Universidad de las Ciencias Informáticas

Facultad 3

Título

"Diseño e Implementación de la solución informática para la Gestión Documental de los Registros Públicos de Venezuela".

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor

Yasmany Molina Díaz

Tutor

Ing. Oscar Camacho Acosta

Caracas, Venezuela

Mayo, 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Dirección de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

_____ _____

Yasmany Molina Díaz. Ing. Oscar Camacho Acosta.

A mis queridísimos padres por ser mis guías,

A mi hermano por todas sus enseñanzas,

A mi nenita linda por su amor y apoyo durante estos maravillosos años.

Agradecimientos

A Oscar mi tutor, por su apoyo en estos días de producción, tesis, clases, pruebas y carga.

Agradecer…

a los Sherpas y Sherpitas, por la maravillosa escalada durante estos 5 años,

a Pedro, por todas sus enseñanzas y dirigir ejemplarmente Sherpa,

a todas las personas de Registro y Notarías,

a Frank, por plantear el problema del ramo de rosas,

a mi familia del Joven Club, en especial a Osiel por iniciarme en el arte de la programación,

a mis padres, por sus genes matemáticos y físicos, con razón soy informático,

a Marien, por todo.

Resumen

En el presente trabajo se realiza un estudio de la situación actual de la Gestión Documental en los Registros Públicos de la República Bolivariana de Venezuela.

Incluye una interesante investigación sobre los sistemas de gestión documental y los planes de modernización de entornos similares en América Latina y la importancia que revisten para los países que deseen aumentar los niveles de informatización de la sociedad.

Se describen las técnicas y herramientas utilizadas en la programación así como la aplicación de patrones dentro del flujo de desarrollo de software siguiendo la metodología y procesos dictados por el Proceso Unificado de Rational (RUP, siglas en inglés Rational Unified Process).

Se definen los principales problemas con respecto a la Ley de Registro Público y del Notariado y se presenta una propuesta de solución de software para la Gestión Documental a desplegar en todas las oficinas del país, posibilitando la estandarización de los procesos registrales en las mismas.

El documento incluye las etapas de Diseño e Implementación del sistema, sobre la base del análisis a la arquitectura que soporta el sistema. Dentro de la solución se describen los casos de usos más significativos y los elementos más importantes dentro de esta.

Índice

Introducción

Después de su separación de la Gran Colombia, Venezuela mantuvo las Escribanías y la Anotación de Hipotecas y de Registro de Derechos en la misma forma, hasta la promulgación del primer Código de Procedimiento Judicial de Venezuela el 19 de mayo de 1836, que atribuía a los Escribanos y los Jueces donde no los había, el otorgar documentos hasta que se crearan las Oficinas de Registro a las cuales pasarían las funciones de la Escribanías. El otorgamiento de poderes y registro de poderes los atribuía a los Tribunales. El 24 de mayo de 1836 fue promulgada la Ley, por lo cual se establecieron y organizaron las Oficinas de Registros, la cual ordenó en la capital de cada Provincia una Oficina Principal de Registro y en cada cantón una Oficina Subalterna dependiente de la Oficina Principal. Con esta Ley quedó establecida en Venezuela la Institución de Registro Público. (Morales Galito, 2005)

La Gestión Documental de estas oficinas, aunque en aquella época no se utilizara este término, se realizaba de manera sencilla y fácil, debido a las pocas inscripciones de aquellos tiempos.

Dos siglos son suficientes para que un país cambie y sus procesos se tornen más complejos. Ejemplo de esto son los diferentes decretos de División Político-Administrativas que ha sufrido el país debido a la extensión del territorio venezolano, dando como resultado que ya no solo existan unas pocas oficinas en todo el país, sino 483 Oficinas dividas de la siguiente forma:

- 207 Registros Públicos: son los encargados de la inscripción y anotación de los actos o negocios jurídicos relativos al dominio y demás derechos reales que afecten los bienes inmuebles.
- 21 Registros Civiles: son los encargados de mantener actualizada la información acerca de los nacimientos, matrimonios, defunciones, entre otros actos del estado civil de las personas.
- 47 Registros Mercantiles: son los encargados de la inscripción y anotación de los actos o negocios jurídicos relativos al dominio y demás derechos reales que afecten los bienes mercantiles.
- 208 Notarías Públicas: son los encargados de manejo con los documentos notariales.

También dos siglos son suficientes para que nuevas ramas de la ciencia sean creadas, entre ellas la que domina actualmente el mundo, la informática. Esta nueva ciencia ha abarcado prácticamente todos los campos del saber, entre ellos también el aspecto legal.

Por estas razones, la complejidad del Sistema Registral actual en la República Bolivariana de Venezuela y la facilidad que brinda la informática en los procesos registrales, en 1993, el gobierno autoriza la digitalización de toda la información registrada en las Oficinas de Registros y Notarías. Diversos consultores nacionales y el Banco Interamericano para el Desarrollo fueron los primeros en realizar pasos concretos en torno a la tarea planteada, destacándose en esta primera etapa el Registro Primero de Caracas apoyando la informatización de 30 oficinas del país.

Los problemas generados en el sistema actual de Registros y Notarías son debido a incumplimientos o no aplicación de forma óptima de la Ley de Registro Público y del Notariado de Venezuela. Dentro de los principales problemas, en cuanto a la Gestión Documental, se destacan los siguientes:

- El 90% de las Oficinas de Registros y Notarías no poseen una infraestructura tecnológica apropiada, por lo cual, la Dirección de Registros y Notarías no puede regular un buen funcionamiento, organización y administración de los Registros Públicos. Este problema está relacionado con el objeto de la Ley, el cual se encuentra en el Artículo 1.
- La Publicidad Registral, como garantía ofrecida en los Registros, no es la mejor, en la mayoría de los casos, sólo se puede obtener la información en el lugar de origen donde se generó esta. La problemática anteriormente planteada se relaciona con el Principio de Publicidad, el cual se expresa en el Artículo 9.
- No es posible llevar a cabo el Proceso Registral y Notarial íntegramente a partir de un documento electrónico. El Artículo 23 es el vinculado con esta dificultad y tiene como centro el Manejo Electrónico.
- La Seguridad Jurídica y la Fe Pública se ve afectada por la débil Publicidad Registral al no existir un fácil acceso a la información. Este problema está vinculado el Principio de Publicidad y el Alcance de los Servicios Registrales, los cuales se ubican en los Artículos 9 y 25, respectivamente.
- No existe un procesamiento digital de la documentación, el cual le da más inseguridad al Proceso Registral, un ejemplo de esto se ve reflejado en el uso de la Firma Electrónica

mencionado en el Artículo 24 de la Ley. La implantación de un sistema con el uso de la Firma Digital ofrece una mayor seguridad, ya que es más difícil falsificar un código de identificación que falsificar una firma manuscrita.

- La información que obtiene el Ministerio del Poder Popular para Relaciones Interiores y Justicia (MPPRIJ) no es oportuna y confiable, lo cual afecta el funcionamiento, organización y administración del Sistema de Registros y Notarias. Este problema está vinculado con el objeto de la Ley, el cual se ubica en el Artículo 1.
- No existe una estandarización del Proceso Registral, por lo que no se le da un cumplimiento adecuado al Principio de Legalidad, el cual se enuncia en el Artículo 8.

La anterior información muestra la disposición actual de las Oficinas de los Registros y Notarías, de ahí la necesidad de un cambio general, tanto en la infraestructura como en el proceso.

El proceso de Gestión Registral constituye, por principio, un flujo de gestión documental que por su extensión y complejidad supera los objetivos trazados para el presente trabajo. El enfoque del proceso abordado particularizará en el tratamiento del Documento Original y la Gestión de Recaudos como características particulares y novedosas propias del negocio sujeto a estudio.

De manera general el problema que enfrenta el presente trabajo consiste en cómo diseñar e implementar la Gestión Documental de los procesos que se realizan en los Registros Públicos mediante un sistema informático integrado.

Para dar solución al anterior problema el trabajo se centra en el Proceso de Informatización de la Gestión Documental en los Registros Públicos y enfoca su solución informática en el diseño e implementación del proceso documental.

Constituye Objetivo General del trabajo diseñar e implementar una solución informática integrada para la Gestión Documental de los procesos que se realizan en los Registros Públicos.

Para alcanzar la meta propuesta y orientada fundamentalmente a proveer las especificaciones del diseño e implementación se han derivado un conjunto de objetivos específicos, siendo estos los siguientes:

- Estudiar como tiene lugar el proceso de Gestión Documental en el sistema actual de Registros Públicos en la República Bolivariana de Venezuela.
- Obtener el diseño de una solución informática que permita realizar la Gestión Documental en los Registros Públicos de la República Bolivariana de Venezuela.
- Obtener la implementación de una solución informática que permita realizar la Gestión Documental en los Registros Públicos de la República Bolivariana de Venezuela.

Con el desarrollo de este trabajo se espera garantizar resultados cuantificables, que se traducen en incremento de beneficios, reducción de costes y mejora de la productividad, además de una mejor aplicación de la Ley de Registros y Notarías:

- Se garantizará una mejor Publicidad Registral, ya que se creará toda la base para soportar una Publicidad Registral on-line.
- El Proceso Registral tendrá mayor integridad, ya que será posible llevarse a cabo con documentos electrónicos.
- La Seguridad Jurídica y la Fe Pública se verá fortalecida al crearse las bases para la gestión de la información a través de Internet.
- Existirá mayor seguridad en el Proceso Registral, pues al utilizarse el procesamiento digital de los documentos se podrá hacer uso de la firma digital.
- El MPPRIJ obtendrá información más confiable, con la que podrá darle un mejor funcionamiento, organización y administración al Sistema de Registros y Notarias.
- Se estandarizará el Proceso Registral a todas las oficinas de Registros Públicos dándole un mejor cumplimiento al Principio de la Legalidad en la Ley de Registros y Notarías.

En beneficio de obtener un trabajo de mayor calidad se trazaron tareas de investigación, las cuales se muestran a continuación:

- Estudiar las leyes que corresponden al funcionamiento de los Registros Públicos en la República de Venezuela.
- Identificar las reglas del negocio de la Gestión Documental en los Registros Públicos.
- Estudiar software con funcionalidades o características similares en otros países.
- Estudiar metodologías de diseño e implementación de sistemas.

- Describir los procesos que se van a implementar en el sistema.
- Valorar las experiencias adquiridas en el trabajo.

El presente documento se estructura en 3 capítulos y varios anexos, que incluye todo lo relacionado con el diseño e implementación de la solución informática que se propone.

El Capítulo 1 incluye aspectos sobre los Registros Públicos en Venezuela y sus objetivos. También se aborda el tema de la Gestión Documental y el papel de esta en los Registros. Más adelante en el capítulo se realiza un estudio del arte de la modernización de los Registros y sus principales pasos en el mundo, en especial en países de América Latina. Como colofón del capítulo se realiza una descripción de las principales herramientas que se utilizaron, así como la plataforma en la cual se trabajó y la metodología utilizada.

El Capítulo 2 aborda la arquitectura del sistema, se explica el modelo que se utiliza y las capas por las que está compuesto. También se realiza una descripción de *framework* utilizado, así como una valoración de este. Seguido se realiza un análisis de las soluciones no triviales presentadas en el sistema, incluyéndose dentro de este análisis el Componente de Escaneo, el Generador Dinámico de Interfaces y el Componente para la Firma Digital.

El Capítulo 3 expone la solución técnica aplicada a la Gestión Documental para los Registros Públicos. Se presentan los diagramas de análisis para el diseño, los diagramas de clases y los modelos de datos tanto lógico como físico. También se muestra el diagrama de componentes y el de despliegue.

Capítulo 1

1.1. Introducción.

En el presente capítulo se analizan los procesos que se llevan a cabo en los Registros Públicos. También se realiza un análisis a la Gestión Documental, así como el flujo de los procesos que interactúan con los documentos. Se mencionan las tendencias y tecnologías actuales en el desarrollo de aplicaciones y finalmente se fundamentan los objetivos propuestos.

1.2. Registros Públicos.

Uno de las definiciones que ofrece el diccionario Larousse a la palabra “registro” es: “Oficina donde se tramitan y guardan los registros o documentos públicos”. Por la anterior definición se pueden considerar las bibliotecas como los primeros registros de la historia, ya que el propósito general de aquellas primeras instituciones fue preservar los registros de la información de la época.

Las raíces de la profesión archivística han podido trazarse hasta el 2400 a.n.e. El primer archivo de negocios identificado fue el de los asirios en Kultepe, Capadocia, el cual data del año 2000 a.n.e. A su vez la creación de registros de las transacciones comerciales y la formación de sus colecciones originó las bibliotecas comerciales. En la antigüedad existían bibliotecas de gobierno; el interés de los gobiernos por el comercio y el registro de las operaciones comerciales llevaron a que dichos registros pasarán gradualmente a ser parte más de las bibliotecas que de los archivos. (Cañedo Andalia, 2004)

En la actualidad y a miles de millas de distancia del primer registro conocido de la historia, en la República Bolivariana de Venezuela, existen 207 Registros Públicos. Estas instituciones son las encargadas de la inscripción y anotación de los actos o negocios jurídicos relativos al dominio y demás derechos reales que afecten los bienes inmuebles.

Además de los actos señalados con anterioridad y aquellos previstos en el Código Civil, en el Código de Comercio y en otras leyes, en el Registro Público se inscribirán también los siguientes actos:

- Los documentos que contengan declaración, transmisión, limitación o gravámenes de la propiedad.

- Todo contrato, declaración, transacción, partición, adjudicación, sentencia ejecutoriada, o cualquier otro acto en el que se declare, reconozca, transmita, ceda o adjudique el dominio o propiedad de bienes o derechos reales o el derecho de enfiteusis o usufructo.
- La constitución de hogar; los contratos, declaraciones, transacciones, sentencias ejecutoriadas y otros actos que se establezcan sobre inmuebles, derechos de uso, habitación o servidumbre o se constituyan anticresis, hipotecas o se divida, se traslade o reduzca alguno de esos derechos.
- Los documentos que limiten de cualquier manera la libre disposición de inmuebles; las declaraciones, los denuncios, los permisos, los contratos, los títulos, las concesiones y los demás documentos que conforme a las leyes en materia de minas, hidrocarburos y demás minerales combustibles deban registrarse.
- Las donaciones cuando tengan por objeto bienes inmuebles.
- La separación de bienes entre cónyuges cuando tenga por objeto bienes inmuebles o derechos reales.
- Las copias certificadas de los libelos de las demandas para interrumpir prescripciones y surtir otros efectos.
- Los contratos de prenda agraria, los contratos de prenda sin desplazamiento de la posesión y los decretos de embargos de bienes inmuebles.
- Los actos de adjudicación judicial de inmuebles y otros bienes y derechos susceptibles de hipoteca, siempre que de las propias actas de remate aparezca que el crédito era legalmente exigible y que además conste en documento de fecha cierta anterior a las prohibiciones expresas.
- La constitución, modificación, prórroga y extinción de las asociaciones civiles, fundaciones y corporaciones de carácter privado.
- Las capitulaciones matrimoniales.

- Los títulos de propiedad colectiva de los hábitats y tierras de los pueblos y comunidades indígenas. (LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA, 22 de Diciembre del 2006)

1.3. Gestión Documental.

Durante más de 5000 años, los seres humanos han creado registros pictóricos y escritos que representan sus ideas, su medio, sus acciones y sus descubrimientos sobre diferentes tipos de materiales: arcilla, metal, madera, papiro, huesos, seda, pergamino, piel, papel y recientemente en el espacio virtual.

Hacia el 295 a.n.e., se fundó la Biblioteca de Alejandría bajo la égida de Tolomeo I. Este lugar fue en su época el cerebro y la gloria de la mayor ciudad del planeta. Sus emisarios recorrieron el mundo conocido hasta aquel entonces para recolectar una copia de cada libro, intento que aunque no pudo completarse, permitió reunir el cuerpo entero de la literatura griega; además de recolectarse una vasta bibliografía sobre Babilonia, la India, el antiguo Egipto y otros países del Asia Occidental. La Biblioteca de Alejandría es el lugar donde los hombres reunieron por primera vez de modo serio y sistemático toda la información del mundo. (Cañedo Andalia, 2004)

Donde quiera que se acumule información, sea esta del tipo que sea, y se necesite una consulta, se impondrá una organización, surgirá allí una gestión documental.

La misión fundamental de la gestión documental es el tratamiento integral, consistente y fiable de los documentos. Esta ayuda a catalogar, consultar, acceder e integrar información y documentación, que en la mayoría de las ocasiones, se encuentra dispersa.

En estos últimos años hemos asistido a un proceso acelerado de integración entre tecnologías (datos, imágenes, sonidos, etc.) de desarrollo de infraestructuras de comunicación y de abaratamiento general de la tecnología. De esta manera, surgen las soluciones y servicios de Gestión Documental Digitalizada, fruto de la fusión de productos físicos y lógicos de documentación, proceso de datos, imagen y sonido, en un entorno de racionalización de recursos y métodos de trabajo. (Gorospe, 2005)

1.4. La Gestión Documental dentro de los Registros Públicos.

Actualmente en las oficinas de los Registros Públicos de la República Bolivariana de Venezuela, la gestión documental se realiza de manera manual. Algunas oficinas del país tienen sistemas que los ayudan en el proceso registral, pero estos no se encuentran estandarizados dentro del Sistema de Registros y Notarías ni están enfocados en una gestión documental completamente digital.

En los Registros Públicos existen dos tipos de trámites:

- Inscripción: Son los encargados de inscribir los documentos presentados en el Registro para su protocolización y archivamiento.
- Solicitud: Son los encargados del otorgamiento de copias y certificaciones de algún documento determinado.

El flujo seguido por los trámites de Inscripción reúne todos los pasos y funcionalidades necesarios para la descripción del proceso de Gestión Documental. De esta manera, se considera innecesario aludir de forma expresa a los trámites de solicitud debido a que los pasos que lo contienen se encuentran descritos en el flujo de Inscripción. En ambos casos el documento es el núcleo de los trámites, por lo que se considera a este como el eje fundamental dentro del proceso registral.

Según la Real Academia Española un documento es: “Escrito en papel u otro tipo de soporte con que se prueba o acredita una cosa, como un título, una profesión, un contrato, etc.”, de allí la importancia que este cobra en todo el proceso, ya que este es el recipiente de toda la información a gestionar.

En el transcurso de los trámites se va conformando el documento a archivar, mediante una serie de pasos. Los trámites de Inscripción comienzan con la llegada del ciudadano al Registro y la obtención de una Planilla de Solicitud para llevar a cabo la inscripción de uno o varios actos reflejados en un documento. En el proceso registral este ciudadano es conocido como el presentante, ya que es el encargado de presentar todos los documentos necesarios para la realización del trámite.

Llegado el presentante al Registro, este llena la Planilla de Solicitud, la cual presenta con el Documento Original que contiene los actos a inscribir y los recaudos necesarios para el curso del trámite. Con todos los datos entregados, el funcionario público realiza un cálculo del costo del trámite, el cual debe ser pagado por el presentante en el banco con una Planilla de Pago generada en el

Registro. El banco le devuelve al presentante un comprobante bancario, el cual pasa a ser automáticamente uno de los recaudos necesarios para la continuidad del trámite.

Con el comprobante bancario en la mano, el presentante regresa al Registro donde se le dará continuidad al trámite, realizando oficialmente la presentación. En la presentación se recogen diferentes datos asociados al documento como son:

- Procedencia del documento: Se refiere a la persona que redactó el documento, el cual puede ser un notario, un abogado redactor u otra persona autorizada.
- Recaudos: Son todos los documentos que sirven para la validación del trámite como por ejemplo la cédula de identidad del presentante.
- Presentante: Se verifica que el presentante sea el mismo que vino al proceso de cálculo, y en caso de no ser así, se recogen los datos del nuevo presentante.
- Otorgantes: Se recogen los datos de los posibles otorgantes del trámite, aunque no es necesariamente obligatorio.

Terminada la presentación de los documentos, el trámite pasa a un estado de Revisión Legal, donde los funcionarios comienzan a realizar chequeos legales entorno al trámite. Este paso es el encargado de dar el visto bueno del trámite desde el punto de vista legal. Internamente dentro de la Revisión Legal se chequean las prohibiciones que puedan existir hacia las personas involucradas al trámite. En caso de existir, el trámite se retiene hasta que se levante la prohibición.

Cuando el trámite termina de ser revisado, se encuentra listo para ser otorgado. En el Otorgamiento se fotocopian el Documento Original junto a todos los Recaudos y se le entregan los originales de todos los documentos al otorgante. Finalmente se conforma el documento final a archivar el cual es firmado por el Registrador como constancia legal de todo el proceso.

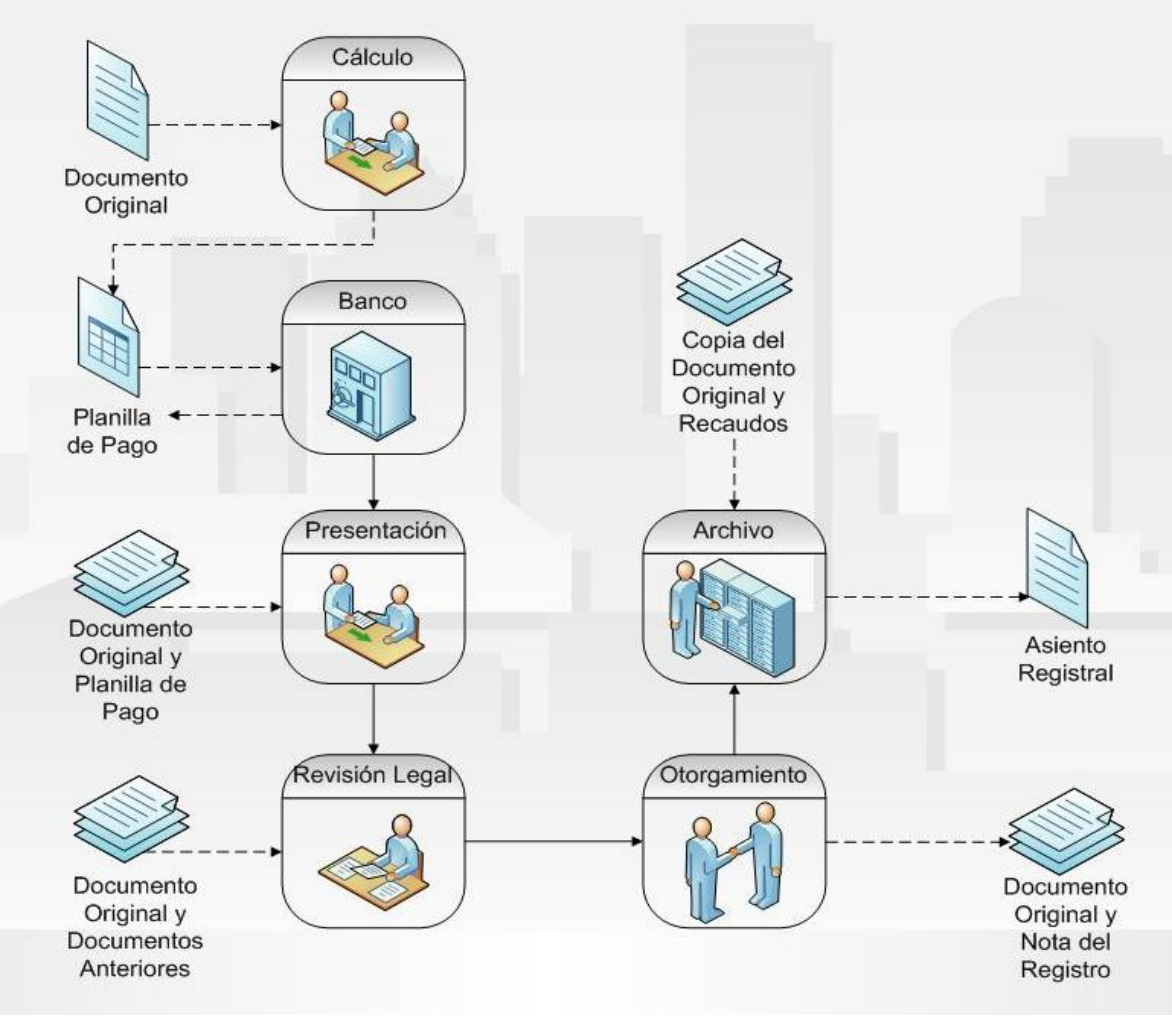

Figura 1.1 Diagrama de flujo de procesos de un documento en el transcurso de un trámite.

Debido a la complejidad del Sistema Registral actual en la República Bolivariana de Venezuela y la facilidad que brinda la informática en los procesos registrales es que se decide realizar un sistema informático integrado, que unifique y estandarice los procesos en todas las oficinas, así como se le dé un nuevo tratamiento a la gestión documental de forma digital.

1.5. Avances Tecnológicos de los Registros en América Latina.

Muchos países han trabajado en el avance tecnológico de sus Registros. Entre ellos es importante destacar Canadá, El Salvador, Brasil y México.

En el año 2003, en el 1er Congreso Iberoamericano de Registros de Propiedad, celebrado en Lima, se reunieron diferentes funcionarios de distintas regiones del mundo. Es importante destacar el avance de América Latina en el afán de la modernización de sus Registros. En este evento países como Perú, República Dominicana y Colombia presentaron sus propuestas para la modernización de sus oficinas. A continuación se muestran algunos de los datos presentados por Colombia en dicho evento.

Modernización en Colombia.

El sistema registral colombiano ha evolucionado por diferentes etapas, las cuales se destacan a continuación de forma cronológica:

- 1887 - Se implanta el sistema de libros múltiples: Este sistema se tornó dispendioso y complejo para reflejar el verdadero estado de la tradición de los inmuebles, al igual que para la expedición de certificados y consultas.
- 1932 - Entra en vigor el sistema real: Se caracterizaba por libros de gran formato, divididos en columnas con la información relativa a la tradición de la propiedad, limitaciones del dominio, gravámenes, embargos y demandas, entre otros aspectos.
- 1970 y 1984 - Se comienza a usar el folio real de matrícula inmobiliaria y el folio de matrícula inmobiliaria por cartulinas: Estos sistemas unificaron la información de la tradición de la propiedad en un folio de matrícula inmobiliaria para cada predio. Contenían información genérica como: el círculo registral, el departamento, el municipio, la nomenclatura, descripción del área y linderos.
- 1984 - Entra en vigor el sistema de folio magnético: Sistematiza el proceso desde el momento en que el ciudadano llega a la ventanilla de caja, hasta la entrega del certificado de libertad y tradición. Este sistema facilitó la operación de la función registral y la consulta por medio de códigos.

En la actualidad conviven el sistema antiguo (libros), el de folio real (cartulinas) y el de folio magnético. Sólo 76, de las 190 Oficinas de Registros de Instrumentos Públicos existentes en Colombia para el 2003, se encontraban informatizadas. Las 114 oficinas restantes manejaban todos su procesos de manera manual, mantenían todos sus archivos en medios físicos (libros y cartulinas),

lo que representaba un alto riesgo en su conservación y almacenamiento ya que las condiciones de los archivos eran precarias.

Al proyecto presentado en el Congreso Iberoamericano se le estimaba una duración de 3 años, donde se planteaba las siguientes tareas:

- Creación de un Servidor, un Portal Web y un Almacén de Datos para la implantación del sistema.
- Migración hacia el nuevo sistema.
- Crear conectividad entre todos los Registros.
- La digitalización de los documentos de los Registros.

Colombia contaba en el 2003 con 76 oficinas sistematizadas y 114 no sistematizadas. En las sistematizadas se encontraban 9 712 800 folios distribuidas en 76 bases de datos, lo cual representaban un 82% del total de folios existentes en las 190 oficinas. Mientras que en las no sistematizadas existían 2 163 675 folios en papel que debían digitalizarse, lo cual equivale al 18% restante. (Lafaurie, 2003)

Modernización en Perú.

En el año 2005, se celebra en Santa Cruz de la Sierra, en Bolivia, el 3er Encuentro Iberoamericano de Derecho Registral. En este evento Perú, muestra sus avances en la modernización de sus Registros.

El sistema registral peruano, al igual que el colombiano, ha transitado también por diferentes etapas, las que se describen cronológicamente a continuación:

- 1888 - Se implanta un sistema basado en tomos: Consistía en archivar las inscripciones en libros foliados de considerable tamaño con 500 páginas cada uno. Era un sistema complejo y tenía muy poca práctica.
- 1971 - Entra en vigor el sistema basado en fichas: Este sistema operaba como un instrumento asignado a cada título de propiedad, persona jurídica o natural, en la cual se registraban todas las modificaciones de los actos y contratos, lográndose de esta forma, cierta facilidad para el manejo de los asientos.

- 1994 - Se crea la Superintendencia Nacional de los Registros Públicos (SUNARP): Tiene la finalidad de mantener y preservar la unidad y coherencia del ejercicio de la función registral en todo Perú, orientado a la especialización, simplificación, integración y modernización de la función, procedimientos y gestión de todos los registros que lo integran.
- 1997 hasta 2001 - Se introducen los Asientos Electrónicos dentro del sistema: En este año se realizaron los primeros Asientos Electrónicos, creándose así el Sistema de Información Registral (SIR), que sustenta los registros de propiedad de inmuebles, personas jurídicas y naturales. Durante 1998 se crea la Sala Virtual de Partidas Registrales y posteriormente en 1999 el servicio de información en línea.
- 2002 hasta 2004 - Se crea interconexión de Publicidad en Línea: Esta interconexión se crea en línea simple a nivel nacional, que comprende la red privada de comunicaciones entre todas las oficinas registrales, gestión remota a acceso a través de Internet y una bodega central.
- 2005 - Se crea la Inscripción Registral de Competencia Nacional: Se refiere a un Proyecto de Modernización que comprende el desarrollo tendiente a estandarizar los medios de producción de los servicios registrales en el ámbito nacional. (Bailón Cabrera, 2005)

El SUNARP se encuentra organizado con una Sede Central, 13 Zonas Registrales dentro de las que se encuentran 58 Oficinas Registrales y 30 Oficinas Receptoras. Cuentan con un total de 1625 trabajadores.

En 1996 durante el proceso de Modernización Tecnológica en la Oficina Registral de Lima, se trabajó en la digitalización de 5 millones de imágenes y su indexación. Esta fue la base de los siguientes pasos que siguió al proceso de modernización, los cuales fueron la Generación de los Asientos Electrónicos, la Creación de la Sala Virtual de Partidas Registrales y la Creación del Servicio de Información en Línea.

Para el 2000 y hasta el 2002 el número de digitalizaciones e indexaciones aumentó hasta 7 millones de imágenes, haciéndose este proceso sobre la información contenida en los documentos registrales (Tomos y Fichas). Esta vez fue la base para la creación del Sistema de Información Registral, la cual se logró realizar de forma estándar y se incorporó la captura de las huellas dactilares. Todo esto se desarrolló sobre servidores de bases de datos de Oracle, usando KeyFile como administrador de imágenes y sobre sistemas operativos en UNIX y NT. (Freitas Alvarado, et al., 2005)

Finalmente las etapas de la modernización tecnológica de los Registros Públicos se dividieron en cuatro, las cuales se muestran a continuación:

- Primera etapa: Selección de una plataforma estándar.
- Segunda etapa: Implementación nacional del Sistema de Información Registral.
- Tercera etapa: Interconexión nacional de las 58 oficinas registrales.
- Cuarta etapa: Proyecto de Inscripción Registral con Competencia Nacional. (Montalván Pérrigo, 2003)

Ninguno de los procesos de modernización que se llevan actualmente en el mundo fuera posible sin el auge que experimenta la informática desde hace unos años. Nuevas e innovadoras tecnologías dominan el mundo y luchan por tener una mayor presencia en el mercado, así como una creación de software más rápido y eficiente.

1.6. Tendencias y Tecnologías.

En el presente epígrafe se realiza un análisis de las tendencias y tecnologías actuales en el campo del desarrollo de aplicaciones. También se analizan los sistemas gestores de bases de datos, RUP como metodología de desarrollo de software, entornos de desarrollo y herramientas CASE. Es importante destacar que cada uno de las herramientas o metodologías que se describen a continuación fueron escogidas por la Dirección del Proyecto de Registros y Notarías.

1.6.1. Rational Unified Process (RUP).

El RUP es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El RUP utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos. (Jacobson, y otros, 2000)

El RUP se sostiene sobre 3 ideas básicas:

- El proceso está dirigido por casos de usos.
- El proceso está centrado en la arquitectura.
- El proceso es iterativo e incremental.

Dirigido por casos de uso.

En el RUP los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de usos y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Centrado en la arquitectura.

El RUP asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando se construye un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.

Iterativo e Incremental.

El RUP es un marco de desarrollo iterativo e incremental compuesto por cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación, Prueba, entre otras. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

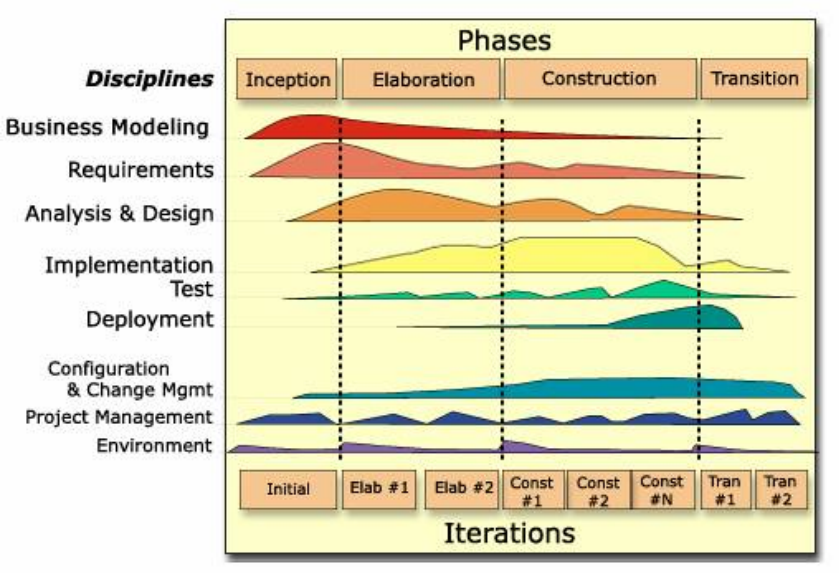

Figura 1.2 Fases e iteraciones del Proceso Unificado.

1.6.2. Programación Orientada a Objetos.

La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (datos), comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Según Grady Booch, es un “método de implantación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son miembros de una jerarquía de clases unidas mediante relaciones de herencia. En tales programas, las clases suelen verse como estáticas, mientras que los objetos suelen tener una naturaleza mucho más dinámica, promovida por la existencia de la ligadura dinámica y el polimorfismo”. (Booch, 1994)

La programación orientada a objetos es una forma de programar. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Por la importancia y continua mención dentro en el documento se considera conveniente enunciar los más destacados:

- Objeto: Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad. Corresponden a los objetos reales del mundo que nos rodea, o a objetos internos del sistema.
- Clase: Definiciones de las propiedades y comportamiento de un conjunto de objetos concretos. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- Método: Algoritmo asociado a un objeto, cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- Evento: Un suceso en el sistema. El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.
- Mensaje: Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Propiedad y atributo: Contenedor de un tipo de datos asociados a un objeto, que hace los datos visibles desde fuera del objeto, y cuyo valor puede ser alterado por la ejecución de algún método.
- Estado interno: Es una propiedad invisible de los objetos, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto.
- Componentes de un objeto: Atributos, identidad, relaciones y métodos.
- Representación de un objeto: Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

Hay un cierto desacuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como "orientado a objetos", pero hay un consenso general en que las características siguientes son las más importantes:

- Abstracción: Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el

sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.

- Encapsulamiento: Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- Principio de ocultación: Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que específica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.
- Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que reimplementar su comportamiento. Esto suele hacerse habitualmente agrupando los

objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple; esta característica no está soportada por algunos lenguajes (como Java).

1.6.3. Patrones.

Los patrones son una disciplina de resolución de problemas reciente en la ingeniería del software que ha emergido en mayor medida de la comunidad de orientación a objetos, aunque pueden ser aplicados en cualquier ámbito de la informática y las ciencias en general. Los patrones tienen raíces en muchas áreas, incluyendo la *literate-programming*.

De la misma forma que sucede con otros conceptos de la informática (y como es el caso de la arquitectura), no es fácil establecer una definición taxativa y definitiva del concepto de patrón.

En el libro “The Timeless Way of Building” (obra de referencia donde se plantea por primera vez la teoría de los patrones aplicada a la Arquitectura Civil y Urbanismo), el arquitecto Christopher Alexander define al patrón en la siguiente manera:

“Cada patrón es una regla de 3 partes, que expresa una relación entre un contexto, un problema y una solución. Como un elemento en el mundo, cada patrón es una relación entre un contexto, un sistema de fuerzas que ocurren repetidamente en ese contexto y una configuración espacial que permite que esas fuerzas se resuelvan entre sí.”

“Como elemento de un lenguaje, un patrón es una instrucción que muestra como puede ser usada esta configuración espacial una y otra vez para resolver el sistema de fuerzas, siempre que el contexto lo haga relevante.” (Alexander, 1979)

Se considera importante e interesante citar 2 principios postulados por Martin Fowler en “*Analysis Patterns: Reusable Object Models*” y que se debe tener en mente en todo momento al utilizar los patrones:

- Los patrones son un punto de partida, no un destino.
- Los modelos no están bien o mal, sino que son más o menos útiles. (Fowler, 1997)

Documentar un patrón puede ser una tarea muy difícil. Citando a James Coplien en *Home of the Patterns Library*, un buen patrón:

- Resuelve un problema: Los patrones capturan soluciones, no principios o estrategias abstractas.
- Es un concepto probado: Capturan soluciones, no teorías o especulaciones.
- La solución no es obvia: Muchas técnicas de resolución de problemas (como los paradigmas o métodos de diseños de software) intentan derivar soluciones desde principios básicos. Los mejores patrones generan una solución a un problema indirectamente (un enfoque necesario para los problemas de diseño más difíciles).
- Describe una relación: Los patrones no describen módulos sino estructuras y mecanismos.
- Tiene un componente humano significante: El software sirve a las personas. Los mejores patrones aplican a la estética y a las utilidades (de hecho, no es casual que varios de los primeros lenguajes de patrones tengan que ver con temas estéticos y utilidades). (Hillside Group, 2003)

1.6.4. Microsoft .NET.

Microsoft .NET es una plataforma de software que conecta información, sistemas, personas y dispositivos. La plataforma .NET conecta una gran variedad de tecnologías de uso personal y de negocios, de teléfonos celulares a servidores corporativos, permitiendo el acceso a información importante, donde y cuando se necesiten.

Desarrollado con base en los estándares de Servicios Web XML, .NET permite que los sistemas y aplicaciones, ya sea nuevos o existentes, conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo.

.NET es un "ingrediente" presente en toda la línea de productos Microsoft, ofreciendo la capacidad de desarrollar, implementar, administrar y utilizar soluciones conectadas a través de servicios, de manera rápida, económica y segura. Estas soluciones permiten una integración más rápida y ágil entre las empresas y el acceso a información a cualquier hora, en cualquier lugar y a través de cualquier dispositivo.

Es claro entonces que el objetivo de la plataforma .NET es simplificar el desarrollo de aplicaciones. Provee las herramientas y tecnologías para transformar la red en una plataforma de computación distribuida en gran escala. Esta plataforma además soporta los estándares sobre los cuales se basan los servicios.

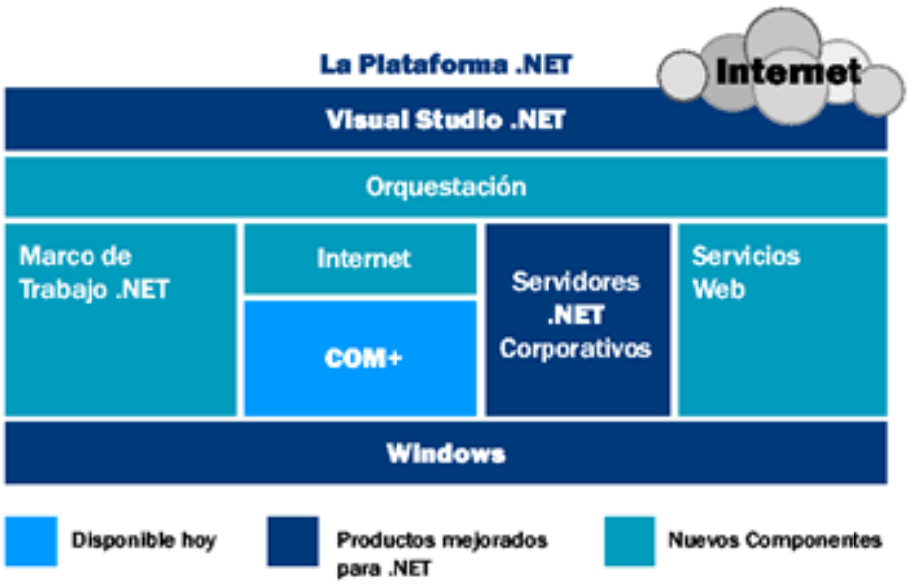

Figura 1.3 Herramientas y tecnologías de la Plataforma .NET.

La plataforma .NET utiliza tecnologías existentes, productos modificados para su uso dentro de la plataforma y elementos nuevos. (Microsoft Inc., 2003)

Con una plataforma de software definida, es importante conocer cómo desarrollar aplicaciones de la forma más sencilla posible sobre dicha plataforma. En .NET quien tiene semejante responsabilidad es *.NET Framework*.

1.6.5. .NET Framework.

El *.Net Framework* es un conjunto de servicios de programación diseñados para simplificar el desarrollo de aplicaciones en entornos altamente distribuidos. El *.NET Framework* se instala como un componente aparte en Windows 2000, mientras que Windows XP y las futuras versiones de Windows lo incorporan directamente al sistema operativo. Como por ejemplo Windows Server 2003 o Windows .NET CE.

El *.NET Compact Framework* permite hacer uso de los servicios en dispositivos móviles. Debido a que es un subconjunto del *.NET Framework* comparte el mismo modelo de programación y herramientas de desarrollo de aplicaciones haciendo posible que los desarrolladores transfieran sus conocimientos existentes al desarrollo de aplicaciones móviles.

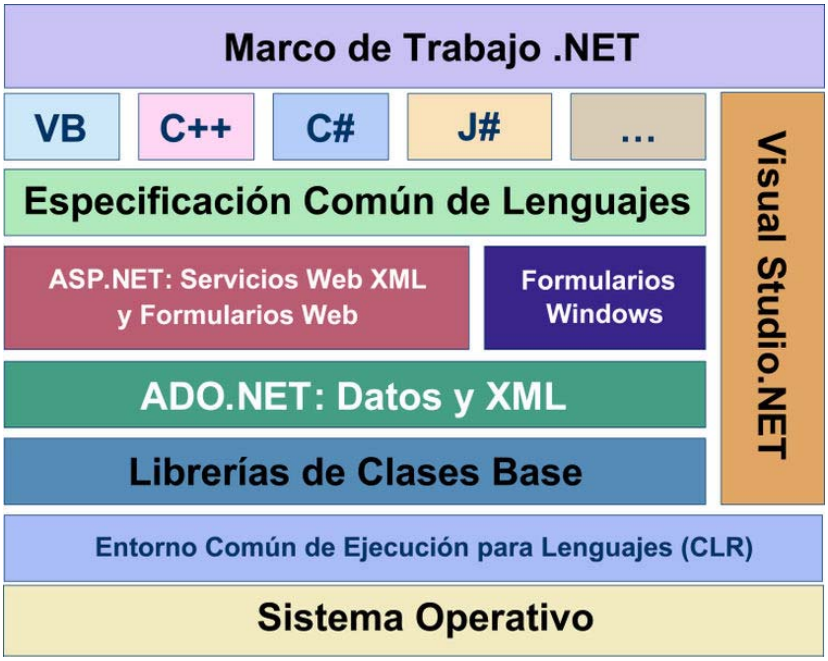

Figura 1.4 Marco de Trabajo .NET.

Los componentes del *.NET Framework* proveen los "ladrillos" necesarios para construir las aplicaciones Web, los servicios y cualquier otra aplicación dentro de Visual Studio .NET.

A continuación presentamos los componentes que componen el *.NET Framework* con sus respectivas funciones:

- Entorno Común de Ejecución para Lenguajes: Provee lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta.
- Cargador de Clases: Permite cargar en memoria las clases.
- Compilador MSIL a nativo: Transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.

- Administrador de Código: Coordina toda la operación de los distintos subsistemas del Entorno Común de Ejecución para Lenguajes.
- Recolector de Basura: Elimina de memoria objetos no utilizados.
- Motor de Seguridad: Administra la seguridad del código que se ejecuta.
- Motor de Depuración: Permite hacer un seguimiento de la ejecución del código aún cuando se utilicen lenguajes distintos.
- Verificador de Tipos: Controla que las variables de la aplicación usen el área de memoria que tienen asignado.
- Administrador de Excepciones: Maneja los errores que se producen durante la ejecución del código.
- Soporte de multiproceso (*threads*): Permite ejecutar código en forma paralela.
- Empaquetador de COM: Coordina la comunicación con los componentes COM para que puedan ser usados por el *.NET Framework*.
- Soporte de la Biblioteca de Clases Base: Interface con las clases base del *.NET Framework*. (Microsoft Inc., 2003)

Conociendo las funcionalidades que ofrece el *.NET Framework* sólo faltaría estudiar un entorno de desarrollo apropiado para crear aplicaciones sobre la plataforma .NET.

1.6.6. Visual Studio .NET 2003.

Visual Studio .NET es un entorno de desarrollo integrado pensado para hacer rápida y fácilmente las aplicaciones de la nueva generación. (Microsoft Inc., 2003)

Figura 1.5 Herramientas de Visual Studio .NET.

Visual Studio .NET 2003 incluye los lenguajes de programación: C#, Visual Basic. NET y Visual C++ .NET. Este IDE puede utilizarse para construir aplicaciones dirigidas a Windows (utilizando Windows Forms), Web (usando ASP.NET y Servicios Web) y dispositivos portátiles (utilizando .NET Compact Framework).

La característica más notable del IDE es su soporte de los nuevos lenguajes .NET. Los programas desarrollados en esos lenguajes no se compilan a código máquina ejecutable (como por ejemplo hace C++) sino que son compilados a algo llamado CIL. Cuando los programas ejecutan la aplicación CIL, ésta es compilada en ese momento al código de máquina apropiado para la plataforma en la que se está ejecutando. Mediante este método, Microsoft espera poder soportar varias implementaciones de sus sistemas operativos Windows (como Windows CE). Los programas compilados a CIL pueden ejecutarse sólo en plataformas que tengan una implementación de *.NET Framework*. Es posible ejecutar programas CIL en Linux o en Mac OS X utilizando algunas implementaciones .NET que no pertenecen a Microsoft, como Mono y DotGNU.

Con un entorno de desarrollo seleccionado, es muy importante a continuación definir el Sistema de Gestión de Base de Datos con el cual se interactuará, ya que los datos con los que se trabajan en el sistema deben ser persistidos.

1.6.7. Oracle.

Oracle es un sistema de administración de base de datos líder en la industria, el cual se puede utilizar para almacenar todo tipo de datos de negocio, incluyendo datos relacionales, documentos, multimedia, XML y datos de localización. Está disponible para un ancho rango de plataformas y escalas desde sistemas de un simple procesador hasta rejillas de sistemas interconectados. Oracle es la primera base de datos diseñada para *Grid Computing*. Es fácil desarrollar y administrar, puede manipular todo tipo de administración de datos y ofrece una excepcional disponibilidad, escalabilidad, fiabilidad y seguridad. (Oracle, 2005)

Uno de los mayores problemas que presenta las bases de datos de Oracle, es el alto precio de sus patentes y licencias, con respectos a otros sistemas gestores de bases de datos, es por eso que a veces sólo se puede encontrar en grandes compañías y multinacionales.

Oracle se basa en la tecnología cliente-servidor, por lo que se requiere que para su utilización se haya instalador antes una herramienta servidor y posteriormente poder utilizar la base de datos desde herramientas de desarrollo como Oracle Designer y Oracle Developer.

Dentro de la arquitectura de nuestro sistema existen 2 servidores de Base de Datos, el primero se encuentra en el Centro de Datos, mientras el segundo se encuentra en cada oficina de los Registros Públicos. El servidor del Centro de Datos cuenta con Oracle Real Application Cluster 10.2.0.3, mientras el servidor de la Oficina cuenta con Oracle Standard Edition 10.2.0.3.

Con el Sistema de Gestión de Base de Datos seleccionado, es importante desarrollar una capa de acceso a datos, que permita llevar los datos desde el modelo del negocio del sistema hasta la base de datos. En esta tarea es muy importante la herramienta que se describe a continuación.

1.6.8. TierDeveloper.

TierDeveloper es una herramienta de mapeo de objetos relacionales que ayuda al desarrollo de complejas aplicaciones de bases de datos en .NET. Dentro de las características que presenta el citado software se encuentran el de mapear objetos desde bases de datos relacionales, crear reglas del negocio y SQL dentro de estos objetos y generar el código de trabajo de forma instantánea y completa.

Algunos de los beneficios que brinda el TierDeveloper se muestran a continuación:

- Incremento de la velocidad de desarrollo en .NET: 50 % del código es generado instantáneamente.
- Mejora la calidad del código: Todo el código generado es consistente y basado en patrones de diseño.
- Reduce los esfuerzos de prueba: Todo el código generado es basado en plantillas probadas anteriormente.

TierDeveloper es una poderosa herramienta de mapeo de objetos relacionales y posee un valioso conjunto de características. Algunas de las principales características que se usan en el desarrollo de aplicaciones son las siguientes:

- Mapeo de objetos: TierDeveloper permite mapear objetos desde uno o varias tablas. Se puede seleccionar cuales columnas se desean mapear como atributos del objeto, renombrar atributos del objeto para así ajustarse a la convención del código que se haya definido, especificar columnas de solo lectura, y mucho más.
- Consultas: Se pueden definir consultas como métodos de los objetos. Estos métodos devuelven colecciones de objetos y proporcionan una verdadera vista orientada a objetos de la base de datos. Las consultas pueden contener complejas uniones, consultas anidadas y mucho más.
- Relaciones: Se pueden manipular fácilmente relación de 1-1 y 1-n en los mapeos. Las relaciones de las bases de datos son representadas como relaciones de objetos en la aplicación. Se pueden definir consultas de estas relaciones.

- Llamadas a procedimientos almacenados: Se pueden definir métodos en los objetos para llamar a procedimientos almacenados. Estos permiten usar los procedimientos almacenados y acceder a estos como si fueran métodos.
- Relaciones padre-hijo: En caso de definirse una relación 1-1 o 1-n de padre - hijo TierDeveloper genera código para manipularlo de forma correcta. Los objetos padres manipulan eficientemente ciclos de sus hijos.
- Operaciones personalizadas: En ciertas ocasiones, la aplicación debe cargar o actualizar algunas pocas columnas en las tablas para impulsar el rendimiento. Este propósito se cumple con las operaciones personalizadas.
- Operaciones en lotes: Define métodos en los objetos para llevar a cabo actualizaciones y eliminaciones en lotes desde la base de datos.
- Eventos personalizados: Personaliza el comportamiento del código generado en eventos, los cuales pueden ser preservados la próxima vez que se genere el código.
- Servicios Web: Si se desarrollan aplicaciones orientadas a servicios distribuidos, se pueden definir servicios web para los objetos generados y usar estos desde aplicaciones remotas. (Alachisoft, 2007)

1.7. Conclusiones.

Como resultado de la investigación y el análisis bibliográfico realizado, a lo largo del capítulo han sido expuestos los principales puntos de interés abordados en la investigación. Se han tratado temas como la necesidad de una reorganización desde el punto de vista informático en el proceso registral venezolano y la situación actual de los Registros en Venezuela.

Se ha abordado también los Registros Públicos, así como sus objetivos. El tema de la Gestión Documental, no quedó detrás y se destacó la importancia que esta toma dentro de los Registros Públicos. Como colofón a la relación entre la Gestión Documental y los Registros Públicos, se realizó un estudio de los avances tecnológicos en América Latina, destacándose Colombia y Perú.

Las herramientas y metodologías a utilizar también han sido objeto de análisis, mostrándose sus características y su funcionamiento, incluyéndose también la plataforma de desarrollo y el Sistema de Gestión de Base de Datos.

Se considera que los objetivos que se pretendían cumplir con el desarrollo de este capítulo han sido cumplidos y como resultado han quedado claros muchos conceptos importantes para dar continuidad y soporte al trabajo de investigación.

Capítulo 2

2.1. Introducción.

En el siguiente capítulo se describe la arquitectura que utiliza el sistema. Se abordan las capas por las que está compuesta la arquitectura y se brinda un breve análisis de estas. También se analizan los componentes más importantes utilizados en la Gestión Documental, así como una visión de las intenciones de los desarrolladores al crearlos.

2.2. Arquitectura.

La palabra arquitectura proviene del griego “αρχ”, cuyo significado es “jefe, quien tiene el mando”, y de “τεκτων” que significa “constructor o carpintero”. Así, para los antiguos griegos el arquitecto es el jefe o el capataz de la construcción y la arquitectura es la técnica o el arte de quien realiza el proyecto.

La arquitectura del software es el arte de proyectar y construir el software, engloba no sólo la capacidad de diseñar los espacios, sino también la ciencia de construir y moldear los bloques por los que está compuesto.

La arquitectura del sistema de software del Proyecto de los Registros y Notarías de la República Bolivariana de Venezuela se encuentra organizada desde dos enfoques, uno vertical y otro horizontal, los cuales se muestran a continuación.

Enfoque horizontal.

El sistema está dividido en 4 módulos principales, de los cuales 3 comparten la misma arquitectura y por la misma razón serán el centro de atención de nuestro trabajo cuando se haga referencia a los módulos.

Cada uno de los módulos responde a un conjunto de funcionalidades y casos de uso específicos del cliente. Todos estos módulos interactúan y comparten datos de interés en dependencia de la funcionalidad de cada uno y siguiendo un estricto régimen de seguridad de la información. Los módulos son:

- Registros Públicos.
- Registros Mercantiles.
- Administración Financiera.

Enfoque vertical.

El desarrollo de cada módulo responde a un modelo multicapas. En este caso se decidió concebir 5 capas las cuales se muestran a continuación:

- Capa interfaz.
- Capa mediadora interfaz basada en acciones.
- Capa lógica del negocio.
- Capa acceso a datos (CAD).
- Capa de datos.

Es importante señalar que la capa de datos de todos los módulos se desarrolló por un equipo común y que esta funciona como intermediaria entre los diferentes módulos, ya que estos pueden escribir y obtener datos siempre teniendo en cuenta la seguridad. A continuación se analiza la arquitectura del modelo y de cada una de las capas.

2.2.1. Modelo basado en capas.

Como un ejemplo de la interacción entre estas capas véase la Figura 2.1. Los tipos y clases comunes son todos los tipos de datos que deben dar servicios a varias capas, por tanto deben ser vistos por varias capas, por ejemplo las clases usadas de la biblioteca de clases de .NET.

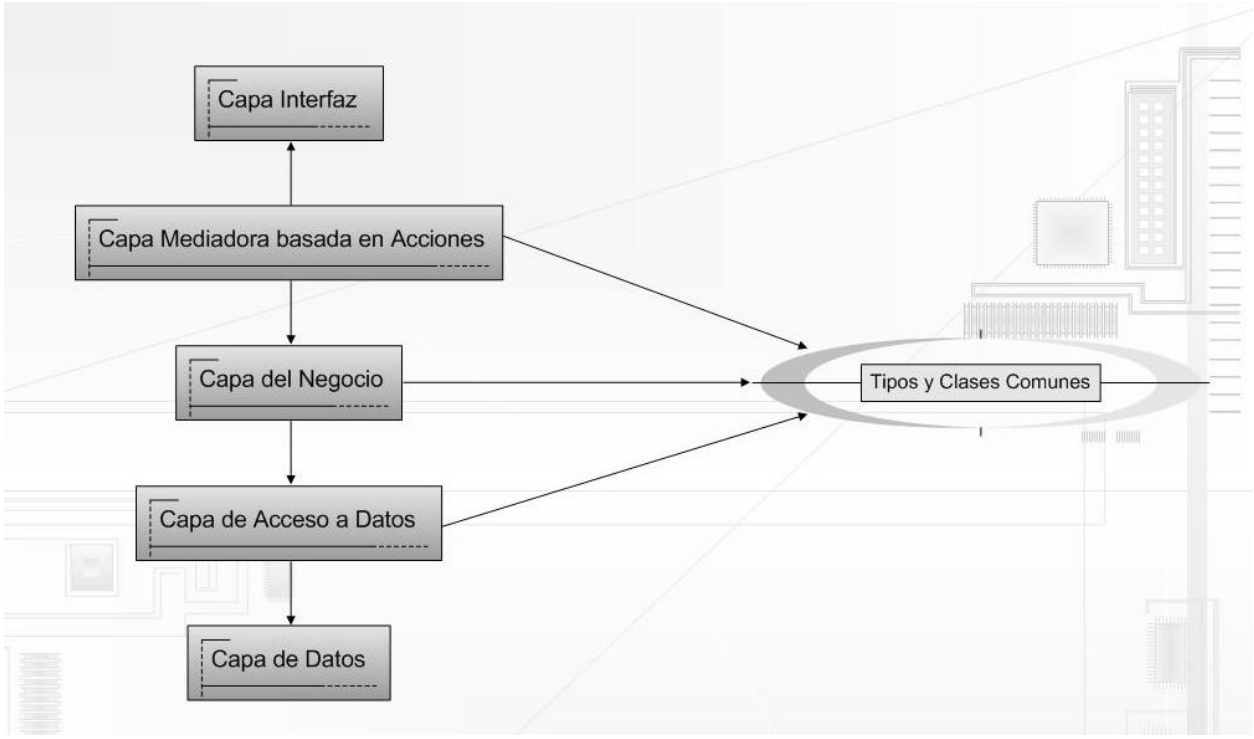

Figura 2.1 Estructura basada en capas. Los rectángulos representan las capas, la elipse representa una componente que contiene lo tipos y clases comunes a diferentes capas, el sentido de las flechas indica que la capa origen conoce la entidad que está en el destino de la misma.

2.2.1.1. *Capa Interfaz.*

La capa de interfaz responde a un comportamiento estándar, todos los módulos tienen prácticamente las mismas pautas de diseño con vistas a facilitar el trabajo con ellos y la estandarización del sistema completo.

Su uso se basa en la explotación del componente interfaz que realmente provee un *framework* facilitando el desarrollo del sistema. Este *framework* se basa en acciones y cada acción que se corresponda con funcionalidades de captura o visualización de datos tiene asociada un formulario cuya forma depende de la funcionalidad específica para la cual fue concebida dicha acción. El concepto de acción será tratado más adelante en la capa mediadora de interfaz basada en acciones.

El inicio del programa debe partir de la llamada del componente navegador como se muestra en Anexo I. El navegador dirigirá todo el flujo de trabajo a través de la aplicación.

La creación de una instancia del navegador tiene varios parámetros el primero corresponde con el nombre del navegador simple que se encuentra dentro del fichero de navegación. El navegador tiene un método IniciarTarea sobrecargado que provee dos formas de trabajar:

- En su primera variante recibe dos parámetros booleanos, el primero de ellos indica si se desea llamar directamente al método ejecutar de la acción cuando se crea y el segundo indica si se va a dar seguimiento de forma persistente a la acción, es decir, si se quiere guardar en la base de datos todas las acciones que realice el usuario que está trabajando en la aplicación.
- En su segunda variante recibe tres parámetros y se usa cuando se quiere iniciar una acción desde un estado previamente guardado en la base de datos. El primero de los parámetros indica el identificador de la acción en la base de datos y los otros dos parámetros son los mismos que en el caso anterior.

Por la importancia que cobra el componente navegador a lo largo del proyecto y el actual trabajo se realizará a continuación un análisis de este.

2.2.1.2. *Componente Navegador.*

El Componente Navegador se creó para facilitar y flexibilizar los flujos de trabajo dentro del proceso registral. Para el uso del navegador se debe usar el *namespace* Comun.FlujoTrabajo. Este componente es común a todos los módulos. Para su funcionamiento el navegador depende de un fichero XML llamado GrafoNavegacion.xml, el cual contiene como su nombre indica un grafo de navegación. La información interna de este fichero varía en correspondencia con la aplicación específica que lo está usando, o sea, cada uno de los módulos tiene su propio GrafoNavegacion.xml, aunque todos se llamen igual.

Dentro de las funcionalidades que encontramos en el componente navegador se encuentran:

- Abstraer el flujo de navegación de las interfaces: El flujo de navegación de una aplicación es una regla de negocio y como tal, no debería estar implementado específicamente dentro de la capa de presentación como pasa en muchas aplicaciones actuales. El componente le brinda flexibilidad y portabilidad a los módulos pues pueden realizar cambios en las reglas de navegación fácilmente.

- Facilita la reutilización del modelo de una aplicación en otra: Como ocurre en nuestro caso, el componente navegador puede ser utilizado en diferentes módulos y modificar el flujo de navegación para cada uno de estos resulta una tarea sencilla.
- Elimina de la interfaz la gestión del estado de la aplicación: Muchas veces se encuentran aplicaciones que gestionan el estado de la aplicación desde las propias interfaces de esta. Eso puede resultar un problema ya que si se hace un cambio en la interfaz, hay que trabajar nuevamente en esta gestión. El componente navegador brinda opciones para métodos que sirven precisamente para ocuparse del estado de la aplicación. Así las vistas, no tienen ninguna información sobre el estado de esta, sino que la obtienen del componente navegador.

En un mismo grafo de navegación hay varios “navegadores simples” cada uno de ellos generalmente corresponderá con un caso de uso del módulo específico con el que se está usando (Ver Anexo II, el cual representa un ejemplo del contenido del fichero XML del grafo de navegación que se muestra en la Figura 2.2).

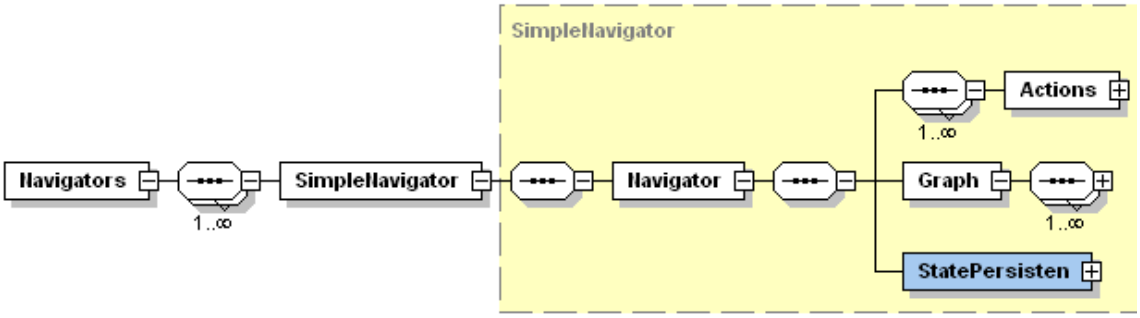

Figura 2.2 Esquema que representa el fichero XML utilizado para la gestión de la navegación.

Es responsabilidad de los equipos de cada módulo desarrollar el fichero XML que representa el grafo de navegación correspondiente a su módulo. Se requiere que todos los grafos de navegación correspondientes al momento inicial de la aplicación tengan como acción inicial “AccionIniciarAplicacion” como se muestra a continuación:

```
<Navigator StartAction = "AccionIniciarAplicacion">
```

El manejo de las transiciones entre formas y acciones se rige por un modelo híbrido que se basa en el uso de los grafos de navegación y en la llamada directa de una acción por parte de otra.

Se propone usar los grafos de navegación en aquellos casos que cumplan las siguientes propiedades:

- En todos los flujos de trabajo del negocio teniendo en cuenta que estos pueden variar y que conviene establecer estados que persisten y teniendo precaución con el pase de parámetros en las transiciones.
- En los procesos de gestión de información que constituyan procesos largos y que su gestión a través del grafo permita una mejor visión del flujo de trabajo.
- En los procesos de gestión de información insertar, actualizar, mostrar que no representen flujos del negocio. Por ejemplo, en la creación de oficinas, etc.

2.2.1.3. *Capa mediadora interfaz basada en acciones.*

Su uso se basa en la explotación del componente interfaz que realmente provee un *framework* facilitando el desarrollo del sistema basado en acciones.

Cada acción debe tener sentido semántico propio y completo. También deben ser atómicas. Las acciones básicamente definen un bloque de código reusable por varias operaciones sobre el sistema. Estas pueden estar asociadas a vistas del sistema y pueden representar un estado del sistema que puede o no persistir.

Una acción es una clase (véase Anexo III) que representa precisamente la ejecución de alguna tarea concreta. Estas tareas no deben ser excesivamente complejas y la clase debe:

- Heredar de la clase Accion o AccionSegura.
- Opcionalmente tener una forma visual asociada.
- Propiedades en función del objetivo específico de la misma.
- Se le debe reescribir el método CrearForma con vistas a mostrar lo que se desee.
- Usar el *namespace* Comun.FlujoTrabajo.Navegador y llamar adecuadamente a Comun.FlujoTrabajo.Navegador.XmlNavigatorGraph como se muestra en btnSiguiente_Click en el Anexo III.
- En caso de que la acción tenga asignada una forma visual, se deben programar dentro de la misma todos los eventos que puedan ocurrir a partir de la interacción con la forma visual.

2.2.1.4. *Capa lógica del negocio.*

La capa lógica del negocio se encuentra compuesta por las entidades del negocio, que representan objetos que son manejados por la aplicación. Esta capa es muy importante para la comunicación o medio de transporte entre las demás capas para llevar y traer datos.

Esta capa maneja 3 tipos de clases fundamentalmente, las cuales se muestran a continuación:

- Entidades: Son las encargadas de almacenar las entidades del negocio. Se puede ver un ejemplo de una clase Entidad en el Anexo IV.
- Colecciones: Tienen la responsabilidad gestionar una o más clases entidades. Se puede ver un ejemplo de una clase Colección en el Anexo V.
- Persistidor: Esta clase es la encargada de la persistencia de los objetos del negocio.

Es importante destacar que en la capa también se pueden encontrar clases que respondan a las reglas del negocio, cómo pudieran ser las clases que están relacionadas con el cálculo del trámite que no necesariamente está incluida en el grupo anteriormente mencionado.

2.2.1.5. *Capa de acceso a datos.*

La capa de acceso a datos es generada con el paquete TierDeveloper versión 4.0 y garantiza todo el manejo de la información que requiera el negocio del problema en cuestión y la conexión con la base de datos.

Se garantiza que el TierDeveloper genera toda la documentación apropiadamente, que se expliquen todos los métodos y las clases que se generan automáticamente.

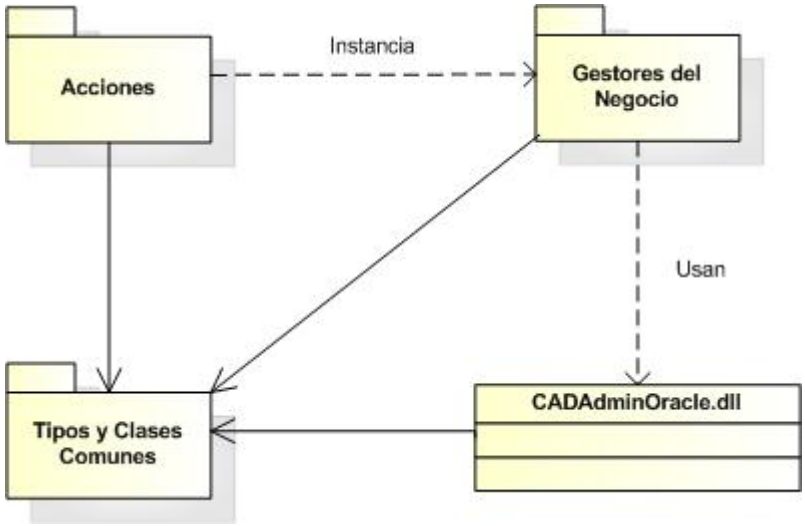

Figura 2.3 La capa de acceso a datos está representada como el componente CADAdminOracle.dll.

La comunicación entre el equipo de desarrollo de la capa datos y de los equipos correspondientes a cada módulo se basa en los artefactos de RUP, además de una estrecha comunicación entre los analistas.

2.2.1.6. *Capa de datos.*

Esta capa corresponde a los almacenes de datos. A ella pertenecen las bases de datos disponibles en los servidores de bases de datos. Esta capa es única y común a todos los módulos del sistema general.

La capa de datos se caracteriza porque el modelo de datos está en al menos tercera forma normal.

Las réplicas entre los servidores de datos tienen lugar de dos formas diferentes. Una forma fundamentalmente orientada a la réplica de los datos entre el servidor central y las oficinas, y se hace usando los propios mecanismos de replicación implementados en Oracle. El otro mecanismo cumple como objetivo fundamental gestionar información por parte de los clientes y de la granja de servidores, esta información puede estar contenida en el servidor de datos Oracle.

2.2.2. Breve análisis del Framework.

La arquitectura utilizada en el desarrollo del sistema de software del Proyecto de Registros y Notarías de la República Bolivariana de Venezuela ha sido protagonista en el esfuerzo por cumplir las metas impuestas por el cronograma. Es válido recordar que para la primera iteración se contaba con un plazo no superior a 30 días y la implementación de casi la totalidad de los casos de uso del sistema. El poco tiempo para concebir la arquitectura forzó a que se tomaran decisiones acertadas y desacertadas, con las cuales ha convivido y corregido en otros casos el equipo de trabajo durante todo el proceso. Una de las medidas iniciales fue la inclusión de un *framework* de trabajo, que facilitaba a los desarrolladores el arduo trabajo de crear una base para la implementación del sistema. Es importante destacar que este framework había comenzado su desarrollo dentro del marco de trabajo de otro proyecto de software con características arquitectónicas similares. A continuación se muestran algunas características que han sido de utilidad con la inclusión del *framework* en el transcurso del trabajo:

- El *framework* responde a patrones bien definidos los cuales brindan una mejor comprensión.
- Presenta un alto nivel de flexibilidad y reusabilidad, ya que este *framework* fue adoptado en el actual proyecto, y no se ha hecho necesario hacer grandes cambios para ajustarla a los objetivos del Proyecto de los Registros y Notarías.
- Ofrece un alto nivel de organización y guía a los programadores hacia un código con buenas prácticas.

Como el *framework* inicialmente no respondía a todas las características que se buscaban en los objetivos del Proyecto de Registros y Notarías, el equipo de desarrollo se vio forzado a incluirles o mejorar funcionalidades que cubrieran sus expectativas:

- Se incluyó el componente navegador, el cual permitió de una manera sencilla la navegación a través de las acciones.
- Se mejoró el chequeo y tratamiento de excepciones, con lo que se organizó de una forma más eficiente el tratamiento de estas.
- Se crearon opciones para la gestión de la sesión vigente en el sistema y la configuración local de las oficinas.

- Se mejoró la seguridad para una mayor protección de los datos creando canales más seguros y confiables.
- Se creó una funcionalidad para el chequeo de la versión del *framework*, el cual ha sido muy útil a la hora de desplegar nuevas versiones en los diferentes escenarios.

Además de las funcionalidades agregadas y antes mencionadas se piensa que en versiones posteriores el *framework* pudiera mejorarse ya que todavía presenta algunos puntos débiles, los cuales se muestran a continuación:

- Presenta una interfaz rígida y estilo web, lo cual le hace perder al sistema las posibilidades que ofrece las aplicaciones de escritorio.
- Muestra todavía algunas dificultades con el manejo de excepciones, pues no trata de la manera más óptima el lanzamiento de las excepciones desde controles.
- Exhibe cierta dependencia de la base de datos, pues por ejemplo la autenticación no permite hacerla contra un dominio o un Directorio Activo existente y para realizar un cambio como este se modificaría bastante.

2.3. Análisis de soluciones no triviales.

La palabra "análisis" proviene del griego y significa "disgregar, descomponer", sin embargo el uso del análisis está dado desde mucho antes a los griegos. Según la Biblia el primer analista de la historia fue José, el cual fue capaz de analizar e interpretar los sueños de Faraón. Por esta tarea fue recompensado y reconocido por toda la tierra de Egipto.

Desde José hasta la actualidad, el análisis ha abarcado muchas ramas de la ciencia y se ha perfeccionado para un mejor desarrollo de las distintas ramas. La informática no se ha quedado exenta de esto, y el análisis en la producción del software cumple una tarea fundamental para la correcta construcción de los programas informáticos.

La informática es una actividad intelectual y creativa que requiere de un elevado nivel de interpretación del entorno debido a que se dedica en gran medida a mejorar la manera de realizar los procesos. El Proyecto de Registros y Notarías no es la excepción, ya que se presentan diferentes situaciones que necesitan de una mirada interpretativa y de un análisis previo que permita implementar las soluciones prácticas.

A continuación se le realizará un análisis a los elementos más importante que se presentaron dentro de la solución de la Gestión Documental en el sistema.

2.3.1. Componente de Escaneo.

Una de las tareas más importantes dentro del sistema sin dudas es la gestión de los documentos que se digitalizan. La Dirección del Proyecto optó inicialmente por la utilización del escáner modelo HP Scanjet 5590, pero por cuestiones de agilizar el Proceso Registral en los Registros se decidió cambiar al HP ScanJet 8390 para el cumplimiento de esta tarea, pues el anterior no permitía escanear por ambas caras los documentos. En la actualidad la mayoría de los fabricantes de periféricos proveen de los drivers necesarios para la comunicación con los dispositivos y brindan un software para la realización de las funciones correspondientes de los mismos, pero como una de las metas de nuestro proyecto es la integración, el equipo de desarrollo se dio a la tarea de desarrollar un componente de escaneo para ser embebido dentro del sistema, con el objetivo de facilitar la Gestión Documental en los módulos de Registros Públicos y Registros Mercantiles. Las funcionalidades principales con las que cuenta el componente son las siguientes:

- Visualización de documentos.
- Escaneo de documentos.
- Impresión de documentos.
- Gestión de documentos.

El componente fue desarrollado en .NET por convenio de la arquitectura y para lograr una mejor integración con el sistema, además que no sería conveniente salir de una aplicación para dirigirse a otra interfaz con pautas de diseño completamente diferentes para realizar una función que se pudiera añadir al sistema. Al escanear el documento se utiliza la librería LeadTools, la cual posee todas las funcionalidades necesarias para su interacción con el escáner y gestión de las imágenes adquiridas. Es válido aclarar que las librerías de la Leadtools hacen uso de los drivers TWAIN, los cuales son los encargados de establecer conexión con el dispositivo de escaneo.

Es importante destacar que la librería Leadtools, es líder mundial en el tratamiento de imágenes y que empresas de gran envergadura como Microsoft, Hewlett Packard, Intel, Boeing, Xerox, Kodak, Ford Motor Company, entre otras, lo usan en sus sistemas. También es importante señalar que el

equipo de desarrollo ha recibido un correcto soporte en cuanto a las dudas que han surgido con el uso de la librería, en especial con la liberación de memoria de los objetos RasterImage, los cuales el recolector de basura de .NET Framework 1.1 no libera la memoria automáticamente.

Los documentos digitalizados se almacenan en la base de datos y sumados a que en un Registro se tramitan 100 trámites promedios por día y que en cada trámite se digitalizarán aproximadamente 30 folios entre distintos documentos, las imágenes en poco tiempo alcanzarán un gran volumen de datos. Por esta razón se necesita una forma económica de almacenar los datos, por lo que se escogieron los siguientes parámetros para la utilización del componente de escaneo en el sistema:

- Las imágenes escaneadas son almacenadas en formato TIFF, ya que este formato permite el paginado.
- Las imágenes son escaneadas en blanco y negro, ya que el peso de la imagen debe ser mínimo.
- Para salvarlas se le aplica el formato de compresión “tifccittgroup4” a 1 bit por pixel. Los formatos de compresión son algoritmos que se utilizan para la disminución de espacio de las imágenes. En este caso el “tifccittgroup4” es el que más comprime las imágenes TIFF en cuanto a espacio en disco con una calidad aceptable para los propósitos del sistema.

Como resultado de la utilización de los parámetros descritos anteriormente se obtuvo una imagen mínima con un peso aproximadamente de 35 KB por folio escaneado.

Es de suma importancia mencionar que todos los parámetros mencionados anteriormente son configurables y se encuentran guardados en la base de datos, los cuales pueden ser gestionados desde la administración de las oficinas.

El componente de escaneo se utiliza en el formulario frmDigitalizar, el cual cumple un papel protagónico en el caso de uso Digitalizar Documento que se mostrará en el capítulo 3. Este caso de uso cobra vital importancia durante todo el proceso, pues en el transcurso del proceso registral se puede digitalizar documentos en diferentes ocasiones. Además este caso de uso se utiliza también cuando se desea visualizar exclusivamente los documentos escaneados, por lo que tiene una gran utilidad dentro del sistema.

2.3.2. Generador Dinámico de Interfaces.

Dentro del sistema registral existen varios puntos donde la recogida de los datos pueda variar con el cambio de leyes. Para esto se necesitaba entonces una aplicación que fuera lo suficientemente flexible para soportar estos tipos de cambios. Existen dos lugares en el sistema registral donde más se ven influenciados por estos cambios y son el cálculo de los tributos relacionados al trámite y la recogida de los recaudos.

En el caso del cálculo del trámite, los conceptos del trámite puede cambiar con el cambio de las leyes, y esto sería caótico para el sistema si no se previeran estos cambios. En el caso de los recaudos y es común con los conceptos que se calculan, existen una gran cantidad, por lo cual no se les puede dar un trato diferenciado a cada uno de ellos, además de poder sufrir cambios repentinos en su estructura.

Por esta razón fue necesario pensar en una forma donde se pudieran realizar el cálculo y la gestión de los recaudos de una forma común, flexible y reusable, incluso para su utilización en los módulos de Registros Públicos y Registros Mercantiles.

La solución que se le dio a esta problemática fue la creación de un componente el cual fuera capaz de generar de manera dinámica un formulario de acuerdo a los tipos de datos que se le pasaran. En este componente recaería la responsabilidad de mostrar tanto los atributos de los recaudos como los conceptos del trámite.

En el *namespace* UIControls se encuentran los enumeradores y clases necesarias para el correcto del componente. A continuación se muestran los enumeradores seguidos de las clases:

Enumeradores

- TipoString: Se utiliza para saber si el tipo de datos *string* va a ser corto, largo o una identificación. Al referirse a corto se refiere a nombres y otras cadenas que no necesitan gran cantidad de caracteres, pues se les da un tratamiento diferente aunque sean el mismo tipo de datos. En cuanto a largo se consideran las direcciones, las cuales pueden tener una extensión considerable. La identificación se utiliza cuando se quiere hacer referencia a una cédula o pasaporte.

- CompareOperator: Dentro del componente existen opciones de validación, para lo cual el actual enumerador es muy útil, pues permite definir que operador se ajusta más al control que se quiere validar, si debe ser igual, menor mayor, entre otros.
- DataType: Este enumerador se refiere a los tipos de datos que se pueden validar dentro del componente dentro los que se encuentran *string*, *byte*, *int*, entre otros.

Clases

- DataItemArray: Esta clase se utiliza para el manejo de colecciones que se representarían como *listbox* o *combobox* en el componente. A la hora de la creación de este tipo de control se puede especificar que tratamiento darle, si puede ser múltiple selección o simple selección.
- Control: Esta es la clase más importante para el uso del componente, pues es la encargada de recoger el tipo de dato o valor del control se desea crear para visualizar, además de recoger todos los atributos que se consideren necesario para su manejo como puede ser la propiedad de sólo lectura por poner un ejemplo. Es importante destacar que esta clase tiene una propiedad el cual devuelve el control visual (*textbox*, *combobox*, *listbox*, etc) que se genera con la creación del control y una propiedad que admite un *object* el cual puede ser útil para guardar algún dato necesario, en nuestro caso lo usamos para guardar objeto del negocio que este relacionado con el control creando así una relación entre ellos.
- ParametroValidador: Una de las opciones que tiene la clase Control es la de adjuntarle su posible validación, para esto se puede utilizar esta clase. También se puede utilizar alguna de las clases hijas de esta las, cuales son más específicas en cuanto a la validación. Estas son:
    - ParametrosValidadorInt32: Se utiliza para la validación de tipos de datos *Int32*.
    - ParametrosValidadorInt64: Se utiliza para la validación de tipos de datos *Int64*.
    - ParametrosValidadorDecimal: Se utiliza para la validación de tipos de datos *Decimal*.
    - ParametrosValidadorString: Se utiliza para la validación de tipos de datos *String*.
    - ParametrosValidadorIdentificacion: Se utiliza para la validación del control que se utiliza para la recogida de la cédula o el pasaporte.

En los constructores y métodos de las clases antes mencionadas es que se utilizan todo los enumeradores que mencionamos anteriormente.

El componente utiliza los siguientes tipos de datos:

- *Int32*.
- *Int64*.
- *Decimal*.
- *String*.
- *Bool*.
- *DataTable*.
- *DateTime*.

Anteriormente se ha abordado los enumeradores y clases necesarias para la utilización del componente, pero no se ha tocado todavía como es el manejo con este. El Generador Dinámico de Interfaces posee una sencilla utilización y esto se ve representado en los métodos que posee para su uso, los cuales se muestran a continuación:

- AdicionarControl: A través de este método son añadidos los controles creados con la clase Control del *namespace* UIControls. Este método se encuentra sobrecargado tanto para recibir un Control, un arreglo de Control o un ArrayList que este compuesto por Control.
- EliminarControles: Este método tiene la finalidad de eliminar todos los controles que se han adicionado al componente, es decir funcionaría como un limpiador del componente.

Estos son los métodos necesarios para trabajar con el componente. Es de suma importancia mencionar que el componente posee un evento onCambio el cual se dispara cuando se modifica el valor de algún control. Este evento es de primordial importancia en el cálculo del trámite, pues cuando cambia un valor en uno de los controles, el cual sería algún concepto, este afectaría a otros conceptos o controles que dependan de este.

Durante la realización del componente se utilizó el patrón de diseño Estrategia (*Strategy*, en inglés) (Gamma, et al., 1994). Uno de los principales motivos que llevó al uso del patrón fue la existencia de un comportamiento diferente en cada uno de los controles aunque estos fueran tratados de igual manera por el componente. En otras palabras todos los controles tienen las mismas funcionalidades: pintarse, actualizar sus valores, crear su control visual, entre otras; pero todas estas funcionalidades se comportan de manera diferente dependiendo del tipo de control que se trate. La intención del patrón en nuestro trabajo es el de definir, encapsular e intercambiar los algoritmos que varíen

independiente al control que se trate. Gracias a esto hemos podido incluir en el componente controles nuevos de manera sencilla, y se ha evidenciado claramente la flexibilidad del patrón.

El componente se utiliza en los siguientes formularios:

- frmEditorDinamico.
- frmEditorDinamicoConcepto.
- frmEditorValorRecaudosBase.
- frmVisualizarRecaudo.

De los anteriores formularios los dos primeros están relacionados con el cálculo del trámite y los dos últimos pertenecen a la gestión de los recaudos. En el capítulo 3 se detallará sobre el caso de uso Gestionar Recaudos.

2.3.3. Componente para la Firma Digital.

Uno de las mayores metas del Proyecto de Registros y Notarías sin dudas es garantizar la mejor seguridad jurídica posible. Para alcanzar un mayor grado de excelencia en nuestro trabajo se decidió incluir la Firma Digital. En base a esto al equipo de desarrollo se le dio la tarea de la realización de un componente que fuera capaz de incorporar la Firma Digital en los documentos digitalizados tanto para los módulos de Registros Públicos como Registros Mercantiles.

Al igual que con el Componente de Escaneo, el de la Firma Digital se hubiera podido realizar independientemente del sistema que proponemos, pero esto implicaría exportar de la base de datos los documentos digitales, firmar digitalmente los documentos en algún programa completamente distinto a nuestra aplicación y después retornarlo a la misma posición dentro de la base de datos, lo que implicaría un proceso muy engorroso para los funcionarios que utilicen el sistema.

Para una mayor comprensión del tratamiento a la Gestión Documental se describirá a continuación la forma en que son tratados los documentos.

El documento digital es almacenado en la base de datos como una imagen. El formato de esta imagen es TIFF y es utilizado por la fácil utilización de la paginación de la imagen. Todos los documentos que son digitalizados, los cuales pueden ser el Documento Original y los Recaudos, son

situados consecutivamente en la misma imagen, empezando por el Documento Original. El orden quedaría dado como muestra la Figura 2.4.

Figura 2.4 Formato de salva del Documento Digital en la base de datos.

El Documento Digital mantiene el formato TIFF durante todo el proceso registral hasta el paso de Otorgamiento. En el paso de Otorgamiento se le realiza un cambio de formato a PDF para registrar la Firma Digital en el documento. Este es el paso elegido para el cambio de formato ya que después de este paso el Documento Digital no sufre más cambios, y la única operación que se puede realizar sobre el mismo es el de su visualización. Durante este cambio se utilizó la librería iTextSharp para la migración de un proyecto existente en Java a .NET, el cual permite llevar del formato TIFF a PDF.

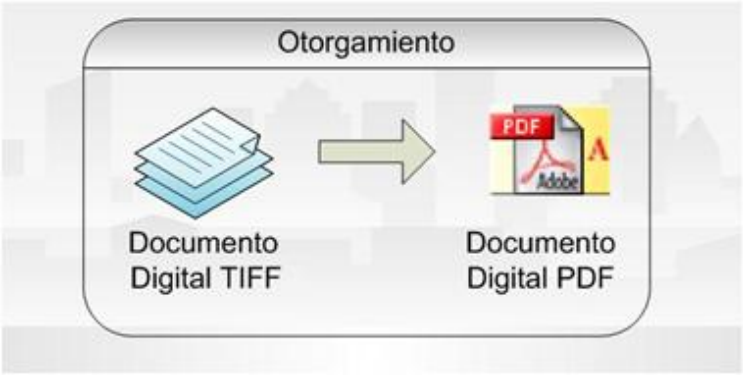

Figura 2.5 Cambio de formato de TIFF a PDF en el paso de Otorgamiento.

Después que el trámite es otorgado existe una opción en el sistema, el cual sólo muestra los trámites otorgados que no han sido firmados digitalmente. En esta opción se puede visualizar el documento y firmarlo, asegurando de esta forma una mayor seguridad jurídica.

El componente al igual que los anteriores fue desarrollado en .NET y está vez se utilizó la librería *SecureBlackBox* de EldoS, el cual permite firmar digitalmente un PDF. También en la construcción del componente se utilizó la referencia a objetos COM *Adobe Acrobat 8.0 Type Library*, el cual se adjunta al Visual Studio cuando se instala el Acrobat Reader 8.0. La referencia al Acrobat se utiliza para mostrar un visor que posee, el cual permite gestionar documentos PDF.

El componente se utiliza en el formulario frmDigitalizar coincidiendo con el componente de Escaneo, de los cuales se muestra uno u el otro dependiendo de la acción que se vaya a tomar a cabo, ya sea visualizar o digitalizar; y dependiendo también del objeto existente en ese momento durante el transcurso del trámite: TIFF o PDF. En este caso el formulario se encuentra dentro del caso de uso Firmar Documento Digital, que se describirá más detalladamente en el capítulo 3.

2.4. Conclusiones.

A lo largo del capítulo se abordaron diferentes aspectos que permiten una mejor comprensión de la base del sistema. Como resultado quedó definida la arquitectura que se utiliza y se realizó un análisis de las diferentes capas por las que está compuesta la arquitectura. También se abordó el *framework* utilizado como base para el desarrollo de la aplicación. Dentro de este, se mencionaron sus características, las funcionalidades agregadas por el equipo de desarrollo y las posibles mejoras para el futuro.

En el capítulo también se realizó un análisis a los elementos más importante que tuvieron lugar dentro de la solución de la Gestión Documental en el sistema. Los elementos analizados fueron el Componente de Escaneo, el Generador Dinámico de Interfaces y el Componente para la Firma Digital.

Con toda certeza los objetivos planteados para la realización del presente capítulo han sido cumplidos y como resultado se obtuvo un análisis de la base del sistema que ayuda para una mejor comprensión de la solución desarrollada.

Capítulo 3

3.1. Introducción.

En el siguiente capítulo se expone la solución técnica aplicada a la Gestión Documental para los Registros Públicos. Se presentan los diagramas de análisis para el diseño, los diagramas de clases, y el modelo de datos tanto lógico como físico. También se muestra el diagrama de componentes y el de despliegue. Finalmente se fundamentan los objetivos propuestos del capítulo.

3.2. Diagramas de Interacciones del Diseño.

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si se considera el "interior" del sistema, se tendrá algún objeto de diseño que recibe el mensaje del actor. Después el objeto de diseño llama a algún otro objeto, y de esta manera los objetos implicados interactúan para realizar y llevar a cabo el caso de uso. En el diseño, es preferible representar esto con diagramas de secuencia ya que el centro de atención principal es el encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

En algunos casos se incluyen subsistemas en los diagramas de secuencia para describir cuáles de ellos participan en una determinada realización de caso de uso, y quizás que interfaces intervienen de entre lo que proporcionan esos subsistemas. Gracias a ello, se puede diseñar los casos de uso a un nivel alto antes de que se hayan desarrollado los diseños internos de los subsistemas que intervienen. Esto es útil, por ejemplo, cuando se tiene que identificar las interfaces de los subsistemas en una fase temprana del ciclo de vida del software, antes de haber desarrollado el diseño interno.

En los diagramas de secuencia, se muestran las interacciones entre objetos mediante transferencias de mensajes entre objetos o subsistemas. Cuando se dice que un subsistema "recibe" un mensaje, se quiere decir en realidad, que es un objeto de una clase del subsistema el que envía el mensaje. El nombre del mensaje debería indicar una operación del objeto que recibe la invocación o de una interfaz que el objeto proporciona. (Jacobson, y otros, 2000)

En los Anexos VI, XVI y XXI se muestran los Diagramas de Secuencias del Diseño para los casos de uso Digitalizar Documento, Gestionar Recaudos y Firmar Documento Digital, respectivamente.

3.3. Diagramas de Clases del Diseño.

Una clase de diseño y sus objetos, y de ese modo también los subsistemas que contienen las clases de diseño, a menudo participan en varias realizaciones de casos de uso. También se puede dar el caso de algunas operaciones, atributos y asociaciones sobre una clase específica que son relevantes para sólo una realización de caso de uso. Esto es importante para coordinar todos los requisitos que diferentes realizaciones de casos de uso impone a una clase, a sus objetos y a los subsistemas que contiene. Para manejar todo esto, se utilizan diagramas de clases conectados a una realización de caso de uso, mostrando sus clases participantes, subsistemas y sus relaciones. De esta forma podemos guardar la pista de los elementos participantes en una realización del caso de uso. (Jacobson, y otros, 2000)

En los Anexos XXII, XXIII y XXIV se encuentran los Diagramas de Clases del Diseño para los casos de uso Digitalizar Documento, Gestionar Recaudos y Firmar Documento Digital, respectivamente.

3.4. Modelo de Datos.

Un modelo de datos es aquel que describe de una forma abstracta cómo se representan los datos, sea en una empresa, en un sistema de información o en un sistema de gestión de base de datos. Básicamente consiste en una descripción de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

Un modelo de datos consiste en:

- Objetos (entidades que existen y se manipulan).
- Atributos (Características básicas de estos objetos).
- Relaciones (forma en que se enlazan los distintos objetos entre si).

En los Anexos XXV y XXVI se muestran los Modelos de Datos Lógico y Físico respectivamente propuesto para el sistema.

3.5. Modelo de Implementación.

El modelo de implementación es una correspondencia directa de los modelos de diseño y de despliegue. Cada subsistema de servicio del diseño normalmente acaba siendo un componente por cada tipo del nodo en el que deba instalarse – pero no siempre así -. A veces el mismo componente puede instanciarse y ejecutarse sobre varios nodos. Hay lenguajes que proporcionan construcciones para el empaquetado de los componentes. En otros casos, las clases se organizan en ficheros con el código que representa los diferentes componentes. (Jacobson, y otros, 2000)

A continuación se muestran los Diagramas de Implementación para el sistema.

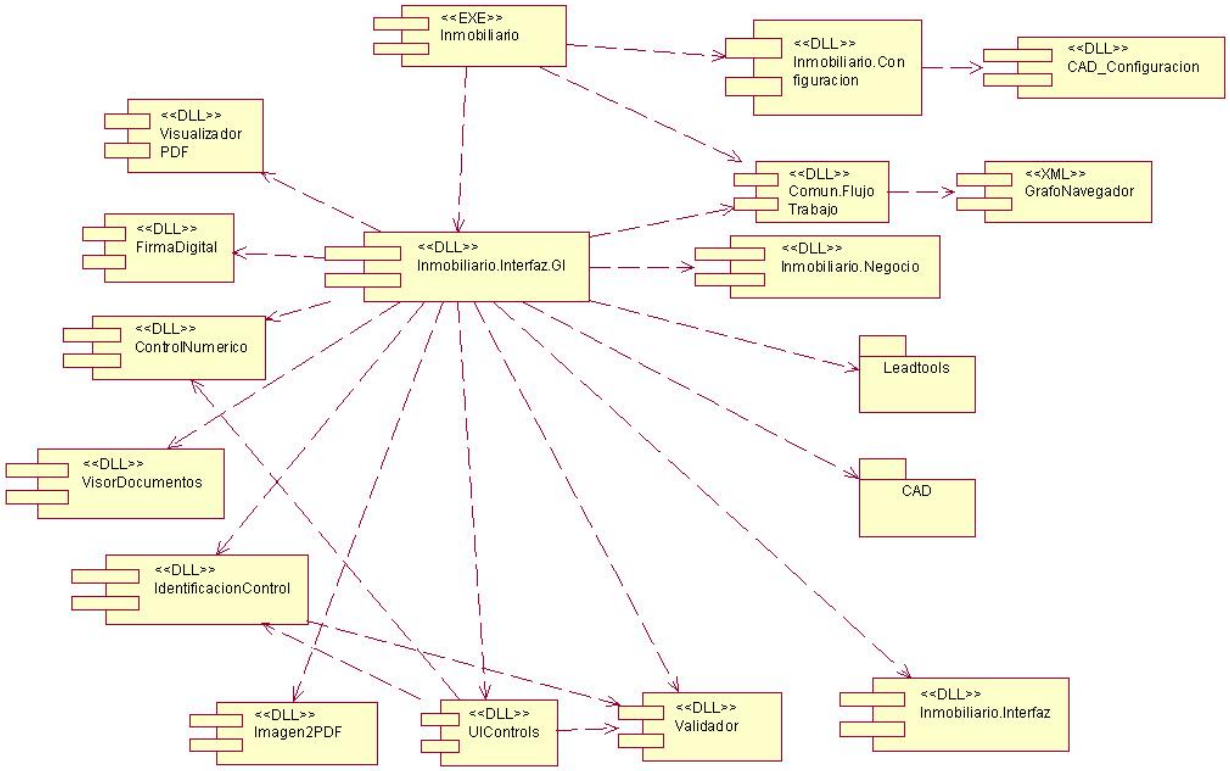

Figura 3.1 Diagrama de componentes del sistema.

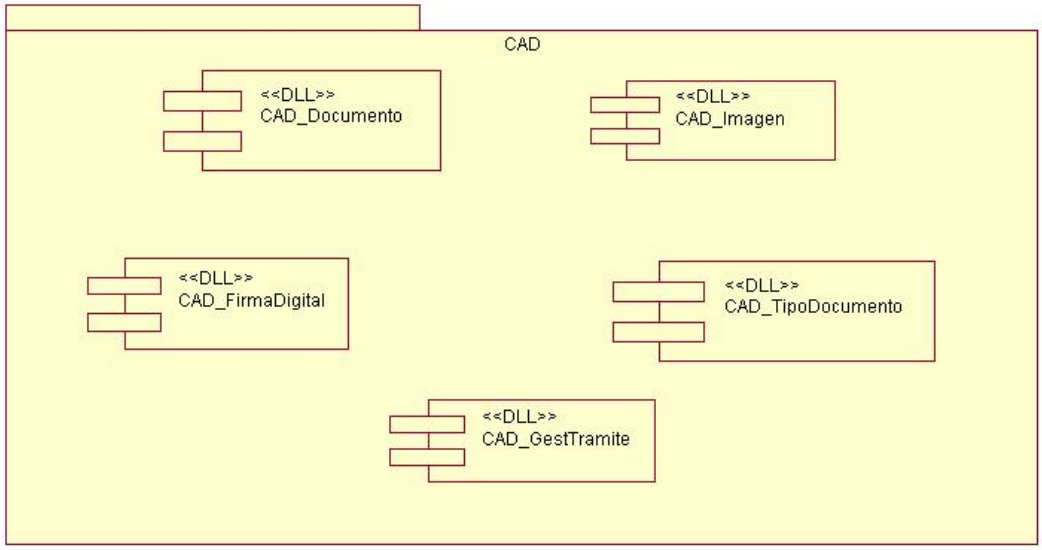

Figura 3.2 Diagrama de componentes para el paquete CAD del sistema.

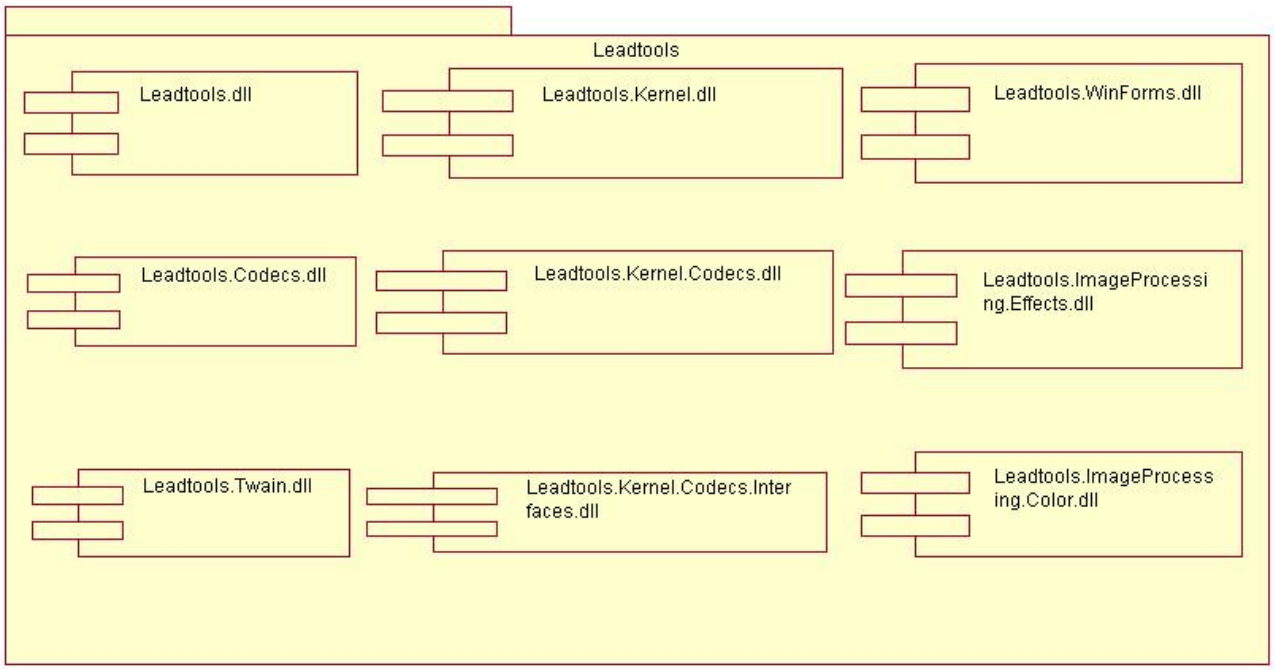

Figura 3.3 Diagrama de componentes para el paquete Leadtools del sistema.

3.6. Modelo de Despliegue.

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software. Con frecuencia se conoce cómo será la arquitectura física del sistema antes de comenzar su desarrollo. Por tanto, se puede modelar los nodos y las conexiones del modelo de despliegue tan pronto como comience el flujo de trabajo de los requisitos (Jacobson, y otros, 2000).

A continuación se presenta el Modelo de Despliegue propuesto para el sistema.

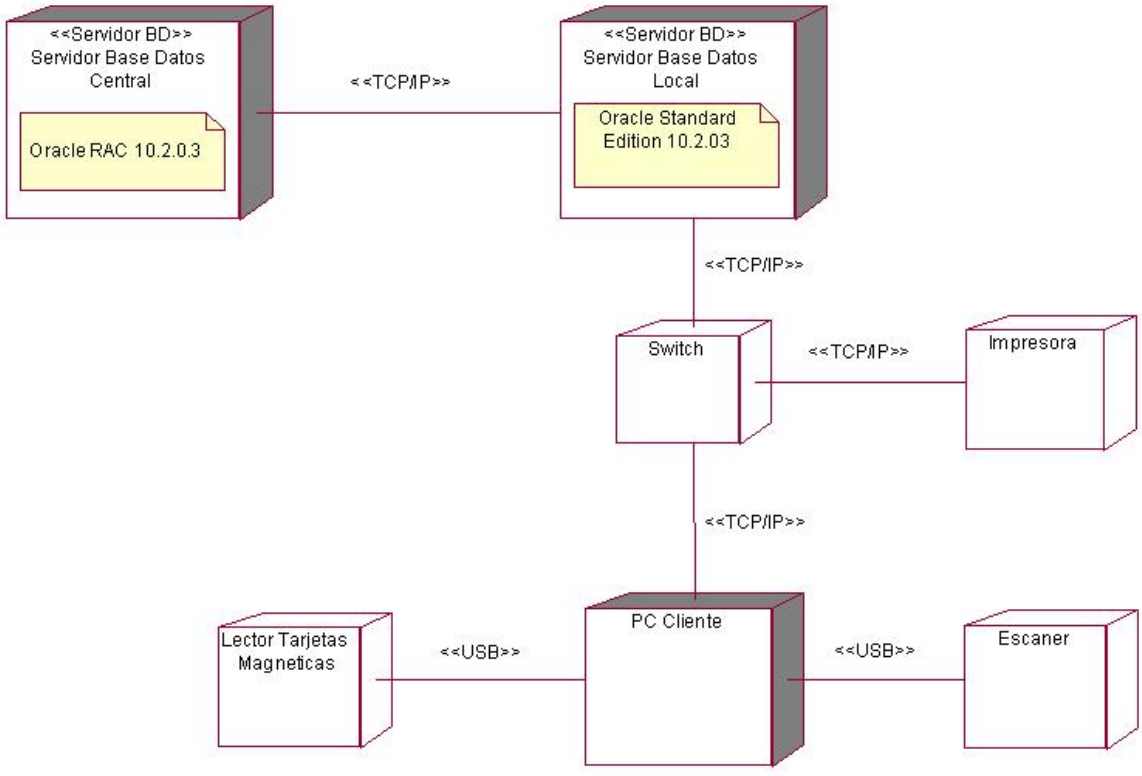

Figura 3.4 Modelo de Despliegue del sistema.

3.7. Conclusiones.

A lo largo del presente capítulo se mostraron los resultados de la etapa de diseño e implementación del sistema.

Se desarrollaron los Diagramas de Interacción de los tres casos de usos más importantes para el desarrollo de la Gestión Documental en el sistema. Con los resultados obtenidos de los Diagramas de Interacción se pudo llegar al Modelo de Clases del Diseño, donde se representaron las clases y sus asociaciones. Seguido se presenta el modelo de datos tanto lógico como físico para la Gestión Documental en el cual se sostiene la base de datos del sistema.

Con la culminación del Diseño se inició la Implementación con el Modelo de Implementación del sistema. Como colofón del capítulo se presentó el Modelo del Despliegue.

Los objetivos trazados para la realización del presente capítulo han sido cumplidos y como resultado se obtuvo la solución técnica aplicada a la Gestión Documental para los Registros Públicos

Conclusiones

Con la realización del presente trabajo se arriba a las siguientes conclusiones:

- Con el estudio del Proceso Registral, se logró identificar y dominar las actividades fundamentales que comprende la Gestión Documental, así como las reglas a las que está sujeta.
- Se logró un incremento de los niveles de Seguridad Jurídica a través del uso de las firmas digitales dentro del proceso de gestión de los documentos electrónicos.
- La publicidad registral como principio legal insoslayable alcanzará niveles nunca antes experimentados por la sociedad venezolana. El uso de la Gestión Documental y particularmente del componente de escaneo constituye el puente a la era digital dentro de un sector que por su carácter histórico demanda con urgencia de una evolución.
- Con los datos obtenidos en las oficinas el MPPRIJ obtendrá información más confiable, con la que podrá darle un mejor funcionamiento, organización y administración al Sistema de Registros y Notarias.
- Se fortaleció la Seguridad Jurídica y la Fe Pública con la creación e incremento del Archivo Digital.
- Se estandariza el Proceso Registral a todas las oficinas de Registros Públicos dándole un mejor cumplimiento al Principio de la Legalidad en la Ley de Registros y Notarías.
- Como resultado del Diseño del sistema se obtuvieron los Diagramas de Interacción, los Modelos de Clases y los Modelos de Datos tanto Lógico como Físico imprescindibles para el buen desarrollo de una solución informática con calidad.
- Con los resultados obtenidos del Diseño del sistema, se determinó el Modelo de Implementación y el Modelo de Despliegue.
- Como resultado general se obtuvo el Diseño e Implementación de una solución informática integrada para la Gestión Documental de los procesos que se realizan en los Registros Públicos.

Recomendaciones

Con la realización del presente trabajo se recomienda:

- Migrar la solución propuesta hacia alguna plataforma de software libre en cumplimiento del Decreto 3390. (Decreto No. 3390, 23 de diciembre del 2004) Se recomienda la utilización de las herramientas compatibles con el Proyecto Mono lo cual permitirá la reutilización de todo el código ya desarrollado para la capa de negocio de la presente solución el cual fue implementado teniendo en cuenta este objetivo.
- Incorporar en el sistema la funcionalidad de poder realizar el proceso registral a partir de un Documento Digital presentado por el ciudadano.
- Optimizar el manejo de las excepciones desde controles en el *framework*.
- Eliminar del framework las dependencias hacia la base de datos que puedan existir.
- Realizar un sistema de validación dinámico el cual permita un negocio más flexible.

Bibliografía y Reseñas Bibliográficas

Alachisoft. 2007. TierDeveloper 5.5 Overview. *www.alachisoft.com.* [Online] Alachisoft, 2007. [Cited: Marzo 7, 2007.] http://www.alachisoft.com/tdev/overview.html.

Alexander, Christopher. 1979. *The Timeless Way of Building.* s.l. : Oxford University Press, 1979.

Bailón Cabrera, Lorenzo. 2005. El Registro Público de la Propiedad en el Estado de Jalisco. *Instituto de Investigaciones Juridicas.* [Online] Diciembre 2005. [Cited: Marzo 26, 2007.] http://www.juridicas.unam.mx/publica/librev/rev/podium/cont/32/pr/pr39.pdf.

1990. Biblia. *La Santa Sede.* [Online] 1990. [Cited: Abril 15, 2007.] http://www.vatican.va/archive/ESL0506/_INDEX.HTM.

Booch, Grady. 1994. *Object-Oriented Analysis and Design.* California : Addison Wesley, 1994.

Cañedo Andalia, Rubén. 2004. De la piedra al web: análisis de la evolución histórica y del estado actual de la actividad bibliológico-informacional. *BVS - Biblioteca Virtual de Salud en Cuba.* [En línea] 20 de Enero de 2004. [Citado el: 7 de Marzo de 2007.] http://bvs.sld.cu/revistas/aci/vol12_1_04/aci04104.htm.

Decreto No. 3390. **Chavéz Frías, Hugo Rafael. 2004.** Caracas : Gaceta Oficial, 2004.

Fowler, Martin. 1997. *Analysis Patterns: Reusable Object Models.* s.l. : Adisson Wesley, 1997.

Freitas Alvarado, Pilar and Delgado Scheelje, Alvaro. 2005. Modernización de los Registros Públicos y políticas de protección de datos. *Centro Internacional de Derecho Registral.* [Online] Mayo 2005. [Cited: Marzo 26, 2007.] http://www.cinder.info/files/Delgado%20-%20Protecci%F3n%20de%20datos.pdf.

Gamma, Erich, et al. 1994. *Elements of Reusable Object-Oriented Software.* s.l. : Addison Wesley, 1994.

Gorospe, Jesús. Ibermática. 2005. *Gestión Documental.* s.l. : Ibermática, 2005.

Hillside Group. 2003. Home of the Patterns Library. *Sitio Web de Hillside.net.* [Online] 2003. [Cited: Marzo 26, 2007.] http://hillside.net/.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000.

Lafaurie, José Félix. 2003. Superintendencia de Notariado y Registro. Proyecto de Modernización Tecnológica. *Sitio Web del Primer Congreso Iberoamericano de Registro y Propiedad.* [Online] Noviembre 2003. [Cited: Marzo 7, 2007.] http://www.sunarp.gob.pe/CongresoRegistral/ponencias/Lafaurie_PRESEN.pdf.

LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA. **Chávez Frías, Hugo Rafael. 22 de Diciembre del 2006.** Caracas : Gaceta Oficial, 22 de Diciembre del 2006.

Microsoft Inc. 2003. Desarrollador Cinco Estrellas. *Microsoft.com.* [Online] 2003. [Cited: Marzo 7, 2007.] www.microsoft.com/spanish/msdn/comunidad/dce/.

Montalván Pérrigo, Hermann. 2003. Interconexión Nacional de los Registros Públicos del Perú en el marco del Proceso de Modernización Tecnológica. *Sitio Web del Primer Congreso Iberoamericano de Registro y Propiedad.* [Online] Noviembre 2003. [Cited: Marzo 26, 2007.] http://www.sunarp.gob.pe/CongresoRegistral/ponencias/ponencia%20GG%20Finalisimo2.pdf.

Morales Galito, Einstein Alejandro. 2005. Monografias.com. *Monografias.com.* [En línea] 28 de Mayo de 2005. [Citado el: 25 de Febrero de 2007.] http://www.monografias.com/trabajos22/derecho-registral/derecho-registral.shtml.

Oracle. 2005. *Oracle Database Documentation Library.* s.l. : Oracle, 2005.

Glosario de Términos

A

Abogado Redactor

El que redacta el documento que se inscribirá en el registro.

Abogado Revisor

Su función consiste en realizar una revisión minuciosa de los documentos, analizando, verificando y confrontando con el o los títulos correspondientes, para determinar que el instrumento reúne todos los requisitos que permitan realizar la operación.

Asiento Registral

Un Documento luego de ser inscrito es archivado o protocolizado ubicándose en Protocolo, Tomo, Folio, lo cual constituye un Asiento Registral.

C

Cantón

Cada una de las divisiones administrativas del territorio de ciertos Estados, como Suiza, Francia y algunos americanos.

D

Derechos Especiales

Valor fijo de que se paga por ciertas operaciones contenidas en el documento.

Documento

Escrito en papel u otro tipo de soporte con que se prueba o acredita una cosa, como un título, una profesión, un contrato, etc.

E

Exención

Figura establecida en las Leyes de la República mediante la cual el estado otorga ventaja a los beneficiados en el no pago total o parcial de impuestos, tasas, o contribuciones.

F

Foliatura

Folio inicial y final desde y hasta donde se extiende la inscripción de un documento, comprobante o cualquier otro instrumento que debe asentarse en un tomo.

Folio

Página, hoja de un documento o libro donde este se asienta. Se numeran consecutivamente, lo cual sirve para referenciar dónde exactamente, en un tomo, está inscrito un documento.

Framework

Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Funcionario de Cálculo

Es quien recibe el documento y calcula el costo del trámite según las operaciones y características que se desprenden del mismo.

Funcionario de Notas

El funcionario que elabora las notas es el encargado de seleccionar entre la variedad de notas existentes, la que se adecue o corresponda a cada documento. Este funcionario debe estar capacitado y tener el conocimiento y la práctica suficiente que le permita colocar la nota que corresponda a cada documento, así como para analizar el documento, determinando los otorgantes que se colocarán en la nota y los recaudos respectivos.

Funcionario de Prohibiciones

Su función consiste en revisar cada uno de los documentos que ingresan a la Oficina de Registro, a fin de verificar si uno, varios o todos sus otorgantes tiene vigente algún impedimento judicial que le impida enajenar o gravar el inmueble, tales como prohibiciones de enajenar o gravar, embargos, secuestro u otras medidas cautelares. En caso afirmativo debe comunicarlo de inmediato al interesado a fin de que éste proceda a la solución del impedimento. Para realizar esta revisión el funcionario de prohibición cuenta con medios seguros, como sistemas automatizados y el Libro de Prohibiciones manuscrito, que está dispuesto en forma de índice.

G

Grid Computing

Es un nuevo paradigma de computación distribuida en el cual todos los recursos de un número indeterminado de computadoras son englobados para ser tratados como un único superordenador de manera transparente. Estas computadoras englobadas no están conectadas o enlazadas firmemente, es decir no tienen porque estar en el mismo lugar geográfico.

H

Habilitar

Pagar por realizar el otorgamiento en fecha anticipada a los 10 días que por defecto han de transcurrir entre la presentación y el otorgamiento.

K

KB

Un kilobyte es una unidad de medida común para la capacidad de memoria o almacenamiento de las computadoras. Es equivalente a 1024 bytes. Generalmente se abrevia como KB, K, kB, Kbyte o k-byte.

L

Libro de Control de Traslados

Donde se asientan las direcciones de los diferentes lugares donde ha sido otorgado un trámite fuera de la oficina, referenciadas con el número de control del trámite y con la fecha del otorgamiento.

Libro de Presentaciones

Libro donde se recogen consecutivamente a diario las notas de presentación.

Literate-programming

Filosofía de la programación basada en la premisa de que los programas deben ser escritos de forma amena para las personas como principal objetivo, como si fuera un trabajo de literatura.

N

Namespace

Esquema de nombres lógico para agrupar los tipos relacionados. .NET Framework utiliza un esquema de nombres jerárquico para agrupar los tipos en categorías lógicas de funcionalidad relacionada, como la tecnología ASP.NET o la funcionalidad de interacción remota. Las herramientas de diseño pueden utilizar *namespace* para que los programadores puedan examinar y hacer referencia más fácilmente a los tipos en el código. Un ensamblado individual puede contener tipos cuyos nombres jerárquicos tienen distintas raíces de *namespace* y una raíz de *namespace* lógico puede abarcar varios ensamblados. En .NET Framework, un *namespace* es una comodidad para la

nomenclatura lógica en tiempo de diseño, mientras que un ensamblado establece el ámbito de nombres para los tipos en tiempo de ejecución.

Nota de Presentación

Se emite en el acto de presentación, como resultado del mismo y contiene de cada documento que es presentado los siguientes datos: fecha, hora y minuto de la presentación del instrumento que se trate, el nombre del otorgante u otorgantes, y el de la persona o personas que aparezcan interesadas en el acto, y un extracto del documento presentado.

Número de Catastro

Número de identificación del inmueble, asignado y emitido por la oficina de catastro municipal, que debe encargarse de censar los inmuebles por zona geográfica.

Número de Control

Número que se le asigna a un trámite cuando se presenta y es su identificador durante todo el proceso de registro.

Número de Inpreabogado

Número único que identifica a cada abogado.

O

Oficio

Se refiere al documento enviado desde el tribunal donde vienen los datos de una suspensión o prohibición.

On-line

Online. Está conectado a una red.

Operación

Son todos los movimientos que se pueden realizar en relación con un inmueble, donde es este el centro de la operación que se realiza.

- ✓ Aclaratoria.
- ✓ Acta asamblea constitución aporte.
- ✓ Acta asamblea.
- ✓ Acta caja de ahorro.
- ✓ Acta constitutiva.
- ✓ Acta de entrega.
- ✓ Acta de remate. (con derecho especial)
- ✓ Acta de Remate. (con porcentaje)
- ✓ Adicional.
- ✓ Adjudicación.
- ✓ Adjudicación. (Exoneradas)
- ✓ Adopción.
- ✓ Anticresis.
- ✓ Aporte.
- ✓ Arrendamiento.
- ✓ Arrendamiento financiero.
- ✓ Autorización.
- ✓ Cancelación.
- ✓ Capitulación matrimonial.
- ✓ Certificación de gravamen.
- ✓ Cesión de crédito.
- ✓ Cesión de derecho.
- ✓ Cierre de titularidad.
- ✓ Cláusula penal.
- ✓ Comodato.
- ✓ Condominio.
- ✓ Constitución de hogar.

- ✓ Cooperativa.
- ✓ Copia certificada.
- ✓ Copia mecanografiada.
- ✓ Copia simple.
- ✓ Curatela.
- ✓ Dación en pago al mismo acreedor.
- ✓ Dación en pago.
- ✓ Declaratoria.
- ✓ Demanda.
- ✓ División de lotes.
- ✓ Emancipación.
- ✓ Extinción de fideicomiso.
- ✓ Fianza.
- ✓ Fideicomiso.
- ✓ Fundación.
- ✓ Fusión.
- ✓ Gastos de hipoteca.
- ✓ Hierros y señales.
- ✓ Hipoteca 2do Grado.
- ✓ Hipoteca 3er Grado.
- ✓ Hipoteca con comisión.
- ✓ Hipoteca legal sin intereses.
- ✓ Hipoteca inmobiliaria.
- ✓ Hipoteca.
- ✓ Integración de parcelas.
- ✓ Liberación.
- ✓ Liquidación de comunidad conyugal.
- ✓ Modificación de condominio de plazo sin intereses.
- ✓ Modificación de condominio.
- ✓ Modificación parcial.
- ✓ Nombramiento de tutor.

- ✓ Parcelamiento.
- ✓ Partición.
- ✓ Permuta.
- ✓ Poder.
- ✓ Préstamo quirografario sin gasto.
- ✓ Prórroga de plazo.
- ✓ Ratificación de hipoteca.
- ✓ Ratificación de lindero.
- ✓ Renta vitalicia.
- ✓ Renuncia de herencia.
- ✓ Renuncia de hipoteca legal.
- ✓ Renuncia de usufructo.
- ✓ Reparcelamiento.
- ✓ Resolución de compraventa.
- ✓ Revocatoria de curatela.
- ✓ Revocatoria de poder.
- ✓ Revocatoria de servidumbre.
- ✓ Revocatoria de testamento abierto.
- ✓ Revocatoria de testamento cerrado.
- ✓ Revocatoria de tutelas.
- ✓ Sentencia de divorcio.
- ✓ Sentencia firme de tribunales.
- ✓ Separación de cuerpos y bienes.
- ✓ Servidumbre gratuita.
- ✓ Servidumbre.
- ✓ Sociedad civil.
- ✓ Subfianza.
- ✓ Subrogación de hipoteca 1ro y 2do Grado.
- ✓ Título supletorio.
- ✓ Transacción y adjudicación.
- ✓ Transacción.

- ✓ Tutela.
- ✓ Usufructo gratuito.
- ✓ Usufructo.
- ✓ Venta (Art. 52) casa, quinta y terreno.
- ✓ Venta. (con valor estimado)
- ✓ Venta con pacto de retracto.
- ✓ Venta.

Otorgante

Persona involucrada en las operaciones que se realizan sobre un inmueble que están reflejadas en un documento.

P

Planilla de Precálculo o Planilla de Servicio Autónomo

Planilla emitida en el departamento de cálculo donde se asientan los conceptos por los cuales se debe pagar al servicio autónomo y el monto a pagar por cada uno, según el cálculo hecho al documento.

Planilla del SENIAT

Planilla emitida en el departamento de cálculo donde se asientan los conceptos por los que se debe pagar al SENIAT y el monto a pagar por cada uno, según el cálculo hecho al documento.

Presentación

Proceso mediante el cual el usuario ingresa su documento en el registro, para darle curso legal al trámite registral.

Presentante

Persona que presenta el documento.

Prohibición

Impedimento judicial, medida cautelar, que impide realizar una operación inmobiliaria, ya sea porque pese sobre la persona o sobre el inmueble.

Protocolo

Clasificación que se hace de los tomos según el tipo de operaciones que están inscritas en él.

R

Recaudos

Documentos, comprobantes, avales, certificaciones, constancias, etc. que deben acompañar a los documentos a la hora de presentarlos, para conferirle valor legal al proceso y respaldar las operaciones contenidas en el mismo.

RIF

Registro de Información Fiscal.

S

SENIAT

Servicio Nacional Integrado de Administración Tributaria.

Servicio Autónomo

Se refiere a la oficina de registro.

Solicitante

Persona que solicita el cálculo de los costos de un trámite.

Suspensión

Medida que levanta, anula, termina una prohibición existente.

T

Testigo Instrumental

La función consiste en llevar a cabo el otorgamiento de los documentos, tomando todas las previsiones necesarias. Todo testigo instrumental está obligado a colaborar con el departamento en la preparación que necesitan los documentos por lo menos con un día de anticipación al otorgamiento. Esta preparación consiste en revisar minuciosamente los documentos verificando que se hayan cumplido los pasos necesarios antes del otorgamiento con las firmas de los funcionarios respectivos, confrontación del original con sus copias, colocación de carátula y sellos respectivos.

TIFF

Tagged Image File Format - formato de fichero de imágenes etiquetado. Formato de fichero para el almacenamiento de imágenes creado a mediados de los 80. Originalmente fue desarrollado por la compañía Aldus (en conjunto con Microsoft) para ser usado en impresoras PostScript. TIFF es un formato popular para imágenes de color verdadero y es ampliamente soportado por las aplicaciones de manipulación de imágenes (como Photoshop, GIMP, Ulead PhotoImpact, Photo-Paint, Paint Shop Pro, entre otras), y por otros tipos de aplicaciones, e incluso por cámaras digitales. Actualmente es Adobe Systems la empresa encargada de controlar las especificaciones TIFF, aunque no ha tenido mayores modificaciones desde 1992. Los archivos TIFF suelen tener la extensión ".tiff" o ".tif". Las imágenes TIFF no pierden calidad en compresión.

Títulos Anteriores

Documentos registrados anteriores a un documento en específico, cuyo contenido es una operación sobre el mismo inmueble que tiene el documento.

Tomo

Libro físico, compuesto por varios folios, donde están inscritos y asentados los documentos, se numeran consecutivamente y también, al igual que los folios, sirven para referenciar dónde está inscrito el documento físicamente.

Tracto (Registral, Legal, Sucesivo)

Conjunto de todos los títulos, en orden cronológico en donde figura el inmueble y que han cambiado o se han modificado debido a las inscripciones posteriores a ellos, que han involucrado en sí al inmueble.

Traslado

Característica que tiene un trámite cuando ha sido solicitado que se constituya el acto de otorgamiento fuera de la oficina de registro.

TWAIN

Los drivers TWAIN son un estándar industrial de un protocolo de software y una API que permite una integración sencilla de datos de imágenes entre los dispositivos de entradas y aplicaciones de software.

Anexos

Anexo I. Clase inicial, nótese que en el main se llama al navegador.

```
using System;
using Sistema.Interfaz.GI;
using Sistema.Interfaz.Acciones;
using prototipo.Interfaz.GI;
using Sistema.Interfaz;
using Comun.FlujoTrabajo;
namespace prototipo
{
    /// <summary> Summary description for prototipo. /// </summary>
    public class prototipo
    {
        [STAThread]
        public static void Main()
        {
            FabricaAcciones.AgregarEnsambladoPorUnTipo(typeof(MiPropiaAccion));

            Comun.FlujoTrabajo.Navegador.XmlNavigatorGraph.Instancia("n2
Nombre").IniciarTarea(true, false);
        }
    }
}
```

Anexo II. Ejemplo de código XML del grafo de navegación.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2005 sp2 U (http://www.altova.com)-->
<Navigators xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\PruebaRN\Navigator2.xsd">
  <SimpleNavigator Nombre="n2">
    <Navigator StartAction="AccionIniciarAplicacion">
      <Actions Name="Mi Accion1">
        <Type>prototipo.Interfz.GI.MiPropiaAccion</Type>
        <Assembly>prototipo.Interfz.GI</Assembly>
      </Actions>
      <Actions Name="Mi Accion2">
        <Type>prototipo.Interfz.GI.MiPropiaAccion2</Type>
        <Assembly>prototipo.Interfz.GI</Assembly>
      </Actions>
      <Actions Name="AccionIniciarAplicacion">
        <Type>Sistema.Interfaz.GI.AccionIniciarAplicacion</Type>
        <Assembly>Comunes.Interfaz.GI</Assembly>
      </Actions>
      <Graph>
```

```
        <Node>
          <NavigateTo Action="Mi Accion2" Actual="Mi Accion1">
            <NavigationValue>Next</NavigationValue>
          </NavigateTo>
        </Node>
        <Node>
          <NavigateTo Action="Mi Accion1" Actual="Mi Accion2">
            <NavigationValue>Back</NavigationValue>
          </NavigateTo>
        </Node>
      </Graph>
      <StatePersisten Name="SQLStatePersistent">
        <Type>String</Type>
        <Assembly>String</Assembly>
      </StatePersisten>
    </Navigator>
  </SimpleNavigator>
  <SimpleNavigator Nombre="n1">
    <Navigator StartAction="String">
      <Actions Name="1">
        <Type>String</Type>
        <Assembly>String</Assembly>
      </Actions>
      <Actions Name="2">
        <Type>String</Type>
        <Assembly>String</Assembly>
      </Actions>
      <Graph>
        <Node>
          <NavigateTo Action="2" Actual="1">
            <NavigationValue>String</NavigationValue>
          </NavigateTo>
        </Node>
        <Node>
          <NavigateTo Action="1" Actual="2">
            <NavigationValue>String</NavigationValue>
          </NavigateTo>
        </Node>
      </Graph>
      <StatePersisten Name="String">
        <Type>String</Type>
        <Assembly>String</Assembly>
      </StatePersisten>
    </Navigator>
  </SimpleNavigator>
</Navigators>
```

Anexo III. Ejemplo de una clase acción.

```
using System;
using Sistema.Interfaz.Acciones;
using prototipo.Interfaz;
using Sistema.Interfaz;
using System.Windows.Forms;
using Comun.FlujoTrabajo.Navegador;
using System.Collections;
namespace prototipo.Interfz.GI
{
    public class accEjemplo : Accion, IRecibidorDeParametros
    {
        #region IRecibidorDeParametros Members
        private int parametro_1;
        private string parametro_2;

        public event Comun.FlujoTrabajo.Navegador.RecibirParametros onRecibir;

        public void Recibir(object parametro)
        {
            //Manipulacion de los parametros;
            parametro_1 = ((ArrayList)parametro)[0];
            parametro_2 = ((ArrayList)parametro)[1];
            onRecibir(parametro);
        }
        #endregion
        public accEjemplo(string text1, string text2)
            : base(text1, text2)
        {
        }
        protected override System.Windows.Forms.Form CrearForma()
        {
            prototipo.Interfaz.frmTest frm = new prototipo.Interfaz.frmTest();
            frm. btnSiguiente.Click += new EventHandler(btnSiguiente_Click);
            this.OnAccionEjecutar += new meAccion(accEjemplo_OnAccionEjecutar);
            return frm;
        }
        private void btnSiguiente_Click(object sender, EventArgs e)
        {
            Accion miAccion =
Comun.FlujoTrabajo.Navegador.XmlNavigatorGraph.Instancia("nombreNavegador").Navega
r("valor de navegacion", false, false);
            miAccion.OnAccionCulminada += new
meAccionCulminada(a_OnAccionCulminada);
            miAccion.Ejecutar();
        }
        private void a_OnAccionCulminada(Accion acc, object Resultado)
        {
            this.Culminar();
        }
```

```
        //Código obligatorio en las acciones que son instanciadas directamente
        //del menú principal para notificarle al navegador el valor actual en que
        //esta parado
        private void accEjemplo_OnAccionEjecutar(Accion acc)
        {
            XmlNavigatorGraph.Instancia("nombreNavegador").Estado.Actual =
this.Nombre;
        }
    } // llave fin de la clase
}
```

Anexo IV. Ejemplo de una clase Entidad en la capa lógica del negocio.

```
public class Oficina
{
    #region Atributos
    private decimal codigo;
    private string nombre;
    private string municipio;
    private string entFederal;
    private string direccion;
    private TipoOficina tipo;
    #endregion

    #region Constructor
    public Oficina(){}
    #endregion

    #region Propiedades
    public decimal Codigo
    {
        get { return codigo; }
        set { codigo = value; }
    }
    public string Nombre
    {
        get { return nombre; }
        set { nombre = value; }
    }
    public string Municipio
    {
        get { return municipio; }
        set { municipio = value; }
    }

    public string EntFederal
    {
        get { return entFederal; }
        set { entFederal = value; }
    }
```

```
    public string Direccion
    {
        get { return direccion; }
        set { direccion = value; }
    }
    public TipoOficina Tipo
    {
        get { return tipo; }
        set { tipo = value; }
    }
    public string StringTipoOficina
    {
        get { return Global.ObtenerDescripcionDB(Tipo); }
    }
    #endregion
}
```

Anexo V. Ejemplo de una clase Colección en la capa lógica del negocio.

```
using System;
using System.Collections;
using Inmobiliario.Negocio.Interfaces;

namespace Inmobiliario.Negocio.Colecciones
{
	/// <summary>
	/// Summary description for ColeccionFuncionarios.
	/// </summary>
	[Serializable]
	public class ColeccionOficina : CollectionBase
	{
		public ColeccionOficina()
		{
		}

		public virtual void Add(Oficina Elemento)
		{
			this.List.Add(Elemento);
		}

		public virtual void Remove(Oficina Elemento)
		{
			this.List.Remove(Elemento);
		}

		public virtual Oficina this[int Indice]
		{
			get
			{
				return (Oficina)this.List[Indice];
			}
		}
	}
}
```