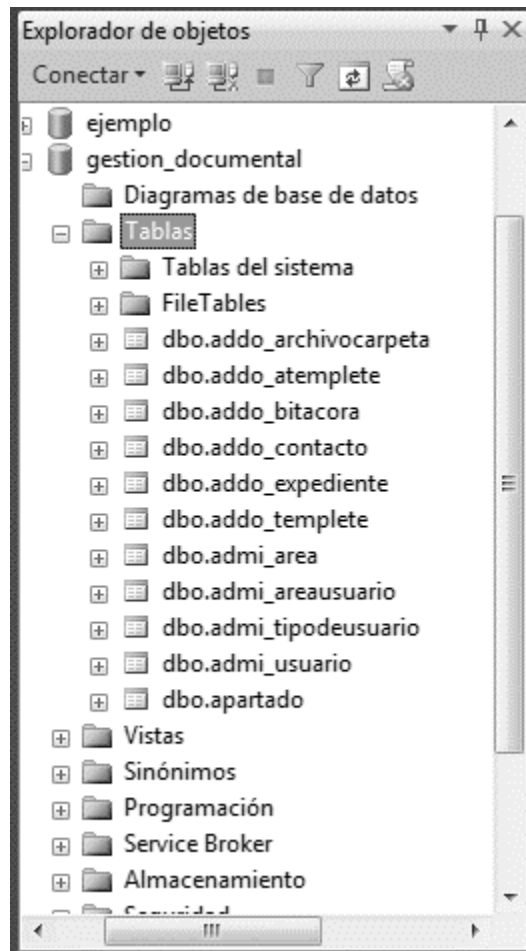


Figura 4.2.11 Tablas de la base de datos gestion_documental

Creación de consultas

Una consulta es el método para acceder a los datos almacenadas en la base de datos. Con las consultas se puede agregar, mostrar, borrar o modificar los datos.

Sintaxis de SELECT

```
SELECT [ Column_Name ] ]  
FROM [ FORCE ] [ DatabaseName ] Table [ [AS] Local_Alias]  
[WHERE Condition [AND Condition ...]  
[GROUP BY GroupColumn [, GroupColumn ...]]  
[HAVING FilterCondition] [UNION [ALL] SELECTCommand]  
[ORDER BY Order_Item [ASC | DESC] [, Order_Item [ASC | DESC] ...]]
```


En la figura 4.2.12 podemos ver todos los campos insertados en la tabla `admi_usuario` mediante código SQL y en la figura 4.2.12 lo mismo pero haciendo uso del asistente.

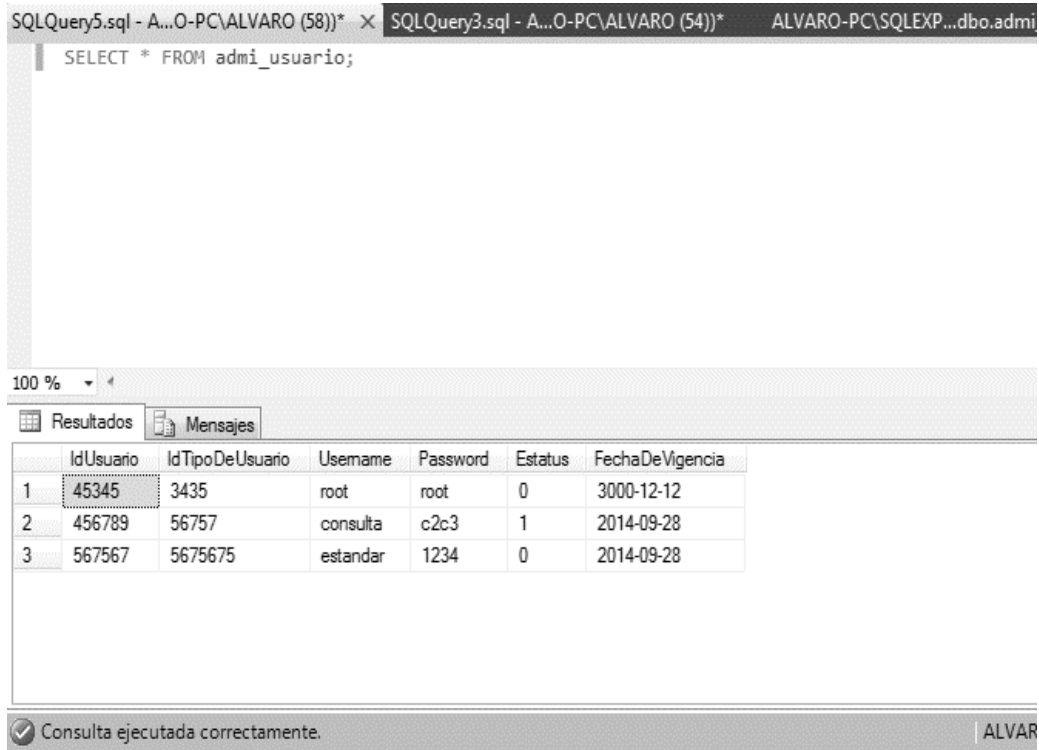


Figura 4.2.11 Consulta Select de la table `admi_usuario`.

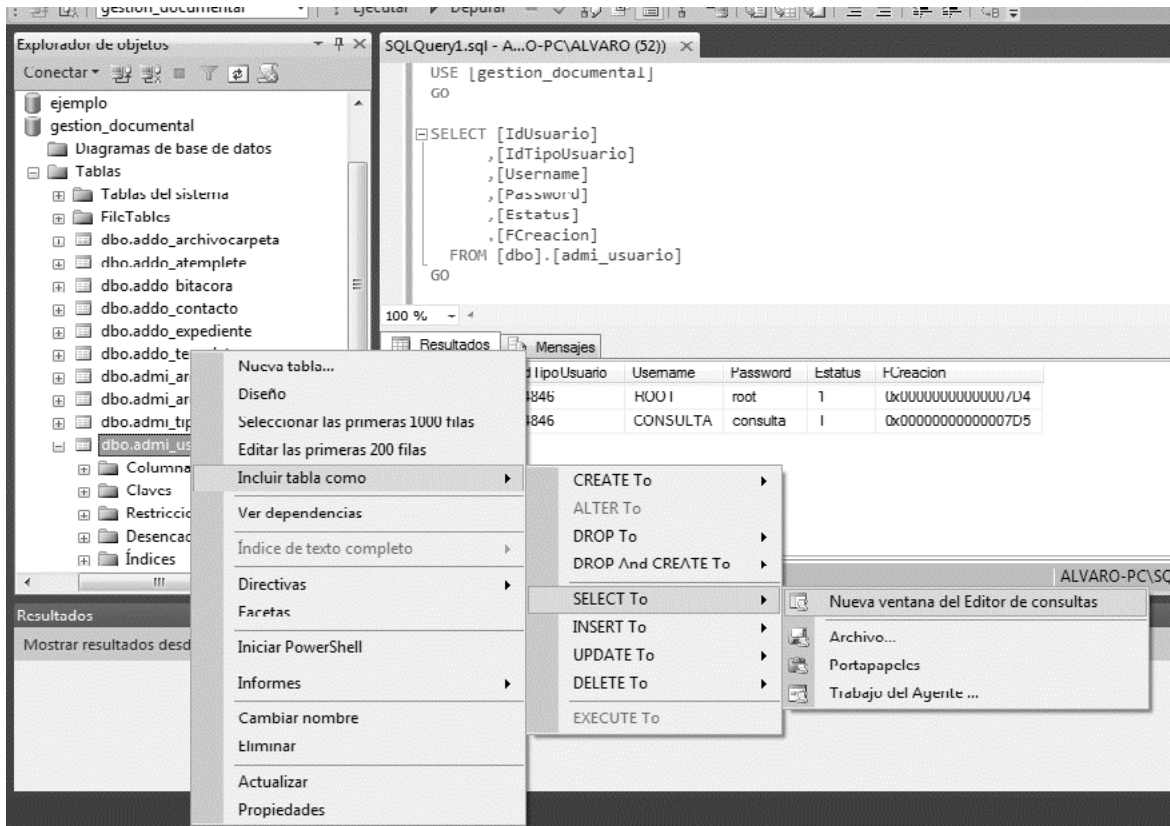


Figura 4.2.12 Consulta desde el asistente de SQL Server.

Creación de procedimientos almacenados.

Un procedimiento almacenado es un objeto que pertenece a la base de datos, contiene un conjunto de instrucciones SQL, tanto para consulta como manipulación de datos, como de control de secuencia.

La sintaxis de dicha instrucción es básicamente la siguiente:

```
CREATE PROCEDURE nombre_del_procedimiento
[Lista de parametros]
AS
(Sentencia SQL)
[Return [valor]]
```

Sintaxis de unos de nuestros procedimientos almacenados.

```
CREATE PROCEDURE user_area
@Area VARCHAR(43)
AS
SELECT c.Nombre, c.AppMaterno, c.AppMaterno, a.Nombre
FROM admi_usuario u JOIN admi_areausuario ua
ON u.IdUsuario=ua.IdUsuario
JOIN admi_area a
ON a.IdArea=ua.IdArea
JOIN addo_contacto c
ON u.IdUsuario=c.IdUsuario
WHERE a.Nombre=@Area;
```

En la figura 4.2.13 podemos observar la creación de un procedimiento almacenado sencillo, el cual no devuelve el nombre completo de los usuarios que se encuentran en una determinada área, este proceso tiene como entrada el nombre del área. En la figura 4.2.14 podemos observar la lista de procedimientos almacenados que están pensados hasta este punto para la base de datos.

The screenshot shows a SQL query editor window with the following content:

```
SQLQuery6.sql - A...O-PC\ALVARO (59))* x SQLQuery5.sql - A...O-PC\ALVARO (58))* SQLQuery3.sql -
CREATE PROCEDURE user_area
  @Area VARCHAR(43)
AS
SELECT c.Nombre, c.AppMaterno, c.AppMaterno, a.Nombre
FROM admi_usuario u JOIN admi_areausuario ua
ON u.IdUsuario=ua.IdUsuario
JOIN admi_area a
ON a.IdArea=ua.IdArea
JOIN addo_contacto c
ON u.IdUsuario=c.IdUsuario
WHERE a.Nombre=@Area;
```

At the bottom of the window, there is a status bar that reads: "Comandos completados correctamente." (Commands completed successfully.)

Figura 4.2.13 Creación del procedimiento almacenado user_area.

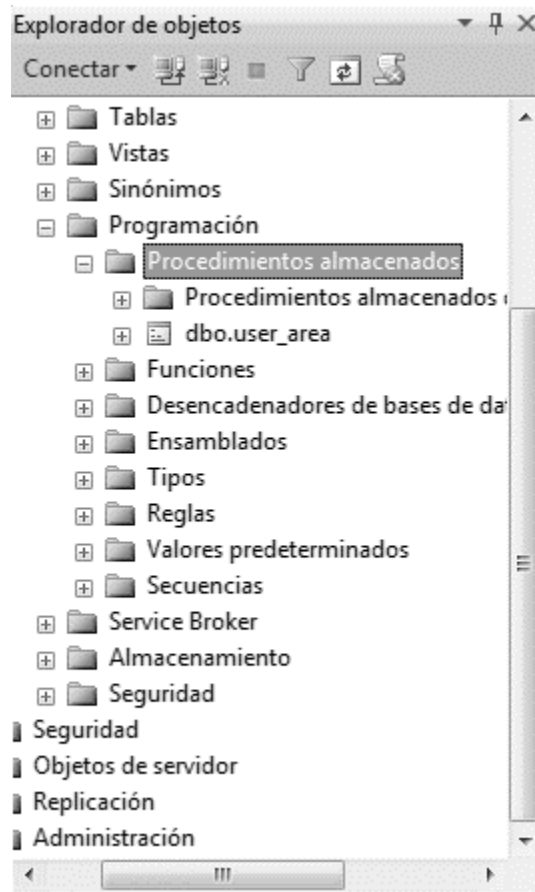


Figura 4.2.14 Lista de procedimientos almacenados de la base de datos.

Creación de triggers

Un trigger es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce el evento, pueden incluir cualquier número y clase de instrucción SQL. Desde el trigger se puede obtener todos los datos de una fila que se haya modificado utilizando inserciones o borrado pero también tiene limitaciones ya que solo se puede aplicar a una tabla específica, no sirve para dos tablas o mas aunque también tiene ventajas como por ejemplo aunque se haya creado en una base de datos puede hacer referencia a otras bases de datos.

Sintaxis:

```
CREATE TRIGGER [Esquema.] nombre  
ON {Tabla | Vista}  
{FOR | AFTER | INSTEAD OF}  
{ [INSERT] [,] [UPDATE] [,] [DELETE] }  
AS  
Sentencias SQL
```

Ejemplo de uno de nuestros triggers empleados en la base de datos.

```
CREATE TRIGGER nuevo_usuario  
ON adm_usuario  
FOR INSERT  
AS  
DECLARE @ID INTEGER  
SELECT @ID = (SELECT IdUsuario FROM adm_usuario  
PRINT ,Nuevo usuario"
```

En la figura 4.2.13 podemos observar la creación de un trigger el cual se encargara de mostrar el username de todos los usuarios que sean creados de la tabla adm_usuario y mandar un mensaje que diga "Nuevo Usuario".

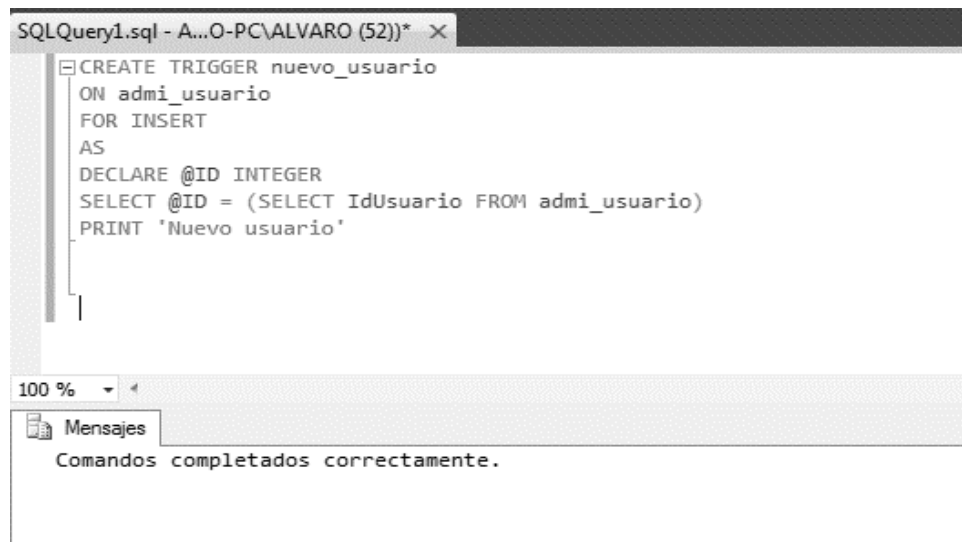


Figura 4.2.13 Creación del trigger nuevo_usuario.

4.3 Creación del front-end con la funcionalidad del órgano rector.

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

La interfaz gráfica de usuario, conocida también como GUI (del inglés graphical user interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Si una interfaz no es lo suficientemente intuitiva el usuario no explotará todas las funcionalidades del software y además pensará que la herramienta no le permite lograr su trabajo, ni mucho menos alcanzar sus objetivos.

Algunas de las características que debe tener una buena interfaz gráfica son las siguientes:

- Facilidad de comprensión, aprendizaje y uso
- El objeto de interés ha de ser de fácil identificación
- Diseño ergonómico mediante el establecimiento de menús, barras de acciones e iconos de fácil acceso.
- Las interacciones se basarán en acciones físicas sobre elementos de código visual o auditivo (iconos, botones, imágenes, mensajes de texto o sonoros, barras de desplazamiento y navegación...) y en selecciones de tipo menú con sintaxis y órdenes
- Las operaciones serán rápidas, incrementales y reversibles, con efectos inmediatos
- Existencia de herramientas de ayuda y consulta

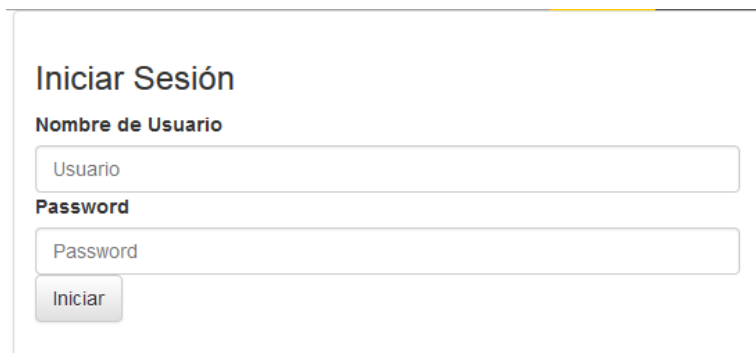
- Tratamiento del error bien cuidado y adecuado al nivel de usuario

A continuación se dará una explicación de algunas de las pantallas que conforman la interfaz de usuario del sistema planteado como solución para la problemática detectada en el Sistema de Administración Documental.

El sistema cuenta principalmente con 5 módulos, y son los siguientes:

- Plantillas.
- Expedientes.
- Bitácora.
- Búsqueda.
- Configuración.

Para tener acceso a los menús anteriores, el usuario debe autenticarse, la pantalla de autenticación contiene un formulario típico de inicio de sesión, el cual está conformado por los siguientes elementos básicos: (Ver figura 4.3.1)



The image shows a login form titled "Iniciar Sesión". It contains two input fields: "Nombre de Usuario" with the placeholder text "Usuario" and "Password" with the placeholder text "Password". Below the fields is a button labeled "Iniciar".

Figura 4.3.1 Pantalla de autenticación.

Una vez que se inserta el usuario y contraseña los datos son validados y de ser correctos tendrá acceso al sistema y se le mostrará la pantalla principal. (Ver figura 4.3.2).



Sistema de Administración Documental.

Figura 4.3.2 Pantalla principal.

En la pantalla principal se encuentran todas las opciones que existen en cada módulo. Se decidió no usar muchos gráficos en la pantalla, debido a que estos ocupan demasiado espacio y pueden tardar bastante tiempo en cargarse y en ocasiones se exige una memoria gráfica que la soporte, y si el usuario está trabajando con una conexión de internet lenta y algunos equipos no son nuevos, esto lo hará más deficiente.

En la gestión de los expedientes el usuario tiene tres opciones, alta, administrar y consultar los expedientes existentes. (Ver figura 4.3.3)



Figura 4.3.3 Gestión de expedientes.

Inicio Plantillas Expedientes Bitácora Búsqueda Configuración Salir

Crear Expediente

Clave de Expediente:

Área

Tipo De Plantilla

Figura 4.3.4 Alta de expedientes.

En la pantalla mostrada en la figura 4.3.4, el usuario podrá dar de alta expedientes el usuario debe ingresar los siguientes parámetros:

- Clave de Expediente. Es un identificador único con el cual podrá ser identificado fácilmente el expediente
- Área. Es una referencia del expediente la cual señala la unidad administrativa dueña de la información.
- Tipo de Plantilla. Permite crear una estructura de carpetas tipo árbol dentro del expediente.

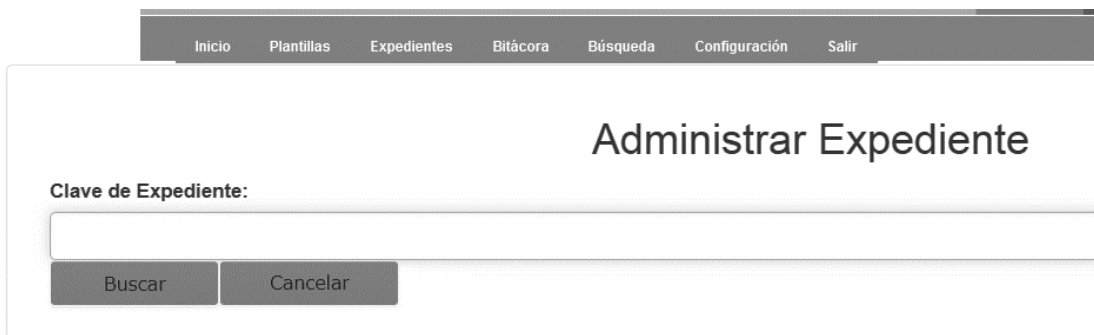


Figura 4.3.5 Administración de expedientes.

En la pantalla mostrada en la figura 4.3.5, el usuario podrá administrar todos sus expedientes dados de alta. Para ello se requiere insertar la clave del expediente y presionar el botón de búsqueda.

En la pantalla mostrada en la figura 4.3.6 el usuario podrá visualizar su expediente asociado con la plantilla y podrá ejecutar los comandos siguientes:

- Renombrar. Este comando permite renombrar únicamente los expedientes creados.
- Agregar Archivo. Permite la carga de archivos digitalizados, pueden ser cualquier tipo de archivo. El usuario previamente debe indicar con un clic el apartado donde cargará el archivo correspondiente.

- Eliminar Archivo. Cuando el usuario requiera eliminar algún documento, únicamente deberá señalarlo con un clic y posteriormente ejecutar el comando correspondiente.
- Actualizar. Esta función permite al usuario actualizar la información de la pantalla después de haber agregado y/o eliminado información.
- Vista Previa. Permite al usuario visualizar la jerarquía del expediente, incluyendo los archivos contenidos. Además por requerimiento del usuario, le permite imprimir dicha pantalla.



Figura 4.3.6 Administración de expedientes.

En la pantalla mostrada en la figura 4.3.7 el usuario podrá realizar una consulta de todos los expedientes que existentes. Podrá ejecutar dos comandos:

- Mostrar Expediente. El sistema únicamente le mostrará al usuario el expediente, los apartados y los archivos que contienen.
- Administrar Expediente. El sistema le mostrará al usuario la pantalla 4.3.6.

En la gestión de los apartados el usuario tiene dos opciones: alta o consultas las plantillas de los apartados existentes. (Ver figura 4.3.8)



Figura 4.3.8 Gestión de las plantillas de apartados.

Inicio Plantillas Expedientes Bitácora Búsqueda Configuración Salir

Expedientes

| Clave de Expediente | Area | Mostrar | Mostrar |
|----------------------|---------------|--------------------|------------------------|
| <input type="text"/> | Select Filter | | |
| Expediente A | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Adquisiciones | Mostrar Expediente | Administrar Expediente |

Cancelar

Figura 4.3.7 Consulta de expedientes.

En la pantalla mostrada en la figura 4.3.9 el usuario podrá realizar una plantilla de apartados la cual tendrá una estructura tipo árbol, para la creación de la misma solo es necesario ejecutar el comando del botón Crear Plantilla y el usuario deberá ingresar el nombre de la misma. (Ver figura 4.3.10)

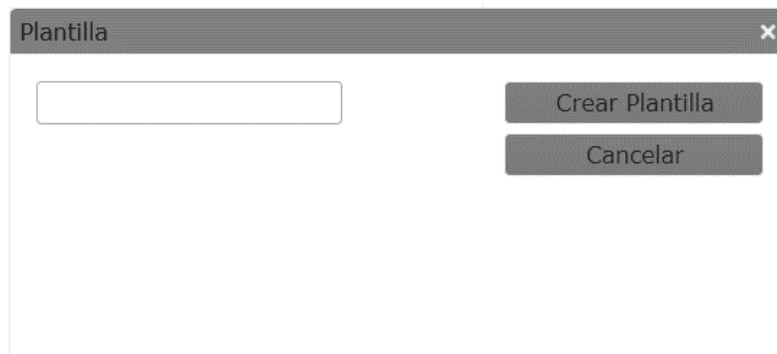


Figura 4.3.10 Nombre de la plantilla.



Figura 4.3.9 Alta de plantillas de apartados.

Una vez que el usuario ingresó el nombre de la plantilla podrá crear los apartados que requiera, o incluso podrá eliminarlos, renombrarlos o cancelar la creación de la plantilla. (Ver figura 4.3.11).



Figura 4.3.11 Crear plantilla.

En la pantalla mostrada en la figura 4.3.12, el usuario únicamente podrá consultar todas las plantillas y su respectiva estructura tipo árbol sobre todos sus apartados creados. (Ver figura 4.3.12).



Figura 4.3.12 Consulta de plantillas.

En el módulo de la bitácora el usuario podrá visualizar absolutamente todos los cambios que se han realizado en el sistema. La información no puede ser borrada ni siquiera por los administradores; ver figura 4.3.13.

| Usuario | Tipo De Operación | Fecha de Registro | Hora |
|---------------|---|-------------------|------------|
| Select Filter | | | |
| UNAM | Consulta Bitacora | 10-11-2014 | 15:30:23.0 |
| UNAM | Consulta Plantillas | 10-11-2014 | 15:26:21.0 |
| UNAM | Ingreso a Crear Plantillas | 10-11-2014 | 15:17:00.0 |
| UNAM | Consulta Plantillas | 10-11-2014 | 15:14:23.0 |
| UNAM | Consulta Expediente: Expediente A | 10-11-2014 | 15:10:46.0 |
| UNAM | Consulta Expediente: Expediente A | 10-11-2014 | 15:09:35.0 |
| UNAM | Ingreso a la edicion del Expediente: Expediente A | 10-11-2014 | 15:09:21.0 |
| UNAM | Ingreso a la edicion del Expediente: EXP2014-A1 | 10-11-2014 | 15:05:23.0 |
| UNAM | Consulta Expediente: EXP2014-A1 | 10-11-2014 | 15:05:15.0 |
| UNAM | Consulta Expediente: Expediente A | 10-11-2014 | 15:05:03.0 |
| UNAM | Consulta Expedientes | 10-11-2014 | 15:04:57.0 |

Cancelar

Figura 4.3.13 Bitácora

En el módulo de búsqueda el usuario podrá realizar búsquedas de archivos, expedientes y apartados, para ello solo requiere ingresar un parámetro en el formulario y posteriormente ejecutar el comando del botón buscar. Ver figura 4.3.14.

| Inicio | Plantillas | Expedientes | Bitácora | Búsqueda | Configuración | Salir |
|--------|------------|-------------|----------|----------|---------------|-------|
|--------|------------|-------------|----------|----------|---------------|-------|

Busque Archivos, Apartados y Expedientes

Figura 4.3.14 Búsquedas

El sistema mostrará los resultados encontrados organizados por expedientes, apartados y archivos (ver figura 4.3.15) y además se podrán modificar los mismos según corresponda.

Inicio Plantillas Expedientes Bitácora Búsqueda Configuración Salir

Expedientes

| Clave de Expediente | Area | | |
|----------------------|---------------|--------------------|------------------------|
| <input type="text"/> | Select Filter | | |
| Expediente A | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Adquisiciones | Mostrar Expediente | Administrar Expediente |

Apartados

| Expediente | Nombre Apartado | Area | | |
|----------------------|----------------------|---------------|--------------------|------------------------|
| <input type="text"/> | <input type="text"/> | Select Filter | | |
| EXP2014-A1 | Apartado 1 | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Apartado 3.1 | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Apartado 2 | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Apartado 3 | Adquisiciones | Mostrar Expediente | Administrar Expediente |
| EXP2014-A1 | Apartado 2.1 | Adquisiciones | Mostrar Expediente | Administrar Expediente |

Archivos

| Nombre de Archivo | Tipo de Archivo | Fecha de Creación | |
|--|-----------------|----------------------|-----------------|
| <input type="text"/> | Select Filter | <input type="text"/> | |
| C:\fakepath\LOGO_GDF.jpg | .jpg | 10/29/2014 | Mostrar Archivo |
| C:\fakepath\LOGO_GDF - (30-10-2014 19:35:8).jpg | .jpg | 10/29/2014 | Mostrar Archivo |
| C:\fakepath\Sesion inicial pol+ética y MAAGTICsi 2014 D+ a 1 V.2 ... | .pdf | 10/29/2014 | Mostrar Archivo |
| logoUNAM.jpg | .jpg | 11/10/2014 | Mostrar Archivo |

Figura 4.3.15 Resultados búsquedas.

4.4 Generación de pruebas, mantenimiento y reportes

4.4.1 Tipos de pruebas.

Las pruebas del software son un elemento crítico para la garantía de la calidad de cualquier proyecto de sistemas y representan una revisión final de las especificaciones, del diseño y de la codificación. No es raro que el coste suponga un 40% del coste total de desarrollo del proyecto.

Interesa señalar que en cada fase del ciclo de vida de desarrollo del software se plantea un conjunto de pruebas que permiten constatar que el software desarrollado satisface las especificaciones de esa fase. Dado lo anterior las pruebas se clasifican de distintas maneras siendo las más comunes las siguientes.

Pruebas de caja blanca.

Las pruebas de caja blanca evalúan el código de cada uno de los módulos de un sistema, considerando los detalles procedurales. Se plantean distintos caminos de ejecución alternativos y se llevan a cabo para observar los resultados y contrastarlos con lo esperado. En principio se podría pensar que es variable la verificación mediante la prueba de caja blanca de la totalidad de caminos de un procedimiento dado que es imposible en la mayoría de los casos reales debido a la exponencialidad en el número de combinaciones posibles; esto es debido a que las pruebas de caja blanca, también llamadas pruebas estructurales, utilizan el código fuente del programa, y especialmente su estructura de control, para de ahí seleccionar los casos de prueba.

Pruebas de caja negra.

Las pruebas de caja negra se derivan a partir de la especificación del sistema; es decir, el sistema se trata como una caja negra cuyo comportamiento solo puede ser determinado estudiando sus entradas y sus salidas relacionadas. El probador presenta las entradas al componente o al sistema y examina las correspondientes salidas. Si las

salidas no son las esperadas entonces la prueba ha detectado un problema con el software.

Pruebas de integración.

Las pruebas de integración se realizan con el fin de detectar posibles errores al momento de integrar los diferentes componentes de un sistema y empezar a interactuar entre ellos.

Dentro de las pruebas de Integración, existen dos enfoques: descendente (top-down) y ascendente (botton-down). En la primera se comienza la prueba por el componente principal, sustituyendo por resguardos los componentes que son llamados por éste. Sobre dicho conjunto se realiza una primera batería de pruebas. A continuación, se integran los componentes subordinados al principal y se continúan realizando pruebas sucesivamente hasta completar la integración del software completo.

En la integración ascendente las pruebas se hacen en sentido inverso, comenzando por los componentes que no necesitan llamar a otro y finalizando por el componente principal.

Pruebas de regresión.

Las pruebas de regresión corresponden a las que se realizan siempre que existe una modificación sobre el software ya probado, sea porque se corrige un defecto encontrado, porque se implementa una mejora o una adaptación que pueda introducir nuevos defectos. Estas pruebas consisten en volver a ejecutar los casos de prueba que fueron diseñados y ejecutados anteriormente sobre el código modificado.

Pruebas alfa.

Las pruebas alfa son las que se destinan solo para circulación interna de la organización y se pasan a un selecto grupo de usuarios muy críticos. El objetivo de estas pruebas es determinar el comportamiento del producto en las diferentes

aplicaciones o módulos del sistema. Después de refinar el producto se procede con las pruebas beta. Está en un ambiente controlado y supervisado por la gente de desarrollo.

Pruebas beta.

Las pruebas Beta se envían a un público más amplio, pero aun selecto, crítico y de ser posible ajeno al proyecto. La idea de este tipo de pruebas es obtener retroalimentación acerca de las experiencias con el producto a liberar. Este tipo de pruebas se realiza con un grupo heterogéneo de personas con el fin de obtener opiniones con diferentes puntos de vista.

Pruebas de estrés.

Las pruebas de estrés sobre un sistema consisten en realizar demandas al sistema que están fuera de los límites del diseño. Las pruebas de estrés se realizan acercándose a la máxima carga del diseño del sistema hasta que el sistema falla. Este tipo de pruebas tiene dos funciones; determinar que el fallo en el sistema al estresarlo no provoque corrupción de datos o pérdidas inesperadas de servicios, y que en un nivel de estrés el sistema no manifieste defectos que normalmente no serían descubiertos. Este tipo de pruebas son particularmente relevantes para los sistemas distribuidos ya que a menudo sufren una degradación cuando son sobrecargados.

Pruebas de validación.

Las pruebas de validación intentan demostrar que el software es el que el cliente quiere y que satisface sus requerimientos siguiendo el “plan y procedimiento de pruebas” que se generó en la etapa de análisis y diseño del proyecto.

Pruebas funcionales.

Las pruebas funcionales son las que realiza en desarrollador para comprobar que el módulo o sección que desarrolló funciona correctamente y previo a las pruebas de integración y/o de usuario. En el caso de encontrar errores en esta fase es un hecho que deben corregirse para evitar incurrir en costos innecesarios cuando este módulo o sección interactúe con el resto.

Pruebas de usuario.

Las pruebas de usuario se realizan en un ambiente previo a Producción, ya con todo el sistema terminado y habiendo ya conjuntado todas las funcionalidades y módulos que lo componen. Estas pruebas se realizan con el usuario que finalmente utilizará en el día a día el sistema. En esta fase de las pruebas el usuario final debe probar el sistema como si éste ya se encontrase instalado y liberado en el ambiente de producción. Muy probablemente el volumen de insumos sea menor dado que en ambientes previos el espacio asignado a es mucho menor al de producción, pero debe ser aceptable para dar un veredicto al final de esta fase.

4.4.2 Casos de pruebas funcionales en el SAD para el alta de un usuario

En esta sección se aplica el concepto de pruebas funcionales sobre el módulo de alta de un usuario en el SAD. Para esto se tendrán dos casos de prueba a considerar: caso 1 la creación de un usuario; probar que el usuario habiéndose creado no cuenta con los privilegios para firmarse al sistema y, caso 2 cuando se activa el usuario y es entonces cuando ya accesa el sistema.

Caso 1: El usuario se da de alta en el sistema SAD. Para esto se elige en el SAD las opciones Configuración/Usuarios/Agregar Nuevo usuario y se especifica Username = XM01AFG, Password = CLATITO9012 y tipo de usuario Consulta(C) como se muestra en la figura 4.4.2.1 Alta de un usuario para consulta en el SAD.

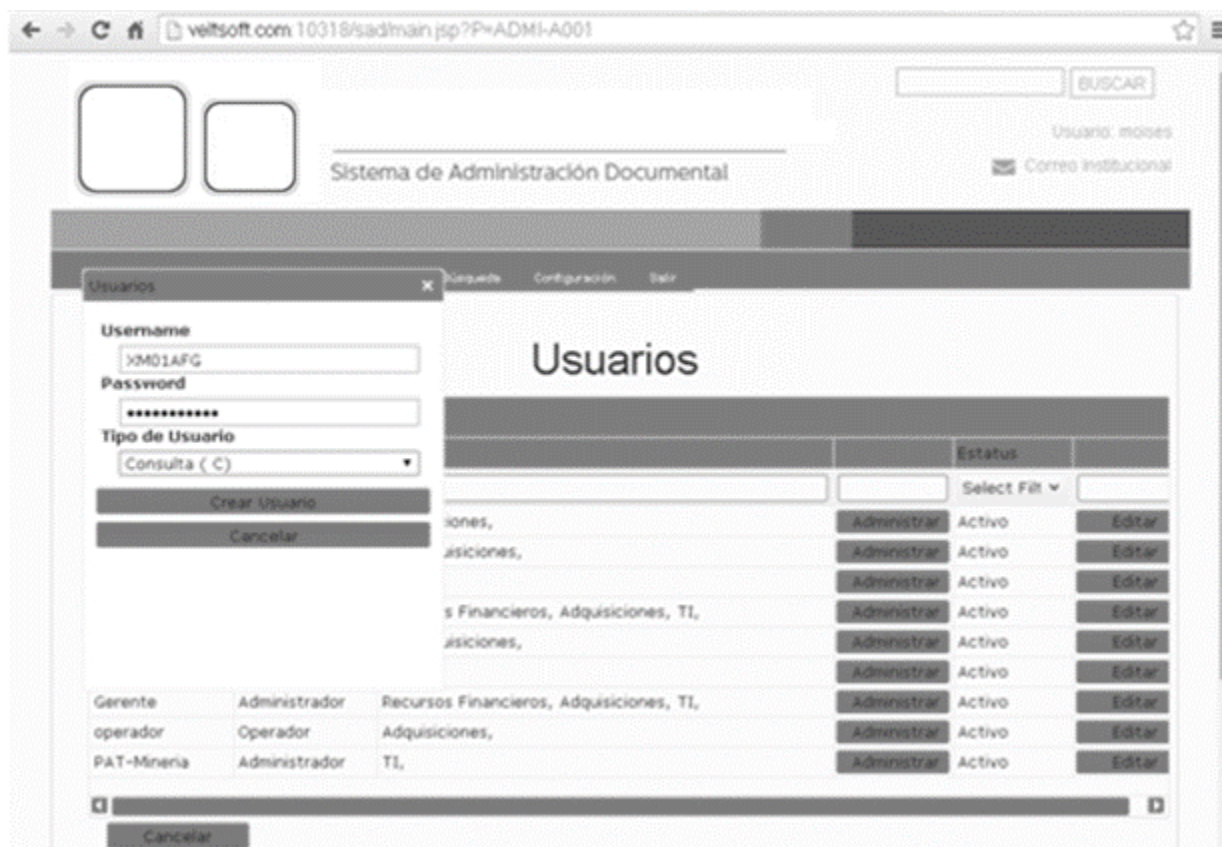


Figura 4.4.2.1 Alta de un usuario para consulta en el SAD.

Y se da crear usuario. Posterior a esta acción se visualiza que el usuario ha sido creado con estatus Inactivo según se muestra en la figura 4.4.2.2 Creación de un usuario con estatus inactivo.

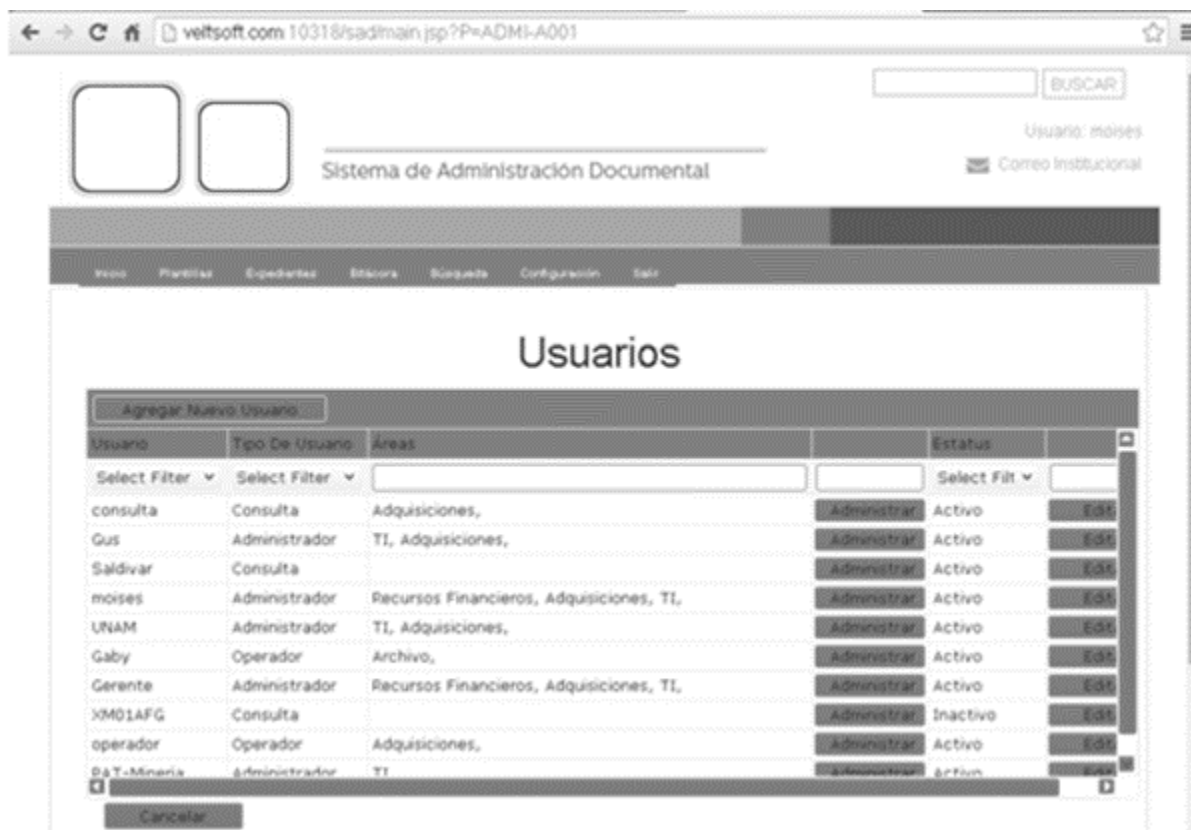


Figura 4.4.2.2 Creación de un usuario con estatus Inactivo.

Para este caso, se intentará entrar al sistema SAD con el usuario XM01AFG recién creado y el sistema debe arrojar un error y no permitir el acceso al sistema lo cual se comprueba en la figura 4.4.2.3 Error con usuario inactivo en el SAD.

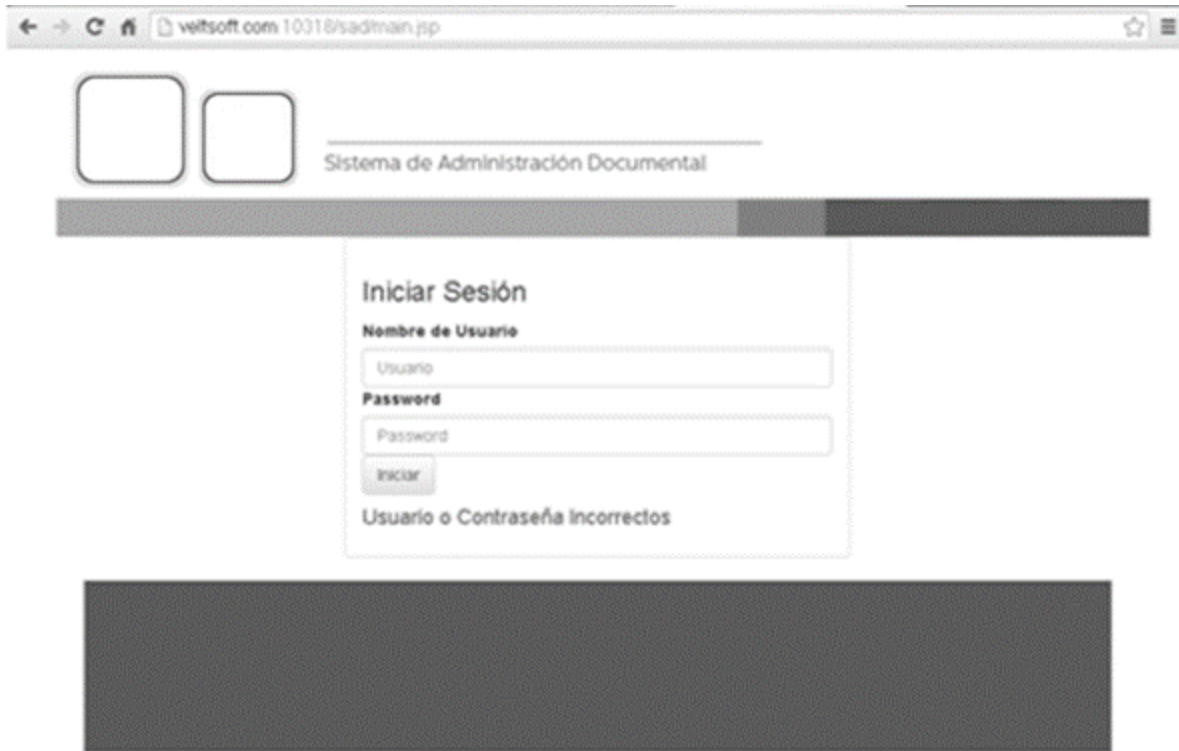


Figura 4.4.2.3 Error con usuario inactivo en el SAD.

Caso 2: Se ingresa al SAD siguiendo las opciones Configuración/Usuarios/Administrar/Cambiar Estatus de Usuario. En este momento el sistema pregunta si se desea activar el usuario como se indica en la figura 4.4.2.4 Confirmación de activación de usuario en el SAD.

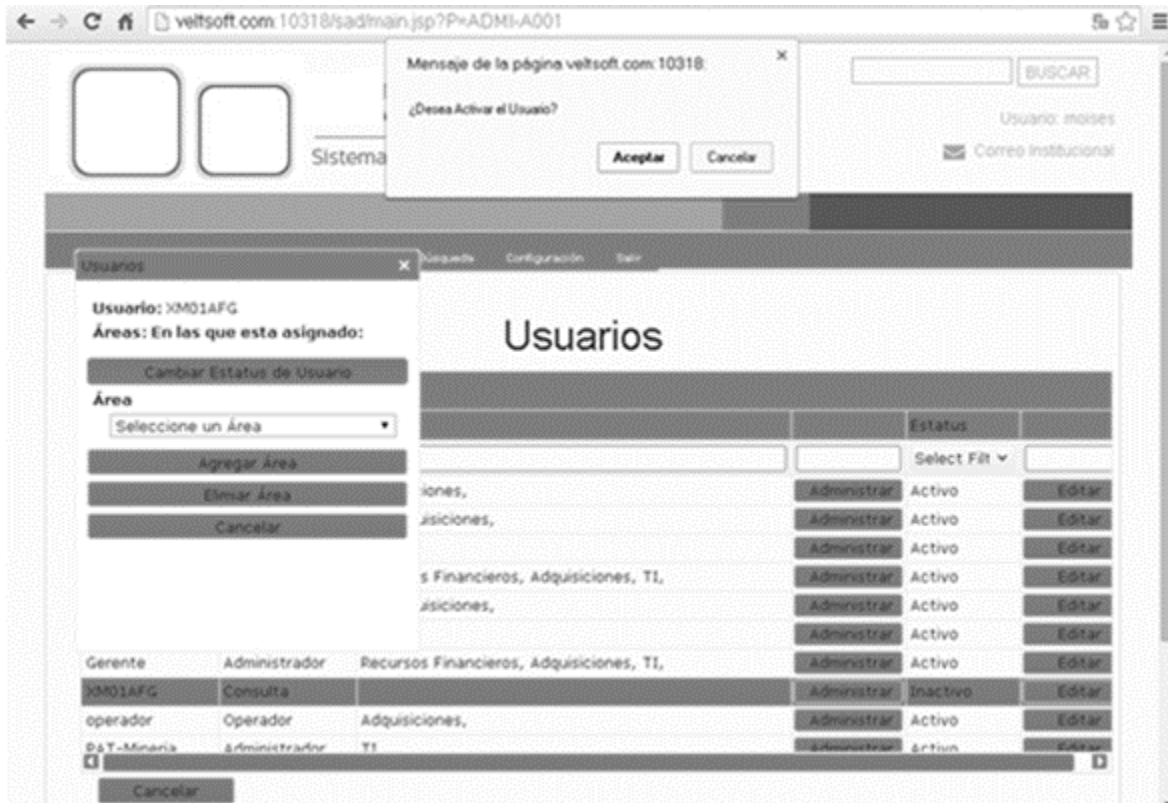


Figura 4.4.2.4 Confirmación de activación de usuario en el SAD

Se elige la opción Aceptar y el usuario cambia a estatus Activo. Posterior a esta operación se intenta ahora firmar el usuario XM01AFG al SAD y se comprueba que efectivamente ya cuenta con el acceso como se muestra en la figura 4.4.2.5 Acceso exitoso de un nuevo usuario al SAD.



Figura 4.4.2.5 Acceso exitoso de un nuevo usuario al SAD.

4.4.3 Casos de pruebas de regresión en el SAD para el caso de agregar la opción de “Eliminar” dentro de la sección “Crear plantilla”

En esta sección se aplica el concepto de pruebas de regresión sobre el módulo de creación de plantillas.

Para el presente caso se considera que se han realizado las pruebas funcionales sobre el módulo mencionado pero al obtener retroalimentación del usuario se detectó que es conveniente agregar la opción de poder eliminar un apartado en el caso de que no sea más utilizado y pueda quedar como “basura” en el sistema.

Dado que la creación de una plantilla ya se probó, se realizarán nuevamente las pruebas funcionales de creación de plantilla incluyendo además lo relativo a la nueva funcionalidad de eliminar apartados.

Se ingresa al sistema y se selecciona la opción Plantillas/Alta de Plantillas/Crear Plantilla y el sistema solicita el nombre de la plantilla. En este momento ingresamos la plantilla que se utilizará para alojar todos los sets de pruebas que en la organización se realizan al software que se desarrolla denominada “Pruebas” como puede verse en la figura 4.4.3.1 Creación de plantilla “Pruebas”.

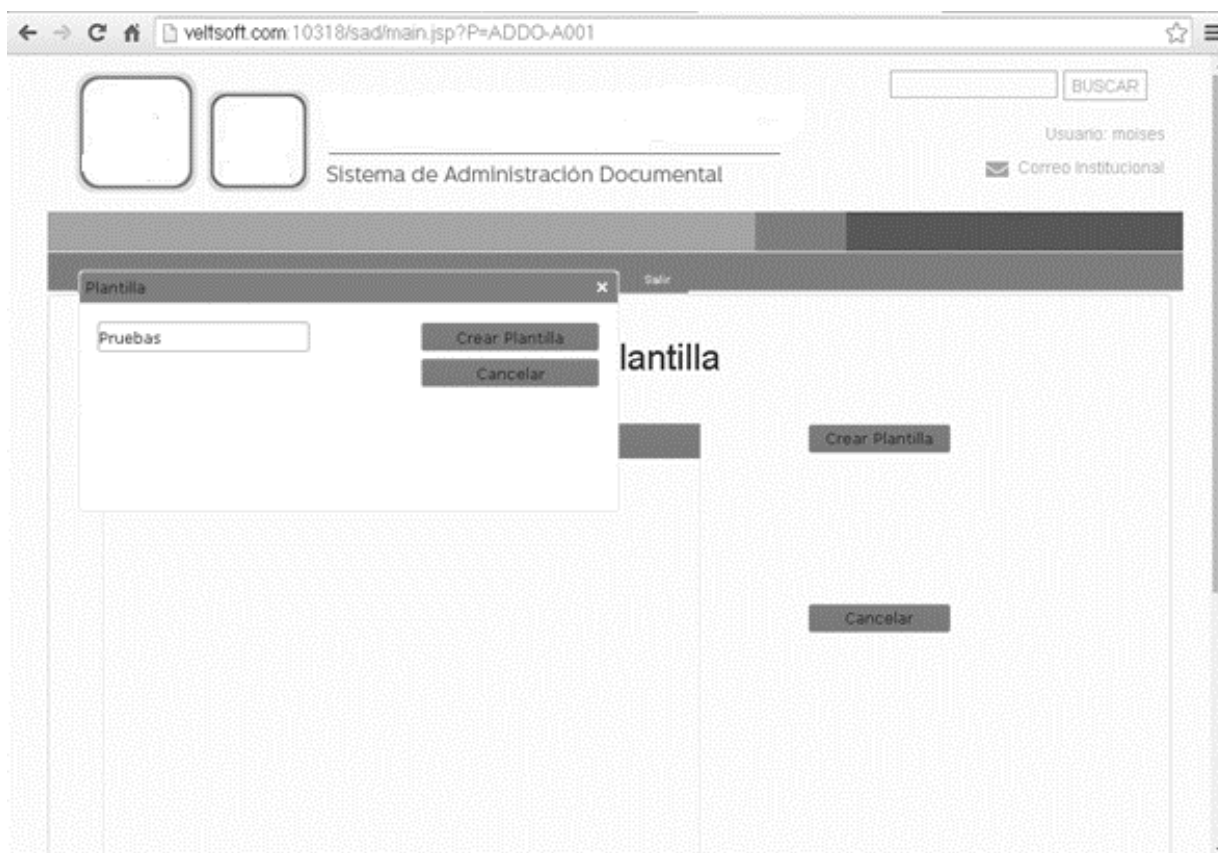


Figura 4.4.3.1 Creación de plantilla “Pruebas”.

Y como resultado podemos ver la creación del folder que contendrá las diferentes pruebas del ciclo de vida de un proyecto como puede verse en la figura 4.4.3.2 Folder para alojar las pruebas de software.



4.4.3.2 Folder para alojar las pruebas de software.

Posteriormente se elige la opción de crear apartado tres veces a fin de alojar las carpetas para las pruebas funcionales y pruebas de usuario que son las que operan en la organización. Aunado a esto se aloja un folder o apartado para alojar los vistos buenos digitalizados de estas pruebas como se muestra a continuación en la figura 4.4.3.3 Apartados de la plantilla “Pruebas”.

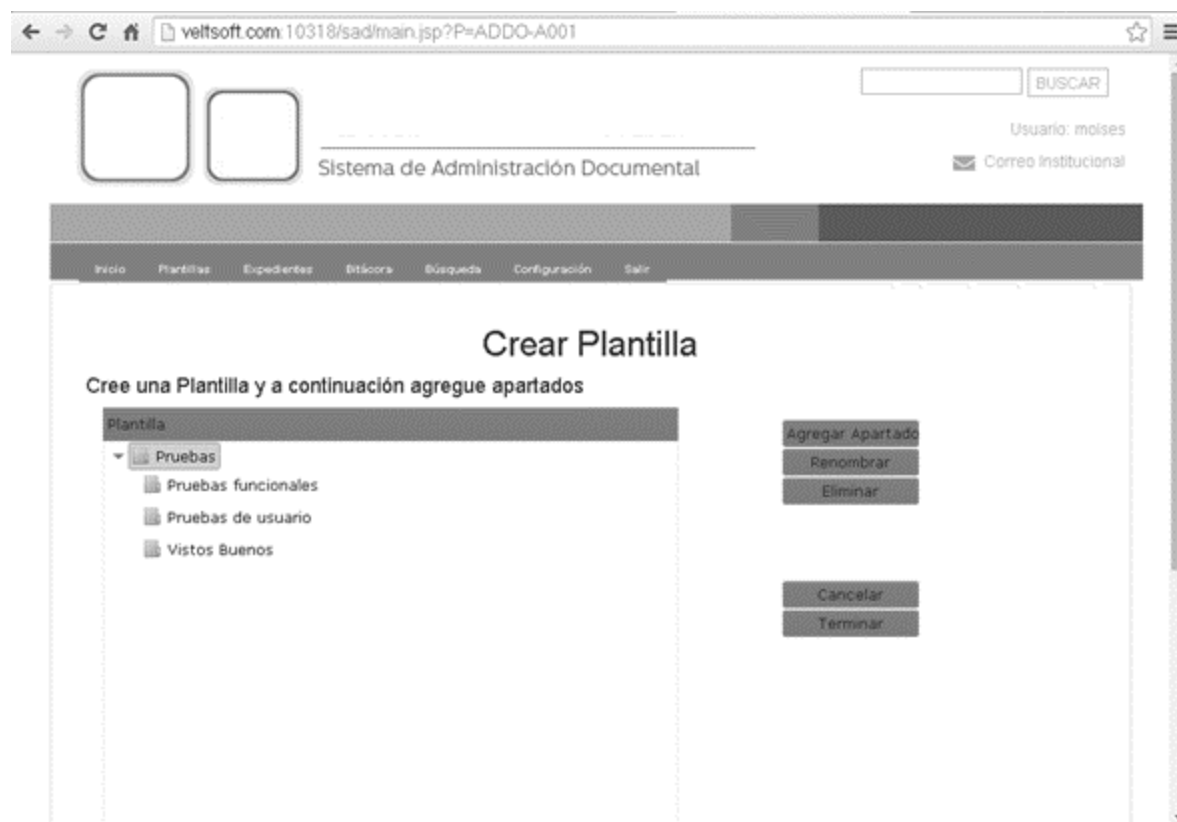


Figura 4.4.3.3 Apartados de la plantilla “Pruebas”.

En este punto elegimos la nueva opción eliminar sobre el apartado “Vistos Buenos” considerando que el usuario decide que estos mismos se alojarán en cada uno de los apartados correspondientes a las pruebas funcionales y de usuario. Para esta operación se selecciona el apartado Vistos Buenos y se elige la opción Eliminar” como se muestra en la figura 4.4.3.4 Eliminación de un apartado.



Figura 4.4.3.4 Eliminación de un apartado.

Y el resultado después de aceptar la confirmación será el siguiente que puede verse en la figura 4.4.3.5 Resultado de eliminar el apartado “Vistos Buenos”.

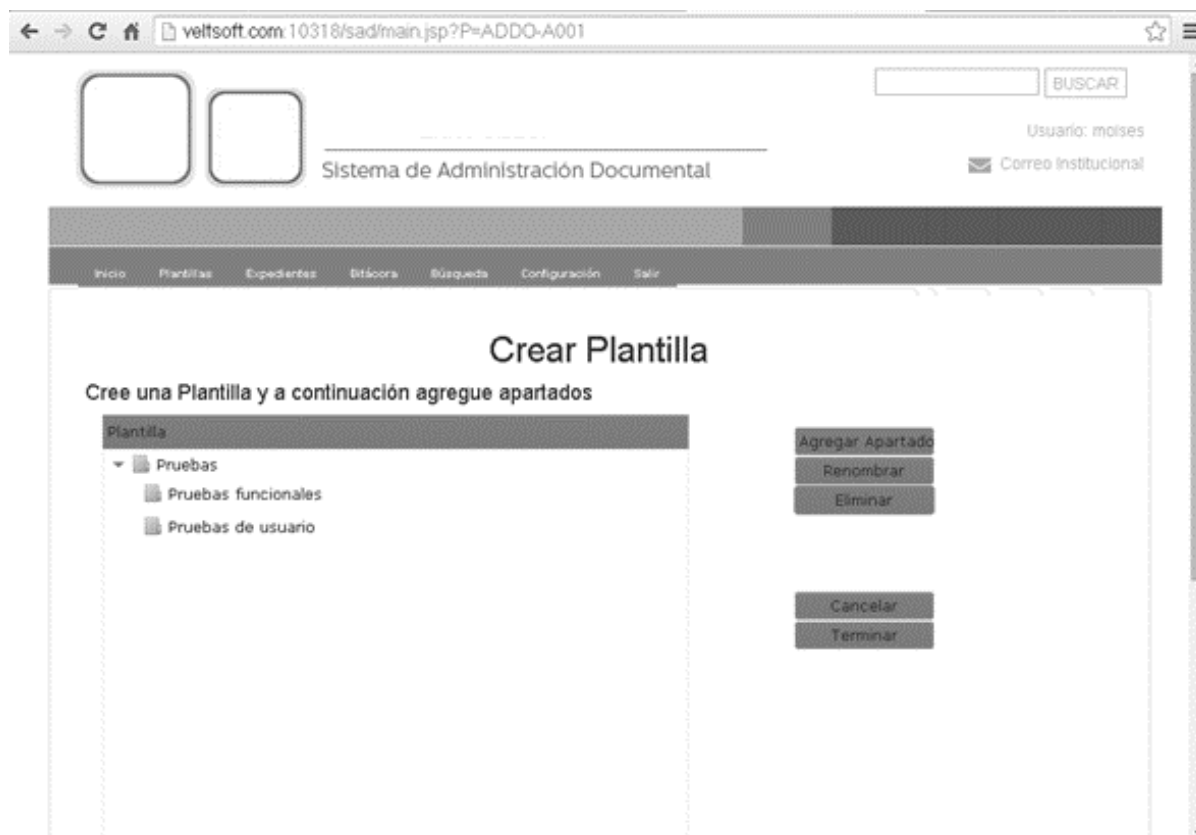


Figura 4.4.3.5 Resultado de eliminar el apartado “Vistos Buenos”.

4.4.3 Reportes del sistema.

El SAD en cuanto a reportes genera un listado de los movimientos realizados por los usuarios sobre el Sistema de Administración Documental. Dentro de esta sección del SAD se pueden especificar diferentes filtros para que el sistema extraiga la información. Estos filtros incluyen al usuario o usuarios que ingresaron al sistema, el tipo de operación realizada (Inicio de sesión, Consulta de plantillas, Cancelación de una Operación, Búsqueda de información, Edición de componentes, etc.), la fecha y hora en que se cada operación se realizó, como puede verse en la figura 4.5.3.1 Bitácora o reporte de movimientos en el SAD.

Bitácora

| Usuario | Tipo De Operación | Fecha de Re... | Hora |
|---------------|----------------------------|----------------------|----------------------|
| Select Filter | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| moises | Consulta Bitacora | 21-10-2014 | 07:33:25.0 |
| moises | Consulta Expedientes | 21-10-2014 | 07:33:08.0 |
| moises | Ingreso a Crear Expediente | 21-10-2014 | 07:32:27.0 |
| moises | Ingreso a Crear Expediente | 21-10-2014 | 07:32:02.0 |
| moises | Consulta Plantillas | 21-10-2014 | 07:31:27.0 |
| moises | Ingreso a Crear Plantillas | 21-10-2014 | 07:30:37.0 |
| moises | Inicio de Sesion | 21-10-2014 | 07:28:18.0 |
| moises | Consulta Expediente: CT/01 | 20-10-2014 | 20:09:20.0 |
| moises | Consulta Expedientes | 20-10-2014 | 20:09:16.0 |
| moises | Ingreso a Crear Expediente | 20-10-2014 | 20:07:38.0 |
| moises | Inicio de Sesion | 20-10-2014 | 20:07:34.0 |
| moises | Consulta Plantillas | 20-10-2014 | 18:39:29.0 |

Figura 4.4.3.1 Bitácora o reporte de movimientos en el SAD.

4.5 Tipos de mantenimiento.

Desde un punto de vista académico, el mantenimiento puede ser de diversos tipos; sin embargo, y a fin de no entrar en numeraciones y descripciones más o menos farragosas, en esta sección enunciaremos los principales tipos de mantenimiento que aplican en el área de sistemas en las organizaciones.

El tipo de mantenimiento más elemental es el que consiste en la reparación de una avería, la cual conlleva las actividades necesarias para restablecer un servicio cortado o deteriorado, por el fallo de algún componente del sistema. Independientemente del tipo de organización que se trate, este tipo de mantenimiento se da en todas, con mucha o menor frecuencia.

Los tipos principales de mantenimiento para nuestro caso se muestran a continuación.

Mantenimiento preventivo.

El mantenimiento preventivo como su nombre lo indica, busca prevenir fallas de un sistema dado que su costo es menor al costo total esperado en una falla del mismo. Este tipo de mantenimiento permite planificar el momento más conveniente y que genere menor riesgo en la organización.

Mantenimiento correctivo.

El mantenimiento correctivo se basa en una reacción a la ocurrencia de una falla y por consiguiente se realiza para recuperar la funcionalidad u operatividad de un equipo o sistema después de perder la capacidad para realizar la función que le corresponde.

Mantenimiento perfectivo.

Este mantenimiento se da cuando el sistema está estable y tiene el fin de mejorar o añadir nueva funcionalidad que requiere la organización. Este mantenimiento puede consistir desde en algo muy sencillo o ligero hasta un nuevo módulo funcional.

Mantenimiento aumentativo.

Este tipo de mantenimiento consiste en la modificación del software para mejorar sus propiedades a fin de aumentar la calidad y/o mantenibilidad sin alterar las especificaciones funcionales. Algunas formas de hacerlo son:

- Incluyendo sentencias que comprueben la validez de los datos de entrada.
- Reestructurar los programas para mejorar su legibilidad.
- Incluir comentarios que faciliten la posterior comprensión del programa.

Mantenimiento vertical.

Proceso por el cual se mejora y optimiza el software instalado, además se realiza la prevención de posibles problemas que pueden llegar a surgir a medida que se utiliza un equipo de cómputo.

Hoy en día este tipo de mantenimiento se realiza en busca de causas que originen inestabilidad del sistema, bajas en el rendimiento, pérdida de productividad, cortes en los sistemas y probables errores en el mismo.

Mantenimiento horizontal.

Tipo de mantenimiento que se enfoca al hardware, es decir a todos los componentes físicos y es necesario llevarlo a cabo para que el usuario, haga un mejor uso de los equipos de cómputo.

Dentro de este tipo de mantenimiento, a su vez se pueden aplicar los mantenimientos preventivos y correctivos previamente explicados.

CONCLUSIONES

Conclusiones.

- Se cumplieron todos los objetivos del trabajo, porque el Sistema de Administración Documental está funcionando.
- Con el desarrollo del Sistema de Administración Documental, se cumplieron con los requerimientos solicitados por el cliente.
- El sistema elimina gastos adicionales en copias, impresiones, carpetas y sobre todo necesidades de mayor espacio para resguardar expedientes duplicados.
- Para desarrollar sistemas de una forma profesional, es indispensable seguir una metodología con el fin de obtener un software de calidad y que sea una buena solución al problema.
- La interfaz Web del Sistema desarrollado permite una fácil interacción del usuario con el sistema, inclusive fuera de las instalaciones de la organización.
- Con este sistema se logra una mejor administración del acervo documental generado por los procesos sustantivos de la organización.
- El sistema ha permitido tener mayor control de los expedientes, llevando un mayor orden, disminuyendo los tiempos de consulta hasta en un 70% del tiempo original.
- Las Tecnologías de Información actuales son herramientas que facilitan las tareas de cualquier usuario y permiten interactuar con la tecnología de Internet.
- Se logró asegurar la integridad de los expedientes, los documentos son digitalizados y se almacenan en un servidor de base de datos seguro, en donde podrán ser consultados en tiempo real siempre que sea necesario.
- Debido a que el sistema tiene una interfaz amigable y fácil de manejar, los usuarios solo requieren de una capacitación sencilla para poder utilizarlo.
- El sistema de administración documental agiliza la gestión documental y garantiza la integridad de los expedientes, ya que eliminó la posibilidad de pérdida o extravío de documentos.
- Se cuenta con una aplicación segura y confiable, esto debido al uso de claves de usuario y contraseñas individuales.

- Las claves de usuario y contraseñas individuales protegen la información y funciones del sistema. Además de que con ellas se monitorean todas las transacciones que se lleven a cabo dentro del sistema.
- Las carreras de Ingeniería en computación y Matemáticas aplicadas a la computación, transmiten a sus egresados los conocimientos necesarios para identificar áreas de oportunidad para su sistematización.

REFERENCIAS

- Silberschatz, Abraham. (2002). Fundamentos de Bases de Datos. México: McGraw- Hill.
- Introducing Microsoft SQL Server 2012. Published by Microsoft Press.
- Lynn Beignley (2007). Head First SQL. United State of America. O'Reilly Media Inc.
- C. Thomas Wu. Introducción a la programación orientada a objetos con Java. España. McGraw-Hill.
- Edwar Yourdon. (1993). Análisis estructurado moderno. México: Prentice-hall hispanoamericana.
- Lucas Gómez Ángel. (2000). Diseño y gestión de Sistemas de Bases de Datos. México: Parainfo.
- Date CJ.. (2010). Introducción a los Sistemas de Bases de Datos Vol 1. Mexico: Addison-Weyley Iberoamericana.
- Ian Sommerville. (2002). Ingeniería del Software. México: Addison-Wesley Iberoamericana.
- Roger S. Pressman, José María Troya. (1988). Ingeniería del Software. México: McGraw Hill.
- Silberschatz, Abraham. (2002). Fundamentos de Bases de Datos. México: McGraw- Hill.
- Schmuller Joseph. (2005). Aprendiendo UML en 24 horas. México: 2005.
- Alarcón Raúl. (2007). Diseño Orientado a Objetos con UML. 2007: Grupo Eidos.
- Alarcón Raúl. (2000). Diseño Orientado a Objetos con UML. Madrid: Anaya Multimedia 2010.

Páginas de Consulta vía internet

- Gaceta Oficial de la Ley de Archivos del Distrito Federal publicado en el DOF
- R.T. Fielding and R.N. Taylor. Principled design of the modern web architecture.
- ACM Transactions on Internet Technology (TOIT), 2(2):115–150, 2002.
- http://dof.gob.mx/nota_detalle.php?codigo=5344323&fecha=13/05/2014
- <http://manuales.dgsca.unam.mx/javascript/Introduccion.html>
- <http://www.w3c.es/>
- http://www.mati.unam.mx/index.php?option=com_content&task=view&id=205&Itemid=51
- <http://msdn.microsoft.com/es-es/library/bb545450.aspx>

- <http://www.w3schools.com/sql/default.asp>
- <http://rua.ua.es/dspace/bitstream/10045/2990/1/ApuntesBD1.pdf>
- <http://www.uoc.edu/masters/oficiales/img/913.pdf>
- <https://cursos.aiu.edu/Base%20de%20Datos/pdf/Tema%203.pdf>
- http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02150.pdf
- <http://www.mssqltips.com/sqlservertip/1667/sql-server-join-example/>
- <http://clases3gingsof.wikifoundry.com/>
- <http://www.sybase.es>
- Rivera Fray León. Bases de Datos Relacionales “Instituto Tecnológico Metropolitano, 1era edición 2008, Kiosera Entorno Cliente Servidor. [Consulta en línea]
- <http://es.kioskea.net/contents/148-entorno-cliente-servidor>.
- <http://javierleal.wordpress.com>

APÉNDICES

Apéndice 1

Instalación del servidor Apache

Para la instalación del servidor apache se requiere seguir las instrucciones siguientes:

- Ir a la página oficial: www.apachelounge.com/downloads. Posteriormente descargar el parche de Visual C++ y el archivo comprimido que contiene el servidor Apache para Windows, únicamente debemos seleccionar el paquete que sea compatible con la arquitectura de 32 o 64 bits. Ver la figura 1.

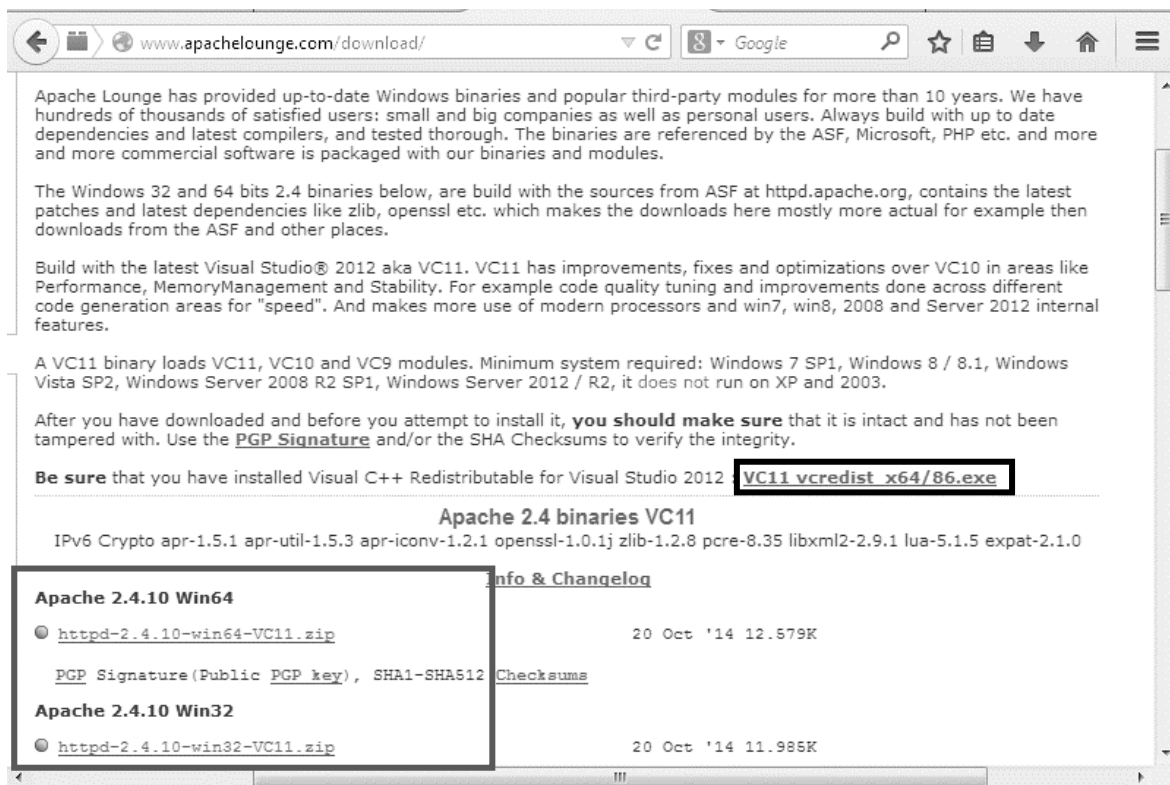


Figura 1. Página Web Apachelounge

- Instalamos el parche de Visual C++, el archivo ejecutable descargado se llama: “vc_redist_x86.exe”, posteriormente debemos aceptar los términos y condiciones del uso del software e instalamos. Ver figura 2



Figura 2. Parche Visual C++

- El siguiente paso es descomprimir los archivos de instalación de apache en una carpeta, abrimos la línea de comandos de Windows, nos dirigimos al directorio descomprimido y a la carpeta “bin” y ejecutamos la instrucción “httpd -k install”, Ver figura 3.

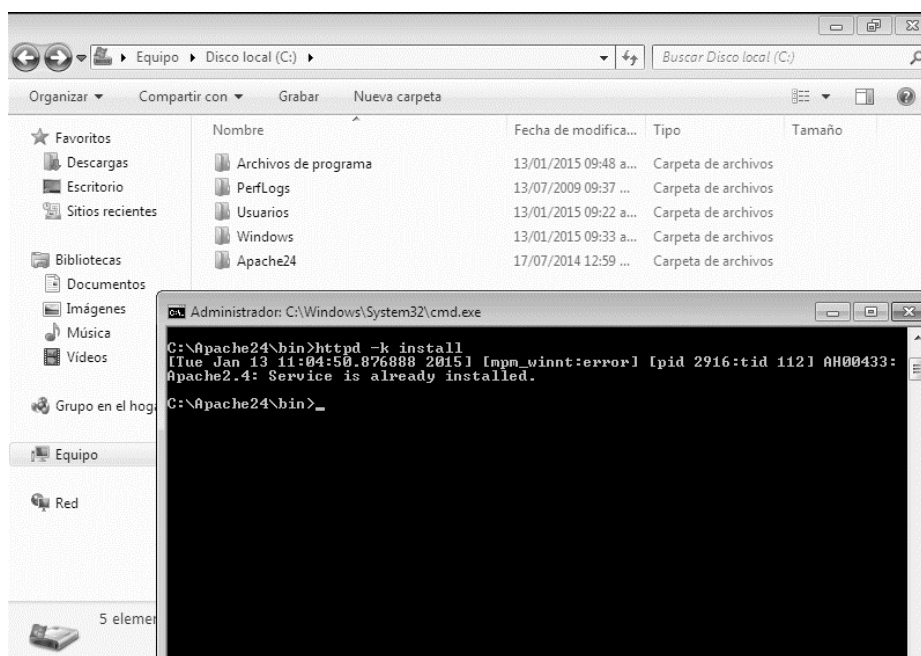


Figura 3 Instalación de Apache

- En la carpeta “/Apache24/bin”, tenemos el monitor del servidor apache, al ejecutarlo podemos administrar nuestro servidor.

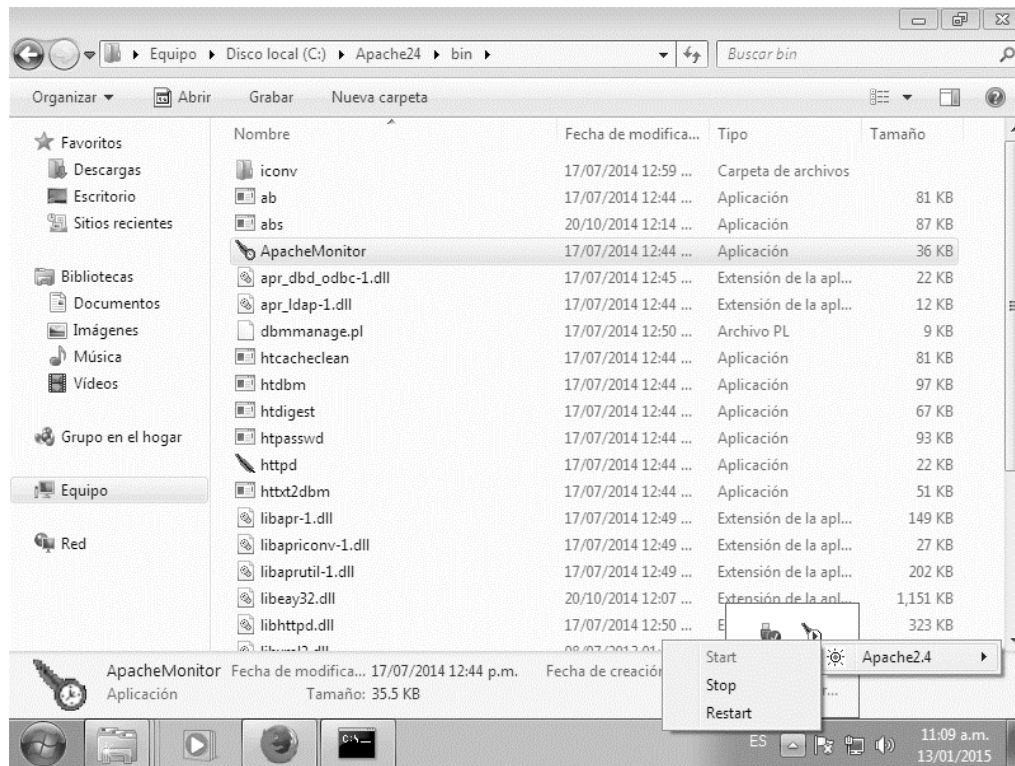


Figura 4 Monitor de Apache.