

Universidad de las Ciencias Informáticas.

Facultad 3.



Ingeniería Informática.

Diseño e implementación de funcionalidades que se llevan a cabo en los registros mercantiles: solicitudes de expedientes, copias de documentos y sellado de libros.

Trabajo de diploma para optar por el título de Ingeniero Informático.

Autores:

Armando Esteban Pacheco Iglesias.

Alejandro Casanova Mutis.

Tutores:

Ing. Rodolfo Pérez Ávila.

Ing. Kiosmy Almenares Herrera.

Ciudad de la Habana, Junio del 2008.

DECLARACIÓN DE AUTORÍA.

Declaramos ser autores de la presente tesis y recomendamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Armando Esteban Pacheco Iglesias

Ing. Kiosmy Almenares Herrera.

Firma del Autor.

Firma del Tutor.

Alejandro Casanova Mutis

Ing. Rodolfo Pérez Ávila.

Firma del Autor.

Firma del Tutor.

DEDICATORIA.

A mi familia: mi mamá, por su gran apoyo en todo momento, por ser la guía e inspiración de toda mi vida, y por haberme soportado todas mis malcriadeces. A mi padre, por ser un ejemplo a seguir, por todos sus consejos, por haberme enseñado que la vida es sacrificio y esfuerzo, y sobre todo por ser el gran padre que es. A mi hermana, la cual quiero con todas las fuerzas de mi corazón, por ser una personita tan buena y tan especial conmigo, que sirva esta tesis de diploma como ejemplo de sacrificio profesional, y que ella lo puede hacer mucho mejor que yo.

A mis amigos: Anisel, Anibal, Isabel, Anita, Antero por ser tan incondicionales en todo momentos y haberme aceptado como soy, por todos sus consejos y sus grandes visiones de la vida, la cual han sido de una gran ayuda para mi vida profesional.

A Tania: por ser como mi segunda madre, por sus consejos y por seguir mis pasos tan de cerca.

A mi novia: por saber comprenderme tanto en estos meses, lo cual me ayudó mucho para ser mejor persona y mejor profesional, y por todo su apoyo incondicional.

Armando Esteban Pacheco Iglesias.

A mi madre, que con su ejemplo ha sabido enseñarme a luchar por mis sueños. A mi padre, que me ha hecho entender que esos sueños están altos y lejanos solo cuando los miramos desde abajo. A ambos, por ser mi sostén, mi inspiración, mi guía, por su apoyo incondicional y comprensión, por ser los mejores padres del mundo.

A mis amigos: los del hoy lejano Sancti – Spíritus, que siempre han estado ahí cuando los he necesitado y que siempre hablan con orgullo de ese que ya no está entre ellos desde hace 5 años. A Carel, Yasmany, Magdiel, Jesús, los que juntos nos sumamos en este hermoso empeño de formar parte del proyecto futuro, por ser desde que estudiamos juntos mucho más que amigos y compañeros.

A Dalila, por saber comprenderme y haber aprendido a lidiar con mi trabajo, mi estudio y mi lejanía.

A Anisia por ser para mí como una madre, a llaría por haberme querido tanto, pues sé que hoy en algún lugar se siente feliz de ver a su Alejandrito convertido en Ingeniero.

Alejandro Casanova Mutis.

AGRADECIMIENTOS.

A Rodolfo y Kiosmy, ayer compañeros de trabajo y hoy nuestros tutores, por su apoyo y sacrificio al ayudarnos con este trabajo.

A nuestros compañeros de trabajo y de estudios, por soportarnos, ayudarnos y estar ahí en todo momento.

A todos aquellos que ya no se encuentran a nuestro lado y que de una forma u otra han marcado la diferencia, contribuyendo a nuestro desarrollo como profesionales y sobre todo como seres humanos.

RESUMEN.

El presente trabajo plantea una estrategia de cómo eliminar los problemas existentes en los Registros Mercantiles de la República Bolivariana de Venezuela, en los procesos que complementan el proceso de Inscripción de documentos en dichos registros, a partir del un diseño e implementación de una solución de software que satisfaga las necesidades existentes.

Estos procesos son de vital importancia pues garantizan la publicidad registral al dar libre acceso a los usuarios a los asientos registrales, permitiendo tanto su consulta como la obtención de copias simples o certificadas de los mismos, además de constituir una garantía de la seguridad ciudadana para la realización de cualquier tipo de trámite. Parte de dichos procesos también complementan el proceso de Inscripción de documentos, porque en determinados momentos se hace imprescindible consultar la información contenida de los expedientes en el Archivo del Registro Mercantil.

El presente es un sistema de gestión de gran alcance, siendo el ámbito legal muy dinámico por lo que se hace necesario realizar un diseño e implementación que brinden flexibilidad y que sean fáciles de mantener de acuerdo a los cambios continuos que puedan surgir en determinado momento por parte de los clientes, ya sean cambios en la ley vigente, entre otros. Para ello se hizo un estudio de varias herramientas para desarrollar todos los artefactos necesarios, así como de patrones de diseño que permiten resolver problemas que suelen encontrarse en repetidas ocasiones, con el objetivo de obtener como resultado un producto de software que satisfaga las necesidades actuales y erradique todas las dificultades que presentan los Registros Mercantiles hoy en día.

ÍNDICE.

INTRODUCCIÓN.	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.	5
1.1 LEYES Y PRINCIPIOS QUE RIGEN LOS PROCESOS REGISTRALES.	5
1.2 REGISTROS MERCANTILES.	8
1.3 PROCESO DE INSCRIPCIÓN.	11
1.3.1 ¿QUÉ ES EL PROCESO DE INSCRIPCIÓN?	11
1.3.2 PASOS EN LA INSCRIPCIÓN DE DOCUMENTOS.	11
1.4 PROCESOS QUE COMPLEMENTAN EL PROCESO DE INSCRIPCIÓN.	13
1.4.1 PROCESO DE SOLICITUDES DE EXPEDIENTES.	13
1.4.2 PROCESO DE SOLICITUDES DE SELLADO DE LIBROS DEL COMERCIANTE.	14
1.4.3 PROCESO DE SOLICITUDES DE COPIAS.	14
1.5 SOLUCIONES INFORMÁTICAS EXISTENTES.	15
1.6 PLATAFORMAS PARA EL DESARROLLO DE SISTEMAS DE COMPLEJIDAD SIMILAR.	16
1.6.1 JAVA Y SU MÁQUINA VIRTUAL.	17
1.6.2 PLATAFORMA MONO.	18
1.6.3 .NET FRAMEWORK.	20
1.7 METODOLOGÍAS Y HERRAMIENTAS DE MODELADO PARA EL DISEÑO DE SOFTWARE.	23
1.7.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.	24
1.7.2 HERRAMIENTAS PARA EL DISEÑO DE SOFTWARE.	32
1.8 CONCLUSIONES.	35
CAPÍTULO 2. PROPUESTA DEL SISTEMA.	38
2.1 ARQUITECTURA DEL SOFTWARE.	38
2.1.1 ESTILO ARQUITECTÓNICO.	38
2.1.2 FRAMEWORK PARA LA GESTIÓN DE LA CAPA DE PRESENTACIÓN.	46
2.2 ESTÁNDAR DE CODIFICACIÓN.	49
2.3 PATRONES DE DISEÑO.	50
2.4 CARACTERÍSTICAS FUNDAMENTALES DE LOS PROCESOS QUE COMPLEMENTAN AL PROCESO DE INSCRIPCIÓN.	51
2.4.1 DESCRIPCIONES DE LOS PROCESOS.	51
2.5 CONCLUSIONES.	56

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN.	58
3.1 MODELO DE DISEÑO.	58
3.1.1 CAPAS Y SUBSISTEMAS EN EL MODELO DE DISEÑO.	59
3.1.2 DIAGRAMAS DE CLASES.	61
3.1.3 DIAGRAMAS DE INTERACCIÓN.	66
3.2 MODELO DE IMPLEMENTACIÓN.	69
3.2.1 DIAGRAMAS DE COMPONENTES.	70
3.3 DIAGRAMA DE DESPLIEGUE.	71
3.4 MÉTRICAS ORIENTADAS A OBJETOS APLICADAS AL DISEÑO DEL SOFTWARE.	72
3.4.1 ÁRBOL DE PROFUNDIDAD DE HERENCIA (APH).	73
3.4.2 NÚMERO DE DESCENDIENTES (NDD).	73
3.4.3 TAMAÑO DE CLASES (TC).	74
3.4.4 NÚMERO DE OPERACIONES REDEFINIDAS PARA UNA SUB-CLASE (NOR).	76
3.5 RESULTADOS OBTENIDOS DE LAS PRUEBAS DE CAJA BLANCA.	77
3.5.1 NUNIT FRAMEWORK.	78
3.5.2 RESULTADOS OBTENIDOS DE LAS PRUEBAS APLICADAS A LOS CONTROLES.	78
3.5.3 RESULTADOS OBTENIDOS DE LAS PRUEBAS APLICADAS A LAS CONTROLADORAS DE INTERFAZ.	79
3.5.4 RESULTADOS OBTENIDOS DE LAS PRUEBAS APLICADAS A LA FACHADA EN LA CAPA DE NEGOCIO.	80
3.6 CONCLUSIONES.	82
CONCLUSIONES GENERALES.	84
RECOMENDACIONES.	85
BIBLIOGRAFÍA.	86
GLOSARIO DE TÉRMINOS.	90
ANEXOS.	93
ANEXO 1. DIAGRAMAS DE CLASES DE GESTIÓN DE EXPEDIENTES.	93
ANEXO 2. DIAGRAMAS DE CLASES DE GESTIÓN DE SELLADO DE LIBROS.	95
ANEXO 3. DIAGRAMAS DE CLASES DE GESTIÓN DE COPIA DE DOCUMENTOS.	97
ANEXO 4. DIAGRAMAS DE SECUENCIAS DE GESTIÓN DE EXPEDIENTES.	99
ANEXO 5. DIAGRAMAS DE SECUENCIAS DE GESTIÓN DE SELLADO DE LIBROS.	101

ANEXO 6. DIAGRAMAS DE SECUENCIAS DE GESTIÓN DE COPIA DE DOCUMENTOS.	103
ANEXO 7. CLASES SUS ATRIBUTOS Y OPERACIONES.	105

INTRODUCCIÓN.

La realización de este proyecto, como parte de los acuerdos de cooperación entre Cuba y la hermana República Bolivariana de Venezuela, viene dada por inconsistencias en algunos sectores de los servicios públicos de este país. La Dirección General de Registros y Notarías, perteneciente al Ministerio del Poder Popular para las Relaciones Interiores y Justicia de la República Bolivariana de Venezuela, es el órgano encargado de ordenar, supervisar y controlar todos los procesos que se llevan a cabo en los registros, tanto Mercantiles, Públicos, Principales, y de Notarías.

Específicamente en los Registros Mercantiles de la República Bolivariana de Venezuela se brindan diferentes servicios que no interfieren con el proceso registral pero que si contribuyen con este y constituyen facilidades a las que tienen acceso todos los clientes. Entre estas funcionalidades podemos encontrar:

1. Solicitudes de Expedientes.

1.1. Los usuarios tienen la posibilidad de solicitar en calidad de préstamo documentos contenidos en piezas de los distintos expedientes para su uso dentro del propio registro.

1.2. Los funcionarios hacen solicitudes al departamento de archivo en el caso en que consideren necesario hacer una consulta a un determinado expediente como complemento del proceso registral, por ejemplo en el caso de Revisión Legal.

2. Solicitudes de Copia.

2.1. Existen dos tipos de solicitudes de copias, certificada y simple, ambos pueden ser mecanografiadas o fotostáticas.

2.1.1. *Solicitudes de Copia Certificada*: se caracterizan por ser una copia fiel del documento original y tienen tanta validez legal como dicho documento.

2.1.2. *Solicitudes de Copia Simple*: Estos tipos de copias también son una copia fiel del original, pero el Registro no certifica su validez legal, por cuanto carecen absolutamente de ella.

3. Solicitudes de Sellado de Libros.

3.1. La persona acreditada por la compañía para hacer este proceso es la que se presenta al registro, el cual le brinda el servicio de sellar los distintos tipos de libros. Esto no es más que el proceso de legalizar o dejar una constancia legal de algún tipo de libro que exista en esa compañía para futuros negocios.

En dependencia de la cantidad de recursos con que cuentan las diferentes oficinas registrales, algunas tienen sistemas para automatizar en mayor o menor medida los procesos que en ellas tienen lugar. El principal problema de la Dirección General de Registros y Notarías está dado por la no existencia de una plataforma tecnológica única, que brinde una solución informática homogénea para todas las Oficinas Registrales y las Notarías, permitiendo el control de los procesos que se generan a nivel de Oficina, por ejemplo en el control del Archivo, las Solicitudes de Copia y el Sellado de Libros, que son de gran importancia y con las condiciones actuales no se pueden llevar a cabo en todas las oficinas de una forma eficiente.

La automatización de todos estos procesos en todos y cada uno de los Registros Mercantiles lleva a plantearse el siguiente **problema científico**: ¿Cómo lograr una solución informática que permita gestionar las actividades que complementan el proceso registral, garantizando en cada momento la seguridad jurídica y la confiabilidad para el ciudadano venezolano?

En esta investigación se tienen como **objeto de estudio** el proceso de desarrollo de software.

Para dar solución al problema planteado, se traza como **objetivo principal**, desarrollar una solución de software en el proyecto Registros y Notarías, para los procesos de solicitud de copias de documentos, sellado de libros y gestión de solicitudes y préstamos de los expedientes para los Registros Mercantiles de Venezuela.

De este se derivan los siguientes **objetivos específicos**:

- Analizar la arquitectura definida para la implementación de la solución.
- Definir una solución de diseño que permita que el sistema sea auditable y controlable, tanto a nivel central como de oficina y que este constituya a su vez una garantía de la seguridad jurídica y del ciudadano venezolano en todo el proceso.

- Diseñar la solución para los procesos de solicitud de copia de documento, solicitud de sellado de libros y solicitud de expedientes en los Registros Mercantiles.
- Implementar la solución diseñada.

Después de ser visto el problema plantado, el objeto de estudio y los objetivos pues se determina como **campo de acción** en esta investigación el diseño e implementación como fases específicas dentro del proceso de desarrollo de software posibilitando la realización de un sistema de gestión mercantil capaz de estandarizar los procesos no registrales en los Registros Mercantiles de la nación venezolana.

Como **hipótesis** de esta investigación se tendría: si se desarrolla un sistema informático que gestione las actividades de solicitud de sellado de libro, solicitud de copia y solicitudes y préstamos de expedientes en los Registros Mercantiles de Venezuela, entonces se garantizaría una buena gestión y control de las actividades que se llevan a cabo en el departamento de Archivo de cada uno de los Registros Mercantiles en la República Bolivariana de Venezuela, la seguridad jurídica y del ciudadano venezolano y la publicidad registral.

Este trabajo tiene varias **tareas de la investigación** para darle seguimiento a los objetivos específicos antes mencionados, como son:

- Estudiar los modelos de los procesos de solicitudes de copia, sellado de libros y expedientes en el Archivo en varios de estos registros de la República con el fin de obtener una solución que cubra todas las necesidades.
- Realizar un estudio sobre las técnicas de programación a aplicar, las metodologías de desarrollo y plataformas de desarrollo como Java, Mono y .NET.
- Hacer un estudio de la arquitectura definida para la implementación de la solución con el fin de conocer si es factible para esta.
- Hacer un estudio de la solución de diseño para velar que se garantice la seguridad jurídica y del ciudadano venezolano en todos los procesos que se hagan en dichos registros.

Con este trabajo quedarán automatizadas todas las funcionalidades de los Registros Mercantiles de Venezuela que se realizan a diario. Entre los principales **resultados esperados** con esta investigación se encuentran:

- Solución configurable para las solicitudes de sellado de libros y copias de documentos.
- Solución para gestionar el préstamo tanto interno como externo de los expedientes que se encuentran en el archivo del Registro Mercantil.
- Incremento sustancial de la seguridad jurídica, lo que conlleva al incremento de la confianza y seguridad del ciudadano venezolano a la hora de realizar cualquier tipo de trámite en los Registros Mercantiles.

Este trabajo está compuesto por tres (3) capítulos fundamentales en los cuales se profundizará posteriormente, estos van desde la investigación hasta la implementación de los temas tratados y propuestos por el equipo.

El Capítulo 1 aborda la fundamentación teórica para la realización de la solución propuesta, para la cual se tienen en cuenta el estado del arte de las actuales técnicas de programación y metodologías existentes para el desarrollo de software, al igual que un estudio de las plataformas que se pueden utilizar para el desarrollo del sistema.

El Capítulo 2 aborda la propuesta del sistema, para la cual se tienen en cuenta un estudio de los distintos tipos de estilos arquitectónicos, fundamentando el que se escogió por la dirección del proyecto para la solución del sistema, como también se abordan los patrones de diseño más utilizados en este. Aquí se hace un análisis de los procesos más importantes dentro del sistema.

El Capítulo 3 aborda el diseño e implementación, de la propuesta que se brinda en el capítulo anterior, esta sección está compuesta por varias definiciones prácticas de los modelos de diseño y la realización de algunos de los diagramas de clases y secuencia como el diagrama de componentes más importantes dentro de los procesos que se proponen en el capítulo anterior. También se verá toda la parte de pruebas que se le hicieron al software, ya sea la aplicación de algunas métricas al diseño como algunas pruebas de unidad al código, llegando este, para el proceso de liberación del producto por parte de los clientes.

CAPÍTULO 1. Fundamentación Teórica.

Este capítulo comenzará refiriéndose a las principales definiciones y principios registrales que se deben tener en cuenta para desarrollar un sistema que automatice los importantísimos procesos como son: la solicitud de copias de documentos, el sellado de libros y la gestión de solicitudes de los expedientes por parte de los usuarios del registro. También explicar las funciones de los Registros Mercantiles y los procesos que nos competen, además de hacer un estudio de las plataformas y las metodologías para el desarrollo de software existentes, explicando sus principales características.

1.1 Leyes y Principios que rigen los procesos registrales.

La solución informática que se dará, automatiza procesos que tienen lugar en las oficinas Registrales Mercantiles. Ésta, además de ser un sistema de gestión y control, estará enmarcada en un contexto jurídico, por lo tanto se ajustará a la Ley vigente y tendrá que cumplir a cabalidad todos y cada uno de los principios que esta establece, siendo la seguridad jurídica y la publicidad registral, objetivos fundamentales en el desarrollo de la misma.

Dentro de la Ley de Registros y Notarías, redactada en la Asamblea Nacional de la República Bolivariana de Venezuela el 15 de Enero de 2007, la cual rige el funcionamiento de los registros mercantiles venezolanos, se identifican los principios por los cuales trabajan las oficinas registrales. A continuación se hará una enumeración de los principios y la interpretación de cada uno de ellos de acuerdo con la ley, con el objetivo de analizar su implicación en el proceso registral y la comprensión de los términos que en materia jurídica se manejan.

Principio de rogación: La presentación de un documento dará por iniciado el procedimiento registral, el cual deberá ser impulsado de oficio hasta su conclusión, siempre que haya sido debidamente admitido. (Artículo Nro 4, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Con este principio se hace referencia a lo que es el centro de atención en un registro, el documento. El proceso registral comienza con la presentación de éste, una vez que cumple con todos los requisitos iniciales y es debidamente admitido para procesarse. Con la aceptación del documento se está asumiendo su validez legal para impulsar el proceso hasta su conclusión. Destacar que siendo el mismo

el ente más importante, los datos que se recojan por el sistema estarán asociados a éste y su objetivo es facilitar la búsqueda y enriquecerlo con metadatos para facilitar el descubrimiento, gestión y conservación de estos recursos.

Principio de prioridad: Todo documento que ingrese al Registro deberá inscribirse u otorgarse con prelación a cualquier otro título presentado posteriormente, salvo las excepciones establecidas por la Ley. (Artículo Nro 5, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Este principio es muy importante cuando se habla de temas jurídicos y registrales, ya que el plantea que el primer documento que ingrese será el primero en tiempo y será por tanto el primero en derechos. En caso de algún litigio jurídico el documento que mayor peso o prioridad tiene es el presentado con mayor antelación. En el caso específico de los Registros Mercantiles tiene una gran importancia en la solicitud de denominaciones comerciales, por ejemplo si dos personas solicitan el mismo nombre por este principio tendrá mayor prioridad aquel que lo haya solicitado antes, al igual que en el caso de que por error a dos instituciones se les haya otorgado la misma denominación, en caso de litigio tendrá derecho legal sobre dicha denominación la primera en haberla obtenido.

Principio de especialidad: Los bienes y derechos inscritos en el Registro, deberán estar definidos y precisados respecto a su titularidad, naturaleza, contenido y limitaciones. (Artículo Nro 6, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

El principio de especialidad establece que los bienes y derechos a ser inscritos estén bien definidos y precisados por cada registro, que pueden ser Registros Públicos, Registros Civiles, Notarias y Registros Mercantiles, de manera tal que ninguno pueda asumir las funciones de otro; esto no quiere decir que un documento solo puede ser inscrito en un registro porque este documento puede variar la naturaleza de acuerdo con el contexto, por ejemplo una Sentencia de Divorcio puede hacerse en un Registro Civil pero también le compete al Registro Mercantil porque su naturaleza es demostrar el vínculo existente entre personas de una compañía. Esto tampoco significa que dos oficinas no puedan compartir el mismo espacio físico, pero en materia jurídica y registral su espacio esté bien delimitado.

Principio de consecutividad: De los asientos existentes en el Registro, relativos a un mismo bien, deberá resultar una perfecta secuencia y encadenamiento de las titularidades del dominio y de los demás

derechos registrados, así como la correlación entre las inscripciones y sus modificaciones, cancelaciones y extinciones. (Artículo Nro 7, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Para entender mejor lo que significa este principio, se deberá explicar brevemente que es un *asiento registral*.

Asiento registral: Es una inscripción o anotación en un registro público, las titularidades del dominio se refieren a los títulos o documentos que pertenezcan al dominio del Registro que se esté analizando, en el caso que nos compete, Registro Mercantil las compañías o personas, en los Registros Públicos los inmuebles. Cuando se dice Registro Público no se refiere a una de las clasificaciones de los Registros, se verá más adelante en uno de los principios que todos los Registros de la República Bolivariana de Venezuela son públicos, por lo tanto se refiere a las inscripciones que se hacen en cualquier tipo de registro. Entiéndase por asientos registrales el conjunto de todas estas inscripciones. (Real Academia de la Lengua Española).

Este principio establece que los asientos registrales deberán resultar en una perfecta secuencia, las inscripciones por lo tanto quedarán enumeradas de forma secuencial, si un conjunto de estas inscripciones se desea agrupar en un libro entonces estos también deberán quedar anotados de forma secuencial y así sucesivamente según tantos niveles de organización de la información se desee. Se establece además con la consecutividad la correlación entre las inscripciones y sus modificaciones, cancelaciones y extinciones, es decir para poder hacer una modificación a una empresa esta debe estar inscrita previamente al igual que para extinguirla, además dentro del contexto de las modificaciones estas también deberán tener su correlación.

Principio de legalidad: Solo se inscribirán en el Registro los títulos que reúnan los requisitos de fondo y forma establecidos por la Ley. (Artículo Nro 8, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Los documentos que se inscriban en el registro deberán ser revisados por la persona o un conjunto de personas calificadas para la revisión de estos requisitos de fondo y forma, verificando de esta forma el cumplimiento de todas las normas y leyes establecidas. El fondo se refiere a lo que está escrito en el

documento y es necesario leer para verificar que esté correcto, la forma se refiere al formato, estructura y organización de la información en el documento inscrito.

Principio de publicidad: La fe pública registral protege la verosimilitud y certeza jurídica que muestran sus asientos. La información contenida en los asientos de los Registros es pública y puede ser consultada por cualquier persona. (Artículo Nro 9, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Fe pública: Es la autoridad legítima atribuida a notarios, escribanos, agentes de cambio y bolsa, cónsules y secretarios de juzgados, tribunales y otros institutos oficiales, para que los documentos que autorizan en debida forma sean considerados como auténticos y lo contenido en ellos sea tenido por verdadero mientras no se haga prueba en contrario. (Real Academia de la Lengua Española).

Este principio implica que los documentos inscritos en los Registros deben poder ser descubiertos y consultados por cualquier persona, por lo tanto es responsabilidad del sistema automatizado brindar la flexibilidad, acceso y metadatos de estos documentos para que puedan ser examinados. He aquí lo que hace a los registros y la información que estos poseen un patrimonio público.

Al haberse explicado los principios que rigen el funcionamiento de los registros, se puede analizar el objeto de los Registros Mercantiles en particular. Todos estos principios en mayor o menos medida estarán reflejados en los procesos u operaciones que se deberán inscribir o anotar en los Registros Mercantiles.

1.2 Registros Mercantiles.

Los Registros Mercantiles tienen por objeto:

- La inscripción de los comerciantes individuales y sociales y demás sujetos señalados por la ley, así como la inscripción de los actos y contratos relativos a los mismos de conformidad con la ley.
- La inscripción de los representantes o agentes comerciales de establecimientos públicos extranjeros o sociedades mercantiles constituidas fuera del país, cuando hagan negocios en la República.

- La legalización de los libros de los comerciantes.
- El depósito y publicidad de los estados contables y de los informes periódicos de las firmas mercantiles.
- La centralización y publicación de la información registral.
- La inscripción de cualquier otro acto señalado en la ley.

Para entender cada uno de estos puntos antes mencionados se debe comprender que un Registro Mercantil se refiere a todos los llamados comerciantes individuales, compañías, y sociedades anónimas, además de todas las personas jurídicas que señale la Ley, siendo este término el que se emplea para referirse a todas las entidades. (Artículo Nro 51, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007)

Persona jurídica: “Organización de personas o de personas y de bienes a la que el derecho reconoce capacidad unitaria para ser sujeto de derechos y obligaciones, como las corporaciones, asociaciones, sociedades y fundaciones”. (Real Academia de la Lengua Española).

De esta definición se observa la diferencia entre persona natural y persona jurídica, la persona natural es exclusivamente un individuo y puede llevar personalmente los documentos al registro, mientras que la persona jurídica es una o más personas naturales, con o sin bienes que actúan en el plano jurídico con las identificaciones y nombres que se exigen.

El Registro Mercantil es el encargado de registrar estas personas jurídicas, así como el conjunto de operaciones que estas realicen. Estas actuaciones quedan anotadas en los registros de acuerdo a un proceso legal, proceso de inscripción, de manera tal que puedan ser consultados posteriormente; quedan anotados además los libros de los comerciantes y estados contables. En fin, todo lo que la Ley regule que deban contemplar las personas jurídicas, nacionales o extranjeras que operen en el territorio, debe quedar asentado en el Registro Mercantil correspondiente. Su principal finalidad son las personas jurídicas pero existen otros actos estipulados por la Ley que también se inscriben en dichos registros.

El Registro Mercantil según la Ley brinda:

Publicidad formal: El Registro Mercantil es público y cualquier persona puede obtener copia simple o certificada de los asientos y documentos, así como tener acceso material e informático a los datos. (Artículo Nro 62, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007)

Este artículo implica que los documentos que estén inscritos en dichos registros pueden ser accedidos por cualquier persona y son una copia formal y legal en todo momento hasta que se demuestre lo contrario por Ley. El conocimiento de los asientos registrales debe estar al alcance de cualquier interesado, de manera efectiva, y sin necesidad de tener que acudir a la intervención obligada de empresas o profesionales para su obtención, sin perjuicio del derecho de cada cual de hacerlo voluntariamente, si lo considera conveniente.

La publicidad formal de los asientos registrales es el único medio legalmente establecido para conocer la situación jurídica de los bienes inmuebles y demás circunstancias inscribibles en el caso de los Registros Públicos y de las instituciones o compañías en el caso de los Mercantiles. La finalidad de la publicación formal es la de probar, judicial y extrajudicialmente, la existencia, extensión y límites del derecho inscrito y que su titular es el único legitimado para disponer de él (efectos defensivo y ofensivo), así como la de agilizar el tráfico jurídico y dar certeza a la contratación, haciendo posible en el ámbito inmobiliario y mercantil el principio de seguridad jurídica.

La posibilidad de acceso directo a la publicidad formal en ningún caso significa que se acceda directamente a la base de datos de los Registradores, de forma que puedan los archivos ser alterados, manipulados, borrados o vaciados. Ello iría en contra de la finalidad del propio Registro y del entero sistema registral, basado en la seguridad de los pronunciamientos registrales. Por ello, se deben tener en cuenta medidas técnicas y organizativas necesarias para garantizar la integridad de los datos contenidos en los archivos y evitar su alteración, pérdida o destrucción.

Una vez analizados los principios que rigen el proceso registral y ver el objeto y leyes del Registro Mercantil se explicará a modo de general proceso de inscripción, aunque los Registros Mercantiles tienen como objeto no solo la inscripción de documentos, esta juega un papel fundamental por su complejidad y ser la base de todos los subsiguientes procesos.

1.3 Proceso de Inscripción.

1.3.1 ¿Qué es el proceso de inscripción?

El proceso de inscripción es una formalidad contenida en el Código de Comercio¹ cuya principal finalidad es la oponibilidad a terceros del contenido de los actos y contratos inscritos. Este efecto solo se obtiene con la publicidad legal, resultado de este proceso. La omisión de la inscripción trae como consecuencia la inoponibilidad de estos actos y contratos, por lo tanto las personas jurídicas se sienten en el interés de inscribir sus actos para hacerlos constar ante terceros en caso de negocios por interés de ambas partes. (Almenares, y otros, 2007).

La oponibilidad es la principal finalidad del Registro, por este medio se da a conocer a terceros y se facilita a estos el conocimiento de los actos y contratos inscritos. Esta oponibilidad es una medida de seguridad en las relaciones jurídicas que se origina en la vida de los negocios, especialmente en materia de contratos, además esta función no se concibe dentro del sistema legal sin las normas específicas para cada tipo de inscripción.

1.3.2 Pasos en la inscripción de documentos.

En un Registro Mercantil se inscriben varios tipos de documentos, de acuerdo con el estudio realizado a las leyes de este país, la naturaleza de los documentos que se inscriben y los resultados que provocan producto de su inscripción, se pueden clasificar en tres (3) grandes grupos:

- **Constitutivos:** Su inscripción trae como resultado, la constitución de una persona jurídica apta para operar en el territorio venezolano.
- **No Constitutivos:** Su objetivo es actualizar las actividades tanto internas ²como externas³.

¹ Congreso de la República Bolivariana de Venezuela, 1955.

² Se refiere al comportamiento de los socios entre si y con la sociedad como personas jurídicas distintas entre ellos mismos individualmente considerados. (Almenares, y otros, 2007).

³ Se refiere al comportamiento entre la sociedad y los terceros con quienes ella contrata. (Almenares, y otros, 2007).

- **Otros Actos:** Son aquellos que están directamente vinculados a una persona jurídica y que por lo tanto no se archivan, junto al conjunto de los documentos relacionados con una persona jurídica.

Estos procesos constan de cinco (5) pasos fundamentales los cuales son de estricto cumplimiento, no por todos los procesos, o sea algunos de estos procesos no tienen que cumplir todos estos pasos, debido a su naturaleza jurídica, estos pasos son:

- *Denominación Mercantil:* Su objetivo principal es reservar para el usuario el nombre de una compañía que posteriormente se utilizará por la legalización de todos los documentos referentes a esta.
- *Presentación de Documentos:* Su objetivo principal es la captura de los primeros metadatos asociados a los documentos y las primeras verificaciones de forma al mismo.
- *Revisión Legal:* Este es el paso donde tiene mayor peso la verificación de los principales requisitos de forma y fondo, siendo este paso el que más aporta al principio de legalidad. Su objetivo principal es, por así decirlo, dar el visto bueno del documento desde el punto de vista legal.
- *Otorgamiento:* Es el paso donde culmina para el usuario el proceso de inscripción de su documento, aquí se entrega el documento original firmado por todos los responsables del Registro Mercantil y se imprimen todas las salidas que se acompañan con este, el documento se asigna a los libros correspondientes y se hace una última verificación de las prohibiciones.
- *Archivo:* Su objetivo principal es la consecuente anotación en los libros del documento que fue asignado a estos en el paso de *otorgamiento*. Existe otra sección en este departamento que son los expedientes, donde ellos existen por cada persona jurídica que inscriba sus actos o contratos con el registro. Estos expedientes están divididos en piezas las cuales son un conjunto de documentos, tanto los expedientes como los documentos que se encuentran dentro de las piezas están enumerados, poniéndose de manifiesto el principio de consecutividad.

Una vez analizados el proceso de inscripción se analizarán aquellos procesos que complementan el proceso antes descrito porque es necesario tener en cuenta otras consideraciones a la hora de dar una solución automatizada en un Registro Mercantil, además, su contribución es de gran importancia a la garantía jurídica y la publicidad formal.

1.4 Procesos que complementan el proceso de inscripción.

Además de la inscripción de documentos, los Registros Mercantiles brindan otros servicios que pueden ser tanto internos como externos, de estos servicios que estaremos explicando a continuación se encuentran en el paso de *Archivo* y son: solicitud de expediente, solicitud de sellado de libros y solicitud de copias, los cuales son procesos que complementan el proceso de inscripción.

1.4.1 Proceso de solicitudes de expedientes.

Este proceso consta de dos (2) mecanismos de solicitud, o sea, puede solicitar el expediente un usuario, persona externa al registro, mediante un proceso denominado *Solicitud Formal*, o lo puede solicitar un funcionario del registro mediante otro proceso denominado *Solicitud Interna*.

1.4.1.1 Proceso de Solicitud Formal.

El proceso de solicitud formal no es más que un servicio que se les brinda a los usuarios, mediante el cual tienen la posibilidad de solicitar en calidad de préstamo documentos contenidos en piezas de los distintos expedientes para su uso dentro del propio Registro Mercantil. Este proceso se inicia cuando el usuario le proporciona el funcionario que lo atiende la información necesaria para encontrar el expediente, acto seguido se localiza el mismo y se le entrega al usuario para su consulta, junto a un recibo como constancia del préstamo y en el cual se hacen explícitas las características del mismo. Como característica principal y especificada por el Departamento de Archivo es prohibida la salida del documento del local habilitado por el registro para ese fin. Cuando el usuario termina de hacer uso del documento se lo devuelve al funcionario, el cual revisará que el mismo se encuentre en perfecto estado y lo restituirá a su ubicación original dentro del archivo.

1.4.1.2 Proceso de Solicitud Interna.

Este servicio se brinda a los funcionarios del registro, los cuales hacen solicitudes al departamento de archivo en el caso en que consideren necesario hacer una consulta a un determinado expediente como complemento del proceso de inscripción; por ejemplo en los casos de Revisión Legal, Solicitudes de Sellado y Solicitudes de Copia Certificada. En estos casos el funcionario que necesite el expediente hará la solicitud del mismo al departamento de archivo, el cual aceptará dicha solicitud en caso de que el expediente se encuentre disponible, en ese caso se lo envía al solicitante, el cual lo acepta cuando llega a sus manos. Una vez terminado el trabajo con el expediente, el funcionario se lo envía al departamento de archivo, el cual confirma la entrada del mismo y lo restituye a su lugar de origen, quedando disponible para nuevas solicitudes.

1.4.2 Proceso de solicitudes de Sellado de Libros del comerciante.

Este proceso es de gran importancia pues constituye la forma mediante la cual los libros del comerciante toman validez legal.

Las personas autorizadas por la compañía para hacer la solicitud de sellado de libros son las que se presentan en el Registro Mercantil, el cual brinda este servicio. El proceso consiste en que el usuario hace la solicitud, el funcionario que lo atiende verifica que dicha persona esté autorizada por la compañía para realizar el sellado, ya sea por ser miembro de su junta directiva o por estar autorizada por un poder. En la solicitud de sellado se especifica el o los tipos de libro que desean sellar, la cantidad de folios, la habilitación, la misma queda registrada y se procede a efectuar el sellado del o los libros para posteriormente hacerle entrega de los mismos ya listos al usuario.

1.4.3 Proceso de solicitudes de Copias.

Este proceso es de suma importancia, pues está concebido para garantizar de forma directa la publicidad registral, permitiendo que los usuarios del registro puedan acceder mediante la solicitud de copias a los distintos documentos que se encuentran archivados en el Registro Mercantil, según

establece el artículo Nro. 62 (ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA, 15 de Enero de 2007).

Las solicitudes de copias se clasifican según su validez legal en simples o certificadas, y de acuerdo a su tipo en fotostáticas o mecanografiadas. Las copias certificadas son fiel reproducción del documento original, teniendo la misma validez legal que el mismo, en tanto las copias simples también constituyen una reproducción fiel del original, pero a los efectos legales no tienen validez alguna. De acuerdo al tipo o forma en que se efectúa: las copias fotostáticas son fotocopias del documento original, mientras que las mecanografiadas no son más que la reproducción de forma manual del documento. En ambos casos el usuario después de culminado el trámite de solicitud, recibe su copia, la cual le pertenece por derecho siempre y cuando haya cancelado los montos correspondientes por la misma.

Una vez analizados los procesos que complementan el proceso de inscripción y las leyes por las cuales se rigen se hace necesario conocer o hacer un estudio de las soluciones informáticas existentes para realizar dichos procesos en la República Bolivariana de Venezuela.

1.5 Soluciones informáticas existentes.

Actualmente existen en los Registros Mercantiles de la República Bolivariana de Venezuela diversas aplicaciones informáticas que automatizan los procesos que nos ocupan, cambiando en su calidad y cantidad de servicios que brindan de acuerdo con los recursos de que disponga cada oficina. La gran mayoría de estas aplicaciones son módulos separados entre los cuales no existen vínculos, por lo que no cuentan con un sistema único dentro de la oficina que permita automatizar todo el proceso. Cada oficina cuenta con uno o varios módulos diferentes de software para automatizar cada uno de los servicios. El problema fundamental es que las soluciones existentes no cuentan con la funcionalidad necesaria para garantizar la seguridad jurídica y del ciudadano venezolano, pues los sistemas que existen son independientes y no se subordinan a un ente controlador.

A partir de las mismas incongruencias que existen en las soluciones existentes pues se hace necesario hacer un estudio de las plataformas existentes para el desarrollo de un sistema de características

similares, y así poder hacer una elección de esta que cumpla con toda la complejidad que este sistema requiere.

1.6 Plataformas para el desarrollo de sistemas de complejidad similar.

Para la implementación de un sistema informático que de solución a la problemática antes planteada es necesario escoger una plataforma de desarrollo que reúna varias características acordes a los intereses del sistema y de las personas e instituciones que se beneficiarán del mismo.

Una plataforma de desarrollo es el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo, sin embargo, también es posible encontrarlas ligadas a una familia de lenguajes de programación o a una Interfaz de Programación de Aplicaciones o API (Application Programming Interface). Se le denomina plataformas a los diferentes ambientes creados para el desarrollo de software, (Almenares, y otros, 2007).

Actualmente en el mundo existe una gran variedad de plataformas de desarrollo para aplicaciones electrónicas y de escritorio. En los últimos años las plataformas de desarrollo han ido evolucionando siempre con el objetivo de ofrecer una interfaz más amigable para el programador y abstraerlo de las funciones más elementales, la creación de aplicaciones con sus prestaciones para diferentes campos de acción como los son, la Web y las aplicaciones de escritorio. En los años noventa solo existían como la representación del desarrollo de software, leguajes de programación (con su ambiente visual en algunos casos) para desarrollar sistemas. Por citar algunos ejemplos se encuentran el Borland Delphi, Borland C++, el Visual Basic y Java. En la actualidad con la aparición de la plataforma .Net de Microsoft, los conceptos del desarrollo de aplicaciones han sufrido unos cambios radicales por el simple hecho de unificar en un único ambiente de desarrollo múltiples lenguajes de programación representados por un Framework⁴ común. De las diferentes plataformas existentes en el mundo para el desarrollo de aplicaciones de escritorio se encuentran:

⁴ Marco de trabajo, denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entornos de ejecución distribuidos.

1.6.1 Java y su Máquina Virtual.

El concepto de máquina virtual es antiguo. Fue usado por IBM en 1959 para describir uno de los primeros sistemas operativos que existieron en la historia de la computación. En 1970, el ambiente de programación de *SmallTalk* llevó la idea a un nuevo nivel y construyó una máquina virtual para soportar abstracciones orientadas a objetos de alto nivel, sobre las máquinas subyacentes. La Máquina Virtual de Java (MVJ) es el núcleo del lenguaje de programación Java. De hecho, es imposible ejecutar un programa Java sin ejecutar alguna implantación de la MVJ. En la MVJ se encuentra el motor que en realidad ejecuta el programa Java y es la clave de muchas de las características principales de Java, como la portabilidad, la eficiencia y la seguridad. Siempre que se corre un programa Java, las instrucciones que lo componen no son ejecutadas directamente por el hardware sobre el que subyace, sino que son pasadas a un elemento de software intermedio, que es el encargado de que las instrucciones sean ejecutadas por el hardware. Es decir, el código Java no se ejecuta directamente sobre un procesador físico, sino sobre un **procesador virtual Java**, precisamente el software intermedio del que habíamos hablado anteriormente. (Carballeira, 2000).

Sun Microsystems⁵ utiliza el término "Máquina Virtual Java", para referirse a la especificación abstracta de una máquina de software para ejecutar programas Java. La especificación de esta máquina virtual, define elementos como el formato de los archivos de clases de Java (.class), así como la semántica de cada una de las instrucciones que componen el conjunto de instrucciones de la máquina virtual. A las implantaciones de esta especificación se les conocen como "Sistemas en Tiempo de Ejecución Java".

Las clases en Java pueden contener métodos que no estén implementados por códigos de operación (bytecode) Java, sino por algún otro lenguaje compilado en código nativo y almacenado en bibliotecas de enlace dinámico, como las DLL de Windows o las bibliotecas compartidas SO de Solaris⁶. El sistema de tiempo de ejecución incluye el código necesario para cargar dinámicamente y

⁵ Empresa informática de Silicon Valley, fabricantes de semiconductores y software, construida en 1982 por el alemán Andreas von Bechtolsheim y los norteamericanos Vinod Koshla, Bill Joy, Scott McNealy y Marcel Newman.

⁶ Es un sistema operativo desarrollado por Sun Microsystems. Es un sistema certificado como una versión de UNIX.

ejecutar el código nativo que implementa estos métodos. Una vez que se enlaza el módulo que contiene el código que implementa dicho método, el procesador virtual atrapa las llamadas a éste y se encarga de invocarlo. Este proceso incluye la modificación de los argumentos de la llamada, para adecuarlos al formato que requiere el código nativo, así como transferirle el control de la ejecución. Cuando el código nativo termina, el módulo de soporte para métodos nativos se encarga de recuperar los resultados y de adecuarlos al formato de la Máquina Virtual Java. (Carballeira, 2000).

Las máquinas virtuales tienen algunas desventajas dentro de las que se encuentran, es que agregan gran complejidad al sistema en tiempo de ejecución. Por ejemplo, la MVJ espera que la computadora sobre la que subyace, soporte el estándar de IEEE para los números de punto flotante de 32 y 64 bits, así como enteros largos de 64 bits. La mayoría de las plataformas lo hacen pero hay algunas que no, lo que implica trabajo extra. (Carballeira, 2000).

La principal desventaja de los lenguajes basados en máquina virtual, es que efectivamente son más lentos que los lenguajes completamente compilados, debido a la sobrecarga que genera tener una capa de software intermedia entre la aplicación y el hardware de la computadora.

Según (Carballeira, 2000), la Máquina Virtual de Java tiene tres (3) deficiencias, un conjunto de instrucciones no ortogonal⁷, difícil de entender, ya que se utiliza un byte para codificar el código de operación de las instrucciones del procesador virtual Java y no posee un árbol de análisis sintáctico, o sea el código intermedio, usado en la MVJ, es simple y plano, es decir no incluye información acerca de la estructura del método original.

1.6.2 Plataforma Mono.

Es un proyecto de código abierto impulsado por Novell y creado por Miguel de Icaza como alternativa al Framework .NET para los programadores de software libre para brindar un grupo de herramientas, basadas en GNU/Linux y compatibles con .NET según lo especificado por el ECMA⁸.

⁷ Un lenguaje de programación es ortogonal, si tiene el mismo número de instrucciones asociadas a cada uno de los tipos de datos.

⁸ European Computer Manufacturers Association, ECMA International es una organización basada en memberships de estándares para la comunicación y la información. Creada en 1961.

Esta plataforma presenta importantes ventajas, es ideal para desarrollo de plataformas cruzadas y se ha posicionado como un entorno que permite ejecutar en Linux aplicaciones diseñadas para Microsoft .Net en entorno Windows, facilitando la migración de aplicaciones a Linux y aumentando su base de desarrolladores y usuarios (Cano Ossa, Agosto 12, 2006).

Según (Seoane, 2004), la plataforma Mono posee independencia de lenguajes, se puede usar clases escritas en cualquier lenguajes que este soporte (Python, C#, Mono Basic, Java, Nemerle, Boo); posee también independencia de plataforma, lo cual las aplicaciones son muy portables y la mayoría compatibles en binario entre plataformas; tienen un gran soporte para bases de datos MS SQL, MySQL, Postgres, OLE DB; posee una gestión automática de memoria, lo cual es una fuente inagotable de errores y se pierde una gran cantidad de tiempo programando esto, si se automatiza, se obtiene más tiempo que se puede dedicar a resolver el verdadero problema; como otra característica presente está las aplicaciones GUI⁹ multiplataforma, lo cual se pueden escribir aplicaciones con interfaz gráfica que se ejecutan invariablemente en multitud de plataformas.

Las aplicaciones se traducen a CIL¹⁰, la ventaja fundamental de CLI frente los formatos usados anteriormente es claramente la independencia del lenguaje, por ejemplo, ya no tendrás que escribir tu código en Fortran o en C, simplemente porque las librerías que debes de usar están escritas en ese lenguaje, puedes, por ejemplo, escribir tus librerías en C# y usarlas desde Python sin ninguna dificultad, de hecho, no hay diferencia ente una librería escrita en Python, C# o Fortran si éstos generan CLI. (Seoane, 2004).

A diferencia de los programas tradicionales que se ejecutan sobre el sistema directamente, los programas en la plataforma Mono se ejecutan sobre un entorno controlado de ejecución conocido como la máquina virtual. Este entorno proporciona numerosas ventajas sobre la ejecución tradicional directa: gestión de memoria automática (el sistema se encarga de recuperar automáticamente la memoria no usada por las aplicaciones simplificando la gestión a las aplicaciones), un entorno seguro de ejecución (donde podemos definir los recursos físicos y lógicos

⁹ Graphical User Interface, tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

¹⁰ Common Intermediate Language, se conoció anteriormente como lenguaje intermedio de Microsoft (MSIL).

a los que la aplicación tiene acceso) y un sistema de control de errores y ejecución que permite una gestión de errores avanzada. (Hernández, 2004).

La principal desventaja que presenta esta plataforma es que es muy inmadura, todavía no ha madurado lo suficientemente como para implementar grandes proyectos de gestión, y debido a esto pues presentan errores como son en el mecanismo de actualización, que maneja el redibujado de la aplicación, no funciona correctamente cuando ésta es ocultada por otra aplicación. Adicionalmente, el mecanismo de eventos presenta algunos problemas con algunos controles de interfaz.

1.6.3 .NET Framework.

Según (MSDN, 2008), el .NET Framework es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Lenguajes de compilación.
- Bibliotecas de clases de .NET.
- CLR¹¹.

.NET Framework soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), Visual Basic o C++ hasta otros lenguajes menos conocidos como Perl o Cobol.

El CLR es el verdadero núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32.

Cuando se está programando una aplicación muchas veces se necesitan realizar acciones como manipulación de archivos, acceso a datos, conocer el estado del sistema, implementar seguridad,

¹¹ Common Language Runtime, es un componente de maquina virtual del .NET Framework de Microsoft.

etc. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita. La principal ventaja de los espacios de nombres de .Net es que de esta forma se tiene toda la biblioteca de clases de .Net centralizada bajo el mismo espacio de nombres (System). Además, desde cualquier lenguaje se usa la misma sintaxis de invocación, ya que a todos los lenguajes se aplica la misma biblioteca de clases.

Uno de los mayores problemas de las aplicaciones actuales es que en muchos casos tienen que tratar con diferentes archivos binarios (DLL's), elementos de registro, conectividad abierta a bases de datos (ODBC), entre otros. Para solucionarlo el Framework de .Net maneja un nuevo concepto denominado ensamblado. Los ensamblados son ficheros con forma de EXE o DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Por tanto la solución al problema puede ser tan fácil como copiar todos los ensamblados en el directorio de la aplicación.

La idea fundamental de Microsoft .NET es un cambio de enfoque en lo que es la informática, pasando de un mundo de aplicaciones, sitios Web y dispositivos aislados a una infinidad de computadoras, dispositivos, transacciones y servicios que se conectan directamente y trabajan en conjunto para ofrecer soluciones más amplias y ricas en contenido. Esta plataforma cuenta con ciertas ventajas a tener en cuenta como son:

- **Código administrado:** El CLR (Command Language Runtime) realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- **Interoperabilidad multilenguaje:** El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL¹²).
- **Compilación Just-in-Time:** El compilador JIT¹³ incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.

¹² Es un código de bites que la tecnología .NET de Microsoft utiliza para lograr independencia de la plataforma y seguridad en ejecución.

- **Recolector de basura:** El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (Garbage Collector¹⁴). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por que liberar la memoria de forma explícita aunque también sea posible hacerlo manualmente.
- **Seguridad de acceso al código:** Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente del Web sin tener que preocuparse si esto va a estropear el sistema.
- **Despliegue:** Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.

Algunas desventajas que presenta esta plataforma es el mantenimiento en múltiples lenguajes. Mantener un proyecto en múltiples lenguajes es costoso. Si una aplicación está realizada en varios lenguajes se necesitan expertos en varios lenguajes para entenderla y mantenerla, aumentando los costos. Otros de las desventajas fundamentales es que no es multiplataforma, o sea ella solo está disponible para la familia de Windows. El tema de las licencias es también otras de las desventajas por lo que es un código cerrado y no hay licencias libres. La infraestructura para desarrollar en .NET representa un alto costo para las empresas. Y para nuestro país la desventaja principal es que esta plataforma está representada por la Microsoft, por lo que por razones políticas no se puede obtener para utilizarse en software.

¹³ Técnica para mejorar el rendimiento de sistemas de programación que compilan a bytecode, consiste en traducir el bytecode a código máquina nativo en tiempo de ejecución.

¹⁴ Mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación.

Después de hacer un estudio de las plataforma más utilizadas para el desarrollo de sistemas de gestión informáticos, se estudiarán y analizarán las metodologías y herramientas que sustentan al proceso de desarrollo de software.

1.7 Metodologías y herramientas de modelado para el diseño de software.

El modelado de sistemas informáticos de altos grados de complejidad requiere del uso de la Ingeniería de Software para llevar a cabo de una manera organizada y bien definida las tareas del software en cuestión.

Según (Medina Pasaje, Marzo 7, 2006), en el mundo existen diferentes metodologías para llevar a cabo del desarrollo de estos sistemas, y cada una de ellas contienen ciertos pasos que identifican el proceso de ingeniería de software, y estos son los siguientes:

- **Análisis de requisitos:** Extraer los requisitos de un producto de software es la primera etapa para crearlo. Los analistas de software se entrevistan con los clientes donde ellos expresan todos los procesos o lo que tiene que hacer el software y de esta consulta pues se extraen los primeros requisitos. Se requiere de gran habilidad y experiencia en la Ingeniería de Software para reconocer requisitos incompletos, ambiguos o contradictorios.
- **Especificación:** Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.
- **Diseño y arquitectura:** Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red.
- **Programación:** Reducir un diseño a código puede ser la parte más obvia del trabajo de Ingeniería de Software, pero no es necesariamente la porción más larga.

- **Prueba:** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral.
- **Documentación:** Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.
- **Mantenimiento:** Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de dos tercios (2/3) de toda la Ingeniería de Software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores. La mayor parte consiste en extender el sistema para hacer nuevas cosas.

1.7.1 Metodologías de desarrollo de software.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, solo se obtendrán clientes insatisfechos con los resultados y desarrolladores aún más insatisfechos.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. En éstas se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. (Medina Pasaje, Marzo 7, 2006).

Existen numerosas propuestas metodológicas que inciden directamente en el proceso de desarrollo de software. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos¹⁵ que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero

¹⁵ Herramientas organizativas identificadas para la confección organizativa y correcta del software en cuestión.

también han presentado problemas en otros muchos. En contraposición, ante las dificultades para utilizar metodologías tradicionales con restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la Ingeniería de Software, asumiendo el riesgo que ello conlleva. En este escenario, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico, por estar especialmente orientadas para proyectos pequeños, éstas constituyen una solución a medida para ese entorno, aportando una elevada simplificación, que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

Estos puntos de vista no implican desechar metodologías, por el contrario, éstas se complementan y es preciso saber en qué casos usar una u otra. (Molpeceres, Diciembre 15, 2002).

1.7.1.1 Metodologías ágiles.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término *ágil*, aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó “The Agile Alliance”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos.

Las metodologías ágiles surgen como una alternativa a las llamadas metodologías tradicionales. Se caracterizan principalmente por su carácter adaptativo y su orientación hacia la gente. La Programación Extrema (XP) es la metodología ágil más estudiada hasta el momento. Ha generado gran interés por su creciente número de casos de éxito en la industria. (Canós José A.; Letelier, Patricio; Panadés, María Carmen; DSIC -Universidad Politécnica de Valencia).

- **Programación Extrema (XP)** (Robles, Gregorio; Grupo de Sistemas y Comunicaciones, Abril 4, 2003).

Constituye uno de los ejemplos más exitosos de metodología ágil, es la más popular en la actualidad. Se centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Propone añadir funcionalidad con retroalimentación continua, tener el manejo de cambios como parte sustantiva del proceso, que el costo del cambio no dependa de la fase en que se encuentra el proyecto, no introducir funcionalidad antes de ser necesaria, el cliente se convierte en parte del equipo de desarrollo.

XP se caracteriza por:

- Pruebas unitarias.
 - Re fabricación.
 - Programación en pares.
- **Scrum** (Baufest, proyecto Exactas, 2006).

Es una de las metodologías ágiles más conocidas para la gestión de proyectos. Surge a mediados de los '80 y se desarrolla originalmente en el sector TIC¹⁶, pero es aplicable en cualquier proyecto en el que exista una lista de funcionalidades o bloques de trabajo por realizar, un entorno complejo con requisitos cambiantes y un equipo de desarrollo asignado a dicha tarea. Scrum fue expuesta por Hirotaka Takeuchi e Ikijuro Nonaka, en el artículo *The New New Product Development Game* (Harvard Business

¹⁶ Tecnologías de la Información y las Comunicaciones.

Review, Enero - Febrero, 1986). El nombre proviene de la posición de salida del rugby homónima, y se justifica diciendo que en esta metodología cada uno desde su puesto contribuye al equipo, haciendo toda fuerza para el mismo lado. El "gol" es que el cliente quede satisfecho con el producto desarrollado. El "contrincante" es la ambigüedad o la definición pobremente que tiene el software y que hace tortuoso su desarrollo.

Nonaka y Takeuchi extraen las bases de Scrum de las prácticas que observan en las empresas con buenos resultados de rapidez y flexibilidad en la producción: Xerox, Canon, Honda, NEC, Epson, Brother, 3M o Hewlett-Packard; y son:

- Inestabilidad consustancial al entorno de desarrollo.
- Equipos auto-organizados.
- Solapamiento de las fases del desarrollo.
- Multi-aprendizaje.
- Control sutil.
- Transferencia de aprendizaje a nivel organizacional.
- **Microsoft Solution Framework (MSF)** (Microsoft Corporation, June, 2003).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Microsoft Solutions Framework provee un marco adaptable para la entrega exitosa de soluciones informáticas de forma rápida, con menos personal involucrado y menor riesgo, con un resultado de mayor calidad. MSF es llamada "Framework", en lugar de metodología, por razones específicas, a diferencia de las metodologías prescriptivas, posee un entorno flexible y escalable que puede ser adaptado a las necesidades de cualquier proyecto, independientemente de su tamaño o complejidad, para planificarlo, construirlo e

implementarlo. Se compone de varios modelos encargados de planificar las distintas partes implicadas en el desarrollo del proyecto: modelo de arquitectura del proyecto, modelo de equipo, modelo de proceso, modelo de gestión del riesgo, modelo de diseño de proceso y modelo de aplicación.

MSF tiene las siguientes características:

- Adaptable.
- Escalable.
- Flexible.
- Independiente de la tecnología.
- Existencia de otras de las metodologías ágiles.
 - Crystal.
 - Evolutionary Project Management (Evo).
 - Feature Driven Development (FDD).
 - Adaptive Software Development (ASD).
 - Lean Development (LD) y Lean Software Development (LSD).
 - RUP Ágil, existiendo dos variantes AUP y EUP.

1.7.1.2 Metodologías tradicionales o no ágiles.

- **Rational Unified Process (RUP)** (Jacobson, y otros, 2000).

El Proceso Unificado Racional es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman,

uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software compró una compañía sueca llamada Objectory AB, fundada por Ivar Jacobson, famoso por haber incorporado los casos de uso a los métodos de desarrollo orientados a objetos. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory (el proceso de la empresa Objectory AB). El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten.

RUP es uno de los procesos más generales de los existentes actualmente, ya que en general está pensado para adaptarse a cualquier proyecto, no tan solo de software. Sus autores destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los casos de uso, está centrado en la arquitectura, y es iterativo e incremental.

- Proceso dirigido por casos de usos.

Los casos de uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los casos de uso representan los requisitos funcionales del sistema. En RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo. Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

- Proceso centrado en la arquitectura.

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, es necesaria para controlar el desarrollo e involucra los aspectos estáticos y dinámicos más significativos del sistema. La definición de la arquitectura

debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo. RUP presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los casos de uso y la forma la proporciona la arquitectura. Existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como casos de uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

- Proceso iterativo e incremental.

De esta forma el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de

trabajo relevantes y refinando la arquitectura. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Estas fases son: inicio, elaboración, construcción y transición.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline¹⁷ de la arquitectura. Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos. En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura. En la fase de construcción, se lleva a cabo la implementación del producto por medio de una serie de iteraciones. Para cada iteración se seleccionan algunos casos de uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

¹⁷ Instantánea del estado de todos los artefactos del proyecto, registrada para efectos de gestión de configuración y control de cambios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

1.7.2 Herramientas para el diseño de software.

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son las aplicaciones informáticas que tienen como tarea principal lograr una mayor productividad en el desarrollo de software y una reducción de costos de desarrollo. Estas herramientas contribuyen de manera directa en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Según (Almenares, y otros, 2007), cada una de estas herramientas persigue nueve objetivos principales:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

A continuación se relacionan algunas de estas herramientas existentes:

- Rational Rose Enterprise (Rational Software Corporation, 2002).

Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Como todos los demás productos **Rational Rose**, proporciona un lenguaje común de modelado, UML, para el equipo que facilita la creación de software de calidad más rápidamente. Posee características como:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
 - Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
 - Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
 - La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
 - Capacidad de análisis de calidad de código.
 - El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
 - **Modelado UML** para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
 - Publicación web y generación de informes para optimizar la comunicación dentro del equipo.
- Visual Paradigm for UML.

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad,

mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta un conjunto de características, las más fundamentales son:

- Modelado colaborativo con CVS¹⁸.
 - Interoperabilidad con modelos UML 2.0 a través de XML.
 - Generación de código - Modelo a código, diagrama a código.
 - Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
 - Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
 - Generador de informes para generación de documentación.
 - Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Enterprise Architect for UML v2.1.

Enterprise Architect (EA) combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. EA puede equipar a su equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. EA provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

¹⁸ Concurrent Versions System. Aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros.

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. También hay Add-ins gratis para CORBA y Python disponibles. Con un editor de código fuente con "resaltador de sintaxis" incorporado, EA le permite navegar y explorar su modelo de código fuente en el mismo ambiente.

- Umbrello UML Modeller.

Umbrello es una herramienta libre para crear y editar diagramas UML, que ayuda en el proceso del desarrollo de software. Fue desarrollada por Paul Hensgen, y está diseñado principalmente para KDE, aunque funciona en otros entornos de escritorio. Umbrello maneja gran parte de los diagramas estándar UML pudiendo crearlos, además de manualmente, importándolos a partir de código en C++, Java, Python, IDL, Pascal/Delphi, Ada, o también Perl (haciendo uso de una aplicación externa). Así mismo, permite crear un diagrama y generar el código automáticamente en los lenguajes antes citados, entre otros. El formato de fichero que utiliza está basado en XML. También permite la distribución de los modelos exportándolos en los formatos DocBook¹⁹ y XHTML²⁰, lo que facilita los proyectos colaborativos donde los desarrolladores no tienen acceso directo a Umbrello o donde los modelos van a ser publicados vía web. Umbrello se distribuye en el módulo kdesdk de KDE y actualmente va por la versión 1.3, liberada el 16 de agosto del 2004. (Hensgen, Paul; Umbrello UML Modeller Authors, October 15, 2003).

1.8 Conclusiones.

En este capítulo se hizo un análisis y un estudio de todas las leyes que se ven comprometidas en el sistema que se quiere desarrollar para la hermana república venezolana, también se ha hecho un estudio

¹⁹ Es un aplicación del estándar SGML/XML e incluye una DTD propia, se utiliza de principalmente en el área de la documentación técnica, especialmente para documentar todo tipo de material y programas informáticos. (Walsh, y otros, Wed, 11 Oct 2006)

²⁰ Lenguaje de Marcado de Hipertexto Extensible, es una versión de HTML, que nace con el objetivo de remplazar a HTML ante su limitación de uso con las herramientas basadas en XML. (Astarita, Emilio; Creative Commons)

profundo en los aspectos que se desean implementar, dígame las solicitudes y préstamos de expedientes, las solicitudes de sellado de libros de comerciantes y las solicitudes de copias tanto simples como certificadas, para resolver los problemas existentes en los diferentes registros mercantiles de la República Bolivariana de Venezuela. Se hizo un análisis de las plataformas existentes para el desarrollo de software y después de haberse visto todas las ventajas y desventajas de estas ofrecen pues se decidió escoger la plataforma .NET por las siguientes características:

- Solo se disponía de dos (2) meses para entregar un prototipo funcional y ser mostrado a la parte venezolana como contrato inicial del proyecto.
- Poca experiencia por parte del equipo de desarrollo en la construcción de proyectos de complejidad similar.
- El número de personas que tenían dominio de la plataforma .NET en la universidad era mucho mayor que los de la plataforma Java y Mono.
- Toda la gama de librerías y facilidades que brinda esta plataforma para el desarrollo de software en tiempos records.
- La plataforma Mono no estaba concluida y no se consideraba un Framework estable y maduro y los entornos de desarrollo no ofrecían las mismas ventajas y velocidad de trabajo que el Visual Studio .NET.

También en este capítulo se hizo un análisis profundo en las metodologías y las herramientas que le dan soportes a estas y se decidió escoger RUP junto a la herramienta CASE Rational Rose Enterprise, se pensó que es una buena elección por un grupo de características que estas presentan como son:

- Las características que posee el software, ya sea por su grandeza y por todos los contratos que exigen los clientes, o sea se desea entregar toda una serie de documentación bastante grande que presenta la metodología escogida.
- El cliente no se considera como otro rol más dentro del proyecto porque el software se haría a distancia casi todo el tiempo.
- Poco conocimiento de otras metodologías de desarrollo e incluso de otras herramientas CASE por el equipo de desarrollo.

- La herramienta CASE escogida ofrece una serie de sub-módulos a parte de la del modelado del negocio y sistema, que sirve para todo este proceso de formalidad entre los integrantes del proyecto y los clientes.

Una vez que se expuso toda la fundamentación teórica, se define el capítulo 2 como una propuesta del sistema que se desea desarrollar, en la cual se vera inmerso todas las definiciones técnicas que se aplicarán en el desarrollo del software.

CAPÍTULO 2. Propuesta del sistema.

En este capítulo se hace una valoración de las arquitecturas existentes a escoger para la realización del sistema, de cada una se hace un análisis de por qué no es apropiada para dicha realización y se exponen los criterios o argumentos necesarios de la seleccionada. Se dará una explicación de los patrones de diseño más usados en la realización del sistema y se explicarán los procesos más fundamentales que intervienen para la construcción del mismo.

2.1 Arquitectura del Software.

Según (Clements, 1996), la arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

Otras de las definiciones de arquitectura de software la brinda el documento de la IEEE Std 1471-2000 que plantea: “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”. Esta es la definición que se ha tomado como oficial por todas las instituciones que desarrollan software a nivel mundial, como ha sido por ejemplo Microsoft.

2.1.1 Estilo Arquitectónico.

No es objetivo de este epígrafe hacer un análisis de los distintos estilos arquitectónicos en su totalidad, sino solo aquellos que por su estructura y solución podrían considerarse para la construcción del sistema, analizando las ventajas y desventajas que estos poseen para determinar si debe ser o no aceptado para el desarrollo del software.

Según (Shaw, y otros, Abril de 1996), identifican los estilos arquitectónicos como un conjunto de reglas de diseño que identifica las clases de componentes y conectores que se pueden utilizar para componer en sistema o subsistema, junto con las restricciones locales o globales de la forma en que

se lleva a cabo la composición. Es en gran medida la interacción entre los componentes, mediados por conectores, lo que confiere a los distintos estilos sus características distintivas.

2.1.1.1 Estilos de Flujo de Datos.

La familia de los estilos de Flujos de Datos, enfatiza la reutilización y la modificabilidad, es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos, ejemplos de estas podemos encontrar las arquitecturas de proceso secuencial por lotes, red de flujos de datos, y tuberías y filtros.

Según (Reynoso, y otros, 2004), el sistema tubería-filtros se percibe como una serie de transformaciones sobre sucesivas piezas de los datos de entrada. Los datos entran al sistema y fluyen a través de los componentes. En el estilo secuencial por lotes (batch sequential) los componentes son programas independientes; el supuesto es que cada paso se ejecuta hasta completarse antes que se inicie el paso siguiente.

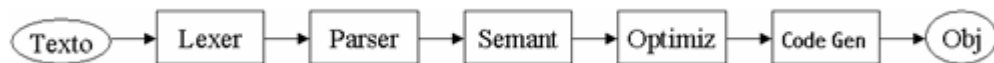


Figura 1. Compilador en tubería-filtro

De las arquitecturas que ofrece esta familia de estilos no se recomienda ninguna de ellas para el desarrollo de un sistema con las características que se necesita. A pesar de que es simple de entender, implementar, al igual que enfatiza la reutilización y la modificabilidad, en los procesos que complementan al proceso de inscripción el expediente pasa por determinados pasos, no se escoge pues no se hacen modificaciones al expediente sino que se le agregan metadatos a lo largo de todo el flujo y en otros casos lo único que se le hace es transmitir esos datos a diferentes departamentos pero sin modificaciones a este documento. (Almenares, y otros, 2007).

Además de las desventajas que traería este estilo para el sistema, se le integran las siguientes:

- El patrón puede resultar demasiado simplista, especialmente para la orquestación de los servicios que podrían ramificar la ejecución lógica de negocios de formas complicadas, (Reynoso, y otros, 2004).
- No se maneja con demasiada eficiencia construcciones condicionales, bucles y otras lógicas de control de flujo. Agregar un paso suplementario afecta el rendimiento de cada ejecución en la tubería. (Reynoso, y otros, 2004).
- Según (Reynoso, y otros, 2004), el estilo no es apto para manejar situaciones interactivas, sobre todo cuando se requieren actualizaciones incrementales de la representación en pantalla.

No se seleccionó ningún estilo de esta familia debido a su simplicidad, no son los más adecuados, pues los procesos que complementan al proceso de inscripción (solicitudes de expedientes, tanto internos como externos, solicitudes de sellado de libros de comerciantes y las solicitudes de copias, tanto simples como mecanografiadas) son complejos en su mayoría por lo cual esto traería desventajas en el sistema

2.1.1.2 Estilos Centrados en Datos.

Otros de los estilos arquitectónicos es el Centrado en Datos, esta familia de estilos enfatiza la integridad de los datos. Se estima apropiada para sistemas que se fundan en acceso y actualización de datos en estructuras de almacenamiento. Entre los subestilos característicos están los Repositorios, Base de Datos y Arquitectura de Pizarra.

En la Arquitectura de Pizarra o Repositorio tiene dos componentes principales, una estructura de datos que representa el estado actual y una colección de componentes independientes que operan sobre él.

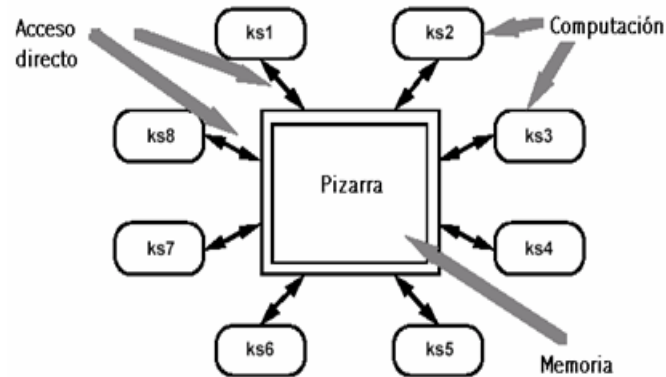


Figura 2. Arquitectura de Pizarra o Repositorio.

Según (Reynoso, y otros, 2004), este estilo arquitectónico se usa en aplicaciones que requieren complejas interpretaciones de procesos de señales (reconocimientos de patrones, reconocimientos de habla), o en sistemas que involucran acceso compartido a datos con agentes débilmente acoplados.

No se considera dicha arquitectura apropiada para la realización del sistema, pues ésta más bien enfatiza los datos y no el modelo, además no son objetivos de este tipo de estilo la reutilización y la flexibilidad del código. En el entorno del sistema o del negocio de los procesos no registrales en los Registros Mercantiles no existen los componentes individuales que actúen sobre la pizarra o el repositorio.

2.1.1.3 Estilos Peer-to-Peer.

Los estilos Peer-to-Peer o también llamada de componentes independientes, enfatiza la modificabilidad por medio de la separación de las diversas partes que intervienen en la computación. Consiste por lo general en procesos independientes o entidades que se comunican a través de mensajes. Algunos de los miembros de esta familia de estilos son las Arquitecturas Basadas en Eventos, Arquitectura Basadas en Servicios y la Arquitectura Basada en Recursos.

A pesar de que los eventos se utilizan para manejar las interfaces de usuario, desde el punto de vista arquitectónico suponen que los métodos y procedimientos no se llaman de forma

directa sino que uno o más de estos métodos puede ser disparado por un evento, de esta forma el módulo ofrece, tanto una interfaz de métodos como una interfaz de eventos. Este estilo más bien se utiliza en ambientes de integración de herramientas, interfaces de usuario o cuando no se quiere que exista un acoplamiento entre el emisor y el receptor.

Entre las desventajas que presentan esta familia de estilos están las siguientes:

- El estilo no permite construir respuestas complejas a funciones de negocio. (Reynoso, y otros, 2004).
- Un componente no puede utilizar los datos o el estado de otro componente para efectuar su tarea. (Reynoso, y otros, 2004).
- Cuando un componente anuncia un evento, no tiene idea sobre qué otros componentes están interesados en él, ni el orden en que serán invocados, ni el momento en que finalizan lo que tienen que hacer. Pueden sugerir problemas de performance global y de manejo de recursos cuando se comparten un repositorio común para coordinar la interacción. (Reynoso, y otros, 2004).

La Arquitectura Basada en Servicios no se acopla al contexto de las oficinas de los Registros Mercantiles, por ser un ente que tiene cierta forma independiente y que debe funcionar pese a problemas de conectividad u otros problemas que surjan.

No se selecciona los estilos Peer-to-Peer, porque el negocio de los procesos que se desean automatizar, tienen sus propias reglas, donde es muy importante el rendimiento del sistema, destacando la flexibilidad y la reusabilidad, además de que existen ciertas condiciones en las que el sistema no podría funcionar, rompiendo así con los esquema propuestos por dichos estilos de arquitectura.

2.1.1.4 Estilo de Llamada y Retorno.

Los estilos de Llamada y Retorno enfatizan la modificabilidad y la escalabilidad, son los estilos más utilizados o más generalizados por los sistemas a gran escala. Dentro de esta familia se encuentran el Modelo-Vista-Controlador (MVC), la Arquitectura Orientada a Objetos y la Arquitectura en Capas.

El **Modelo-Vista-Controlador** (MVC), ha sido propio de las aplicaciones en *SmallTalk* por lo menos desde 1992, antes que se generalizaran las arquitecturas en capas múltiples, según (Reynoso, y otros, 2004), en ocasiones se define más bien como un patrón de diseño como práctica recurrente.

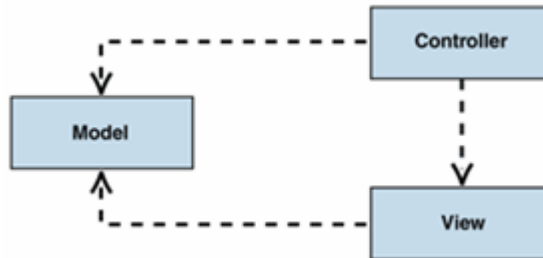


Figura 3. Modelo-Vista-Controlador.

La idea fundamental de este estilo es separar el modelo del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes, de manera tal que puedan ser reutilizados y desarrollados de forma paralela por diferentes personas.

Para el desarrollo del sistema se escogió un Framework creado en la Universidad de las Ciencias Informáticas el cual se basa en el patrón MVC (Modelo-Vista-Controlador), de esta forma se aprovechan las ventajas del patrón y se deja abierta la posibilidad de escoger otra arquitectura diferente, por toda la complejidad del negocio de los procesos antes mencionados; aunque desde el punto de vista arquitectónico la selección de la arquitectura sería diferente. (Almenares, y otros, 2007).

Las **Arquitectura en Capas** constituye uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo. Según (Shaw, y otros, 1996), definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a cada capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior.

Esta arquitectura puede decirse que finalidad es abstraer las funcionalidades de una capa de manera tal que pueda ser totalmente remplazada. De esta arquitectura la más común es la

compuesta por tres (3) capas, *presentación, modelo o reglas del negocio y acceso a datos*. De esta forma se puede reemplazar cualquier capa sin afectar a las otras solamente cambiar las referencias de las implicadas en el cambio. Aunque sea la de tres (3) capas la más común, esto no significa que mientras crezca la complejidad, se mantienen estas tres capas, sino al contrario, si la complejidad crece y es necesario entonces pueden aparecer otras capas para descomponer las funcionalidades que en estas aparezcan. A su vez estas capas pueden estar compuestas por subcapas y una capa o subcapa puede estar compuesta por una o varias clases del diseño.

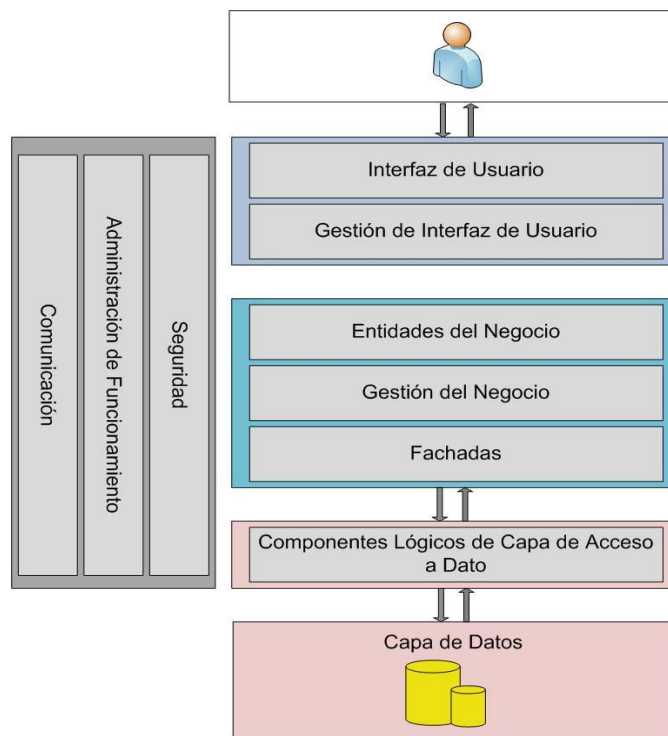


Figura 4. Arquitectura en Capas.

Esta característica de la arquitectura de capas permite implementar las reglas del negocio en una capa aparte, para que estas reglas puedan ser usadas por otros sistemas o por otros servicios que necesiten estas funcionalidades, como es la integración entre dos sistemas dentro del mismo Registro y Notarías, los cuales son el sistema de Mercantil y el sistema de Administración Financiera, todo esto evita la duplicación de código y una mejor organización.

Según (Trowbridge, 2003), las ventajas que presenta la arquitectura de capas son:

- El mantenimiento y las mejoras de la solución son fáciles debido al bajo acoplamiento entre las capas, la alta cohesión de las capas y la habilidad de cambiar su implementación sin cambiar las interfaces. Esto es muy importante para el desarrollo del sistema pues este debe permitir cambios ya sea de requisitos funcionales, agregaciones o mejoras en las funcionalidades.
- Otras soluciones pueden rehusar las funcionalidades expuestas por las diferentes capas, especialmente si las capas de las interfaces son diseñadas con la reutilización en mente.
- El desarrollo distribuido es fácil si este se puede dividir con las capas como fronteras. Esto lo que permite es el desarrollo de forma paralela y posibilitando la especialización de los desarrolladores e incrementando la productividad.
- Distribuir las capas a lo largo de múltiples capas físicas puede mejorar la escalabilidad, tolerancia a errores y rendimiento. Esto lo que quiere decir es que como una capa física puede estar compuesta por una o más computadoras, si se logra distribuir el rendimiento aumenta y de esta forma el desempeño, la durabilidad y la escalabilidad del sistema.
- Beneficios a la hora de realizar las pruebas teniendo bien definidas las interfaces de las capas por la habilidad de cambiar las implementaciones de estas capas manteniendo la interfaz.

También se señalan algunas desventajas en este estilo como son:

- La sobrecarga extra de pasar los mensajes a través de las capas en lugar de llamar los componentes directamente, puede impactar de forma negativa en el rendimiento. Lo cual se puede mitigar con el uso de un modo relajado.
- El desarrollo de las interfaces de usuario puede algunas veces tomar tiempo si la estructura de las capas evita el uso de componentes de interfaz de usuario que interactúan directamente con la Base de Datos. Esto ya está mitigado por la plataforma

de desarrollo que se escogió para el desarrollo del sistema, lo cual la mayoría de los componentes interactúan con DataSet²¹.

- El uso de capas ayuda a controlar y encapsular la complejidad de aplicaciones grandes, pero agrega complejidad a las aplicaciones simples. Como se ha expuesto anteriormente, el sistema que se plantea no es una aplicación simple por lo que es necesario esta encapsulación.
- Cambios en las interfaces de las capas inferiores tienden a propagarse a los altos niveles, especialmente si el modo relajado es usado.

El modo relajado es una de las formas de concebir la arquitectura en capas, se le llama modo relajado cuando las capas superiores pueden interactuar con las capas inferiores, esto aumenta el rendimiento pero implica menos flexibilidad en la aplicación, mientras que la otra forma de concebir esta arquitectura es que una capa este ligada únicamente con la capa inferior inmediata, esta mantiene un bajo acoplamiento pero puede implicar impactos negativos en el rendimiento, según (Almenares, y otros, 2007).

Ésta fue la arquitectura que fue escogida para el desarrollo del sistema, por todas las ventajas que posee en comparación con otros estilos antes mencionados y el hecho de que se combine con el patrón Modelo-Vista-Controlador esto obtiene resultados muy satisfactorios en el desarrollo del sistema.

Resaltar que existen otros tipos de estilos arquitectónicos que aquí no se mencionaron porque no correspondían sus aplicaciones con el sistema que se pretende desarrollar, para estudiar estos estilos más a profundidad, y los que no se mencionaron aquí pues se puede ver (Trowbridge, 2003).

2.1.2 Framework para la Gestión de la Capa de Presentación.

Básicamente un framework evita construir una aplicación desde cero (0), típicamente un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para

²¹ Según (Bipin), es una estructura de datos del Microsoft .Net framework que encapsula un conjunto de datos y soporta un modelo relacional de una o más tablas.

ayudar a desarrollar y unir los diferentes componentes de un proyecto, también se debe tener en cuenta que para comenzar a desarrollar un framework se debe conocer sus especificaciones. (Almenares, y otros, 2007).

El framework escogido como se mencionó antes fue el desarrollado por el proyecto de Identidad de la Universidad de las Ciencias Informáticas, el cual tiene elementos del patrón Modelo-Vista-Controlador (MVC). Estas ideas quedan encapsuladas en la capa de presentación por ser la que está compuesta por los elementos de este patrón. El uso de este framework no cambia para nada el estilo escogido para el desarrollo de la aplicación, el uso de éste es solo para la gestión de la capa de presentación.

Se analizarán las ventajas y desventajas del uso de este framework, algunas de ellas son propias del patrón MVC y otras de la implementación en concreta de este.

2.1.2.1 Ventajas.

Entre las ventajas del framework escogido están las siguientes:

- *Múltiples vistas usando el mismo modelo.* Dado que las vistas se encuentra separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. (Reynoso, y otros, 2004). En el framework el modelo y el controlador se pueden unir, pero si los desarrolladores tienen esto en cuenta, separando los dos componentes esta ventaja se puede aprovechar.
- *Fácil para soportar nuevos tipos de clientes.* Para soportar nuevos tipos de clientes solo se debe escribir la vista y el controlador para este y se reutiliza el modelo ya existente. (Dass). Esto es muy importante si alguna vez se decide migrar el sistema a software libre.
- *Claridad de diseño.* Mirando la lista de métodos públicos, debe ser fácil entender cómo controlar el funcionamiento del modelo. (Dass). Esto depende además en lo complejo

de la solución, a medida que el software crece es necesario ir desarrollando otros diagramas como apoyo.

- *Fácil de crecer.* Los controladores y las vistas pueden crecer según crece el modelo, viejas versiones de las vistas y los controladores pueden usarse mientras que la interfaz común es mantenida. (Dass).
- *Seguridad a nivel de aplicación.* El framework sienta las bases para brindar seguridad a nivel de aplicación, mediante un fichero XML²² de configuración donde se definen roles y en base a estos se gestiona el acceso a las acciones, las acciones constituyen en el framework de cierta forma los controladores, aunque pueden tener otras funciones.
- *Colores de interfaz configurable.* Se puede cambiar el color de toda la aplicación con solo unos pocos cambios en la configuración de éste.
- *Configuración del Menú de Opciones.* Mediante un fichero XML de configuración se puede cambiar la estructura del menú de opciones, donde cada opción puede contener opciones y así sucesivamente por varios niveles.
- *Múltiples controladores usando la misma vista.* Debido a que la vista está completamente separada del controlador y no se conoce, sino que al revés, la misma vista puede ser utilizada por distintos controladores, además permite que la vista sea modificada en su estructura y diseño sin afectar a los controladores.

2.1.2.2 Desventajas.

Entre las desventajas del framework escogido se pueden ver las siguientes:

- *Complejidad.* Incrementa la naturaleza de basada a eventos del código de la interfaz de usuario, lo cual puede ser más fácil de depurar. (Trowbridge, 2003).

²² Extensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

- *Acoplamiento de la Vista y el Controlador con el Modelo.* Cambios en la interfaz del modelo requieren cambios paralelos en la vista y cambios adicionales en el controlador, algunos de estos cambios del código pueden llegar a ser bastante difíciles.
- *Interfaz.* El framework posee una interfaz al estilo Web donde en la parte derecha se muestra el menú con todas las opciones, si se desea cambiar esta apariencia puede conllevar un gran esfuerzo por el equipo de desarrollo, partiendo desde el punto que se cuente con el código del mismo, de lo contrario es imposible.

2.2 Estándar de Codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento y mantenimiento del software.

En el caso de Registros y Notarías, el estándar de codificación fue definido por la dirección del proyecto en sus inicios, se define el formato para los siguientes puntos (ALBET, Ingeniería en Sistemas, Octubre 10, 2006):

- Organización de los ficheros.
- Indentación.
- Comentarios.
- Declaraciones.
- Espacios en blanco.
- Convenciones de declaración.

- Prácticas de programación.

Una vez visto y explicado las bases sobre las cuales se construyó el sistema se presentarán algunos de los patrones de diseño que más significación tuvieron en la construcción de la aplicación, para su mayor flexibilidad en todos los procesos.

2.3 Patrones de Diseño.

Los diseñadores expertos en orientación a objetos y también otros diseñadores de software, van formando un amplio repertorio de principios generales y de expresiones que los guían al crear software. A unos y a otras podemos asignarles el nombre de *patrones*, si se codifican en un formato estructurado que describe el problema y su solución, y si se les asigna un nombre. (Larman, 1999).

Según (Larman, 1999), en la terminología de objetos, el *patrón* es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas.

Entre los patrones de diseño que se aplicaron durante el desarrollo del proyecto están los siguientes:

- **Patrón Fachada.** Según (Gamma, y otros, 1995), el patrón fachada es promover una interfaz unificada para un conjunto de interfaces en un subsistema, haciendo a éste fácil de usar.

Este patrón se usa muy a menudo para las operaciones que se realizan sobre determinadas entidades del negocio, por ejemplo existe una *fachada* para las operaciones del trámite y otra para las operaciones sobre los documentos jurídicos. Las fachadas se ven con más facilidad para el traspaso de datos y la encapsulación a través de las capas.

- **Patrón Mediador.** Según (Gamma, y otros, 1995), el patrón mediador tiene como intención encapsular la interacción entre los objetos, promoviendo el bajo acoplamiento de manera tal que estos puedan variar independientemente.

Este patrón se usa generalmente para la transformación de los datos que se devuelven como estructuras de datos desde la base de datos, generados por el TierDeveloper, a las entidades del

negocio que son las que interactúan con las capas superiores. De esta forma las entidades que genera el TierDeveloper podrían variar sin afectar a las clases del modelo.

- **Patrón Método Plantilla.** Según (Gamma, y otros, 1995), este patrón define el esqueleto de un algoritmo dejando las especificidades a las subclasses y permitiendo que estas redefinan ciertos pasos del algoritmo.

Este patrón es uno de los más utilizados para la reutilización de una misma vista por uno o más controladores, acciones según las define el framework. De esta forma el controlador base podría definir el comportamiento para las operaciones comunes y dejaría a las subclasses que respondieran con sus propias implementaciones a las que debería realizar.

De manera general estos fueron los patrones más utilizados, pero ello no excluyen que se hayan utilizados otros patrones con menos intensidad que los antes mencionados, como fueron el *Solitario*, *Estrategia*, *Métodos de Fabricas* y *Estado*.

Después que se analizó los patrones de diseño más utilizados en el desarrollo del sistema, pues se analizará como está compuesta la distribución de clases por cada capa según la arquitectura antes mencionada.

2.4 Características fundamentales de los procesos que complementan al proceso de inscripción.

En este epígrafe se verá como está compuesto el negocio de los procesos solicitudes de copias, solicitudes de sellado de libro y las solicitudes de expedientes, como parte de la solución que se pretende desarrollar.

2.4.1 Descripciones de los procesos.

2.4.1.1 Descripción del proceso Solicitudes de Copias.

El proceso de solicitudes de copias se divide en diferentes flujos de trabajo, empezando por la realización de la solicitud hasta la entrega de las copias al usuario según el tipo seleccionado,

ya sea simple o certificada. Los flujos por los que pasa esta solicitud son: Solicitud de Copias, Solicitudes Pendientes, Solicitudes Listas y como opcional tenemos Editar Solicitud de Copia.

El proceso comienza cuando un usuario solicita al funcionario de *Procesar Copias* en el departamento de Archivo la copia de algún documento, éste elige la opción Solicitud de Copia. En este momento primero se selecciona el expediente al que pertenece dicho documento, se hace una búsqueda del documento al cual se le quiere realizar la copia, después el sistema captura los datos del usuario y solicita que se le especifique el tipo de copia que se quiere realizar: Copia Certificada o Copia Simple, ya sea mecanografiada o fotostática, y los datos de dicha solicitud, entre los que se encuentra la cantidad de copias que se van requieren, se le introducen los datos donde está ubicado el documento a realizarle la copia, el sistema le genera la fecha límite para la entrega de las copias y le da la posibilidad de generar el cálculo en el mismo flujo, después de aquí se guarda dicha solicitud y se imprime una Planilla de Constancia de la Solicitud de Copia del expediente y una Planilla Única Bancaria para saldar la deuda del trámite en el banco.

El funcionario de *Procesar Copias* solicita al sistema realizar búsqueda de los trámites pendientes para procesarles las copias, esto es en la opción Solicitudes Pendientes. El sistema realiza la búsqueda de todos los trámites pendientes y finalizada ésta selecciona un trámite de la lista para darle curso, el cual para este paso no puede tener recaudos pendientes o lo que es lo mismo, el usuario debe haber cancelado el monto calculado y especificado en la Planilla Única Bancaria. En este momento se hace la solicitud del expediente que contiene el documento al cual se le va a hacer la copia, si no se solicitó en el paso anterior, después de que el funcionario de Procesar Copia tiene el expediente, el sistema muestra un resumen de los datos del trámite, y si la copia que se solicitó fue una copia simple, el usuario solicita al sistema actualizar el trámite como finalizado, en caso de que la copia sea certificada, se le muestran las planillas que certifican la copia, más el sello que certifica la copia correspondiente, solicitando al sistema que actualice el trámite como finalizado, el sistema actualiza el Libro Diario de Copias y da finalizado el trámite en este paso.

El último paso que se realiza para el proceso de Solicitudes de Copias es: Solicitudes Listas, donde el sistema muestra una interfaz de búsquedas para los trámites que estén listos para

ser entregados al solicitante, el funcionario del departamento de Copias selecciona el trámite y solicita la entrega al sistema, este actualiza el estado del trámite para entregado, y es aquí donde se le entregan las copias solicitadas al cliente.

En el negocio de copias existe un paso intermedio que es una opción para las solicitudes de copias, la cual se pueden editar, es un paso opcional y se debe hacer antes de Procesar las Copias, cuando se procesen las copias no se podrá editar ninguna solicitud que se haya hecho, es en el paso Editar Solicitud de Copia. El sistema le da la posibilidad de buscar los trámites a los que se desea modificar los datos, estos siempre serán los que no han sido procesados, el sistema le brinda los datos a modificar, el usuario los modifica y actualiza la solicitud con los cambios de la misma y se realiza un re-cálculo asociado a la modificación, para que salde los montos en el banco especificado.

2.4.1.2 Descripción del proceso Solicitudes de Sellado.

Este proceso tiene como objetivo sellar los diferentes tipos de libros que son generados en las compañías, como son: Libro de Accionistas para Compañías Anónimas y Sociedades Anónimas, Libro de Compras, Libro de Contabilización como el Mayor, el SENIAT, el Libro de Socios para Sociedades de Responsabilidades Limitadas, Libros de Ventas, en fin este es el proceso mediante el cual adquieren validez legal los libros que se llevan en una empresa o compañía en la República Bolivariana de Venezuela.

Se inicia el proceso cuando un cliente de alguna de las compañías venezolanas se presenta en el Registro Mercantil para realizar el sellado de uno o más de los libros de su empresa, el sistema le brinda varias opciones por las que va a pasar el nuevo trámite de solicitud de sellado, empezando por Solicitar Sellado de Libro, pasando por Sellar Libros hasta que se finaliza con Entregar Sellos, también existe un paso que es opcional en el proceso que es el de Editar Solicitud de Sellado.

Como se expresó anteriormente se comienza por el paso de Solicitar Sellado de Libro, donde se solicita la búsqueda del expediente de la compañía que desea hacer la solicitud de sellado, después de seleccionado el expediente el sistema brinda una interfaz de usuario para

seleccionar el solicitante del trámite, que es un miembro de dicha empresa, después de haber validado esto el sistema se encarga de entrar los datos de la solicitud de sellado, compuesta por una colección de libros a ser sellados y de cada uno se especifica el tipo de libro y la cantidad de folios con que cuenta, después de esto se le especifica al usuario la fecha límite que tiene para recoger los libros ya sellados en el registro y el funcionario solicita al sistema generar el cálculo para esta operación, el cual arroja los montos que el cliente deberá cancelar en el banco por el servicio solicitado, concluyendo así la solicitud de sellado y el trámite pasaría a la siguiente fase, procesar dicha solicitud.

Cuando el funcionario solicita Sellar los Libros, se realiza la búsqueda de todos los trámites pendientes para este paso, es requisito indispensable que el usuario haya cancelado los montos calculados en el paso anterior, sino no se continúa con el trámite, si cumple esta condición el funcionario del departamento de Sellado de Libros solicita al sistema generar los sellos para dicha solicitud, los cuales se imprimen junto a la constancia de que ha pasado por este paso, seguidamente el sistema actualiza el libro de control del sellado de los libros.

El próximo y último paso de este proceso es la Entrega de Libros, que comienza realizando una búsqueda de todos los tramites que están pendientes a ser entregados, donde el usuario debe seleccionar cuál de ellos es el que va a entregar y lo entrega, donde se muestra una interfaz de resumen del trámite, y se confirma la entrega de los libros al cliente o solicitante, el cual recibe los libros con sus respectivos sellos, concluyendo así el proceso de solicitudes de sellados de los libros de comerciantes.

Se cuenta también en el caso de Sellado de Libros con el paso opcional de: Editar Solicitud de Sellado, dicha edición se debe hacer antes de procesar la solicitud o sellar los libros y comienza cuando se realiza una búsqueda y selección de los trámites que van a ser modificados, se solicita al sistema la edición de del trámite seleccionado, el cual captura los datos modificados y actualiza los cambios realizados en la Base de Datos y se solicita hacer un re-cálculo asociado a la actualización de dicha solicitud, el cual genera montos que deberán ser cancelados por el usuario antes de entrar en la siguiente fase: Sellar los Libros.

2.4.1.3 Descripción del proceso Solicitudes de Expedientes.

El proceso de solicitudes de expedientes se divide en dos subprocesos diferentes, uno de estos es cuando un usuario desea hacer una solicitud de algún expediente al Registro Mercantil en calidad de préstamo; este proceso se le denomina Solicitud Formal, el otro subproceso es cuando un funcionario del registro necesita hacer una solicitud de un expediente como complemento a alguna de las funciones que realiza, como por ejemplo en el caso de Revisión Legal, Solicitudes de Copias, etc. A este proceso se le denomina Solicitud Interna. A continuación se explicará cómo se ejecutan estos procesos y cuáles son los requisitos para su puesta en marcha.

2.4.1.3.1 Descripción del proceso Solicitud Formal.

En el proceso de Solicitud Formal pasa por dos pasos fundamentales, el primero es cuando se realiza la solicitud, en Solicitud Formal y el otro paso es cuando se devuelve el expediente al Archivo, en Devoluciones.

El proceso comienza cuando el usuario solicita un expediente dado, el sistema brinda la posibilidad de realizar una búsqueda de dicho expediente, donde se solicita una relación de las piezas asociadas a este, brindando la información si esta pieza está disponible para su préstamo o no, el sistema solicita la captura de los datos de la persona que está realizando la solicitud porque ésta será la responsable de devolver el expediente en buen estado y es quién responderá por cualquier daño que el expediente reciba. Se muestra una interfaz con un resumen de la solicitud, en la cual se acepta esta solicitud y el expediente se le entregará al cliente, actualizando el expediente como accedido.

Este es el estado en el que estará el expediente hasta que sea devuelto. Es muy importante tener en cuenta que ya que no se podrá realizar ninguna solicitud sobre esa pieza del expediente mientras se encuentre fuera del Archivo.

Después que el cliente termina de revisar o consultar el expediente al cual él accedió, se presenta para su devolución al departamento de Archivo, el funcionario comprueba que el expediente esté en buen estado y entonces se realiza la operación de devolución en el

apartado: Devoluciones, en donde se realiza una búsqueda que arroja como resultados todos los trámites de solicitudes externas, seleccionando el que se le va a dar curso, este es el momento donde se actualizará el expediente como devuelto y se podrán volver a realizar solicitudes sobre la pieza en cuestión.

2.4.1.3.2 Descripción del proceso Solicitud Interna.

El funcionario de Solicitudes Internas en el departamento de Archivo puede realizar dos tipos de operaciones fundamentales, una de ellas es Aceptar Solicitudes Internas y la otra es la de Devolver Expedientes Solicitados.

Para poder Aceptar Solicitudes Internas, se tuvo que en algún proceso anterior haber solicitado el expediente, como se hace en Revisión Legal, en Solicitudes de Copias o en Sellado de Libros, en donde el sistema brinda una serie de informaciones del expediente que se está solicitando para que el usuario de Solicitudes Internas en Archivo lo acepte. Cuando pasa esto el trámite se actualiza como *Liberado Por Archivo*, dando paso al funcionario que emitió la solicitud la opción de que acceda al expediente. Cuando el solicitante termine de usar el expediente requerido este debe emitir una actualización del estado para que el usuario de Solicitudes Internas en Archivo solicite la devolución del expediente solicitado y el sistema actualice el estado como *Aceptado Por Archivo*.

Durante todo este proceso de solicitar el expediente por un funcionario del registro hasta que él envía la solicitud de entrega del mismo, este será bloqueado por la aplicación, sin dar oportunidad que se realizan operaciones sobre dicho expediente, solo se podrá acceder para realizar cualquier operación sobre dicho expediente cuando el usuario de Archivo lo acepte en Devoluciones y se actualice el estado del expediente.

2.5 Conclusiones.

En este capítulo se hizo un análisis de varios de los estilos arquitectónicos más conocidos para la realización de sistemas similares. El resultado de este análisis comprobó que la Arquitectura en Capas era la más factible para la realización del sistema, por todas las cualidades que tenía el proyecto; como son las peticiones de cambios que vienen por parte de los clientes, por lo que el sistema debe tener

mucha flexibilidad, para que estos cambios constantes no afecten el tiempo de entrega del producto, además de todos los argumentos presentados en la explicación en el epígrafe correspondiente.

También se realizó el análisis de una serie de patrones para que se pudiera definir un diseño apropiado para el sistema, donde se explicó cuáles eran los candidatos más favorables para el desarrollo constante de la aplicación, donde también se planteó que existirán otros patrones para darle seguimiento a algún proceso en específico.

Por último para una definición completa del diseño se explicó de manera general las funcionalidades o procesos más importantes para el diseño y la implementación, de manera que se tuvieran los elementos y artefactos necesarios para la realización de la propuesta del sistema.

Una vez vistos estos aspectos se define en el siguiente capítulo la solución de diseño para la realización de sistema donde se podrán observar expresados dichos patrones y a todo lo que se planteó en el actual capítulo.

CAPÍTULO 3. Diseño e Implementación.

El diseño del sistema se ocupa de desarrollar las directrices propuestas durante el proceso de análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados, tanto del punto de vista funcional como no funcional, guiándose para este por los requerimientos levantados en dicha etapa.

Según (Rational Software Corporation, 2002), el proceso de diseño de un sistema complejo se puede realizar de forma descendente como se muestra a continuación:

- Diseño de alto nivel (descomposición del sistema a diseñar en subsistemas menos complejos).
- Diseño e implementación de cada uno de los subsistemas.
- Especificación consistente y completa del subsistema de acuerdo con el objetivo establecido en el análisis.
- Desarrollo según las especificaciones.
- Pruebas.
- Integración de todos los subsistemas.
- Validación del diseño.

En el transcurso de este capítulo se verán algunos de los aspectos antes mencionados, donde se explicarán y se presentarán una serie de diagramas pertenecientes al diseño e implementación del sistema, también se verán algunas de las pruebas que se le hicieron al software, llegando éste a su etapa final, la terminación del producto y la aceptación por parte del cliente.

3.1 Modelo de Diseño.

El diseño de software orientado a objetos es difícil, y el diseño de software reutilizable orientado a objetos lo es aún más. Se deben identificar los objetos persistentes, clasificarlos dentro de las clases con la granularidad correcta, definir interfaces de clases, jerarquías de herencia y establecer relaciones claves dentro de ellos. El diseño debe ser específico al problema que se tiene entre manos, pero suficientemente general para adaptarse a problemas y requerimientos futuros. Se debe evitar el rediseño o por lo menos

minimizarlo. Antes de que un diseño sea terminado, usualmente se tratan de reutilizar este, varias veces, modificándolo cada vez. (Gamma, y otros, 1995).

3.1.1 Capas y Subsistemas en el Modelo de Diseño.

El diseño de los procesos que complementan al proceso de Inscripción, solicitudes de expedientes, solicitudes de copias y solicitudes de sellado de libros, se dividen en varias capas o subsistemas, las cuales son:

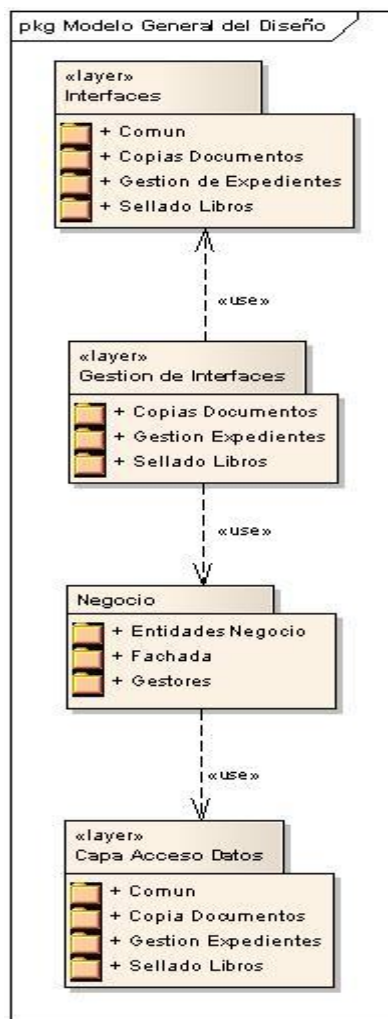


Figura 5. Modelo General del Diseño.

- **Capa de Presentación.**

Esta capa estará compuesta por dos subcapas, la cual se explicarán seguidamente:

- **Capa de Interfaz.**

Su uso se basa generalmente en la explotación de los componentes de interfaz que realmente provee un framework antes explicado facilitando el desarrollo del sistema, donde se corresponde con funcionalidades de visualización de datos a través de formularios de presentación.

- **Capa de Gestión de Interfaz.**

Su uso se basa generalmente en la explotación de los componentes de interfaz que realmente provee el framework facilitando el desarrollo de sistema basado en acciones, donde una acción básicamente define un bloque de código reusable por varias operaciones sobre el sistema. Esta no es más que una clase que representa la ejecución de alguna tarea concreta y no debe ser excesivamente compleja.

- **Capa de Negocio.**

Dentro de esta capa existen otras capas que completan y organizan toda la lógica de negocio y los datos que se manejan en el sistema. La capa de Gestores engloba todas las clases encargadas de la persistencia y la realización de funciones para que el negocio funcione correctamente. Otra de las subcapas que existen es la de Entidades del Negocio, que consta de clases que son principalmente persistentes en la base de datos, contenedores de información. La última subcapa que está presente es la Fachada, la cual le da una independencia total a la capa de negocio con la de acceso a datos, brindándole a esta un bajo acoplamiento e independencia de la tecnología que se utilice para la capa de acceso a datos.

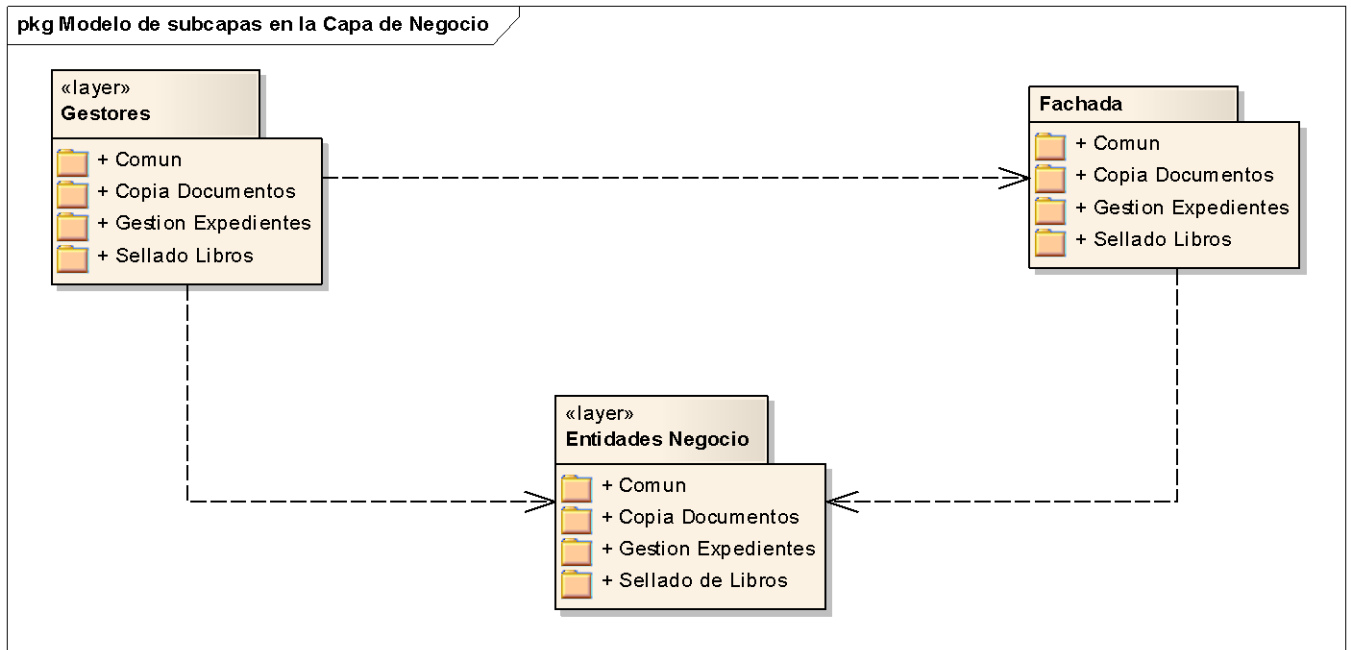


Figura 6. Interrelación entre las subcapas de la Capa de Negocio.

- **Capa de Acceso a Datos.**

Esta capa fue generada por el TierDeveloper v4.0, la cual es una herramienta para la generación de capas de acceso a datos. Ella garantiza que la información fluya desde el negocio hasta la base de datos y viceversa. Esta está formada por dos (2) tipos de clases principalmente, las entidades mapeadas de la Base de Datos, denominándosele Entidades, otras de las clases que se generan se denominan Factorías, que son las que gestionan todas las funcionalidades que se realizan en la base de datos sobre las entidades, ya sean procedimientos almacenados o vistas.

3.1.2 Diagramas de Clases.

A continuación se describen los diagramas de clases, en este caso se representan mediante el Diagrama de Clases. Cada subepígrafe de los que a continuación se muestran contiene el modelo de cada uno de los procesos antes mencionado.

3.1.2.1 Diagramas de Clases de Gestión de Expedientes.

Los diagramas de clases de *Gestión de Expedientes* se dividen en dos procesos diferentes, como se explicó en el capítulo anterior, (Ver [Descripción del proceso Solicitudes de Expedientes.](#)).

Los diagramas de clases en Solicitud Formal se dividieron por los dos casos de usos como se muestra a continuación:

- *Solicitud de Expedientes.*
- *Devoluciones de Expedientes.*

Al igual los diagramas de clases de Solicitud Interna se dividieron cinco (5) casos de usos:

- *Realizar Solicitud de Expediente.*
- *Aceptar Solicitud.*
- *Realizar Préstamo.*
- *Devolver Expediente.*
- *Recibir Expediente.*

A continuación se muestra, un diagrama de clases de los diferentes casos de usos que surgieron en *Gestión de Expedientes*, el cual refleja todas las clases con sus relaciones, ya sean clases de entidades del negocio, representadas por <ceNombreEntidad>, las clases controladoras de interfaz, representadas por <accNombreClase>, las clases de interfaz, representadas por <frmNombreInterfaz>, y las clases gestoras de negocio, representadas por <ccNombreGestor>.

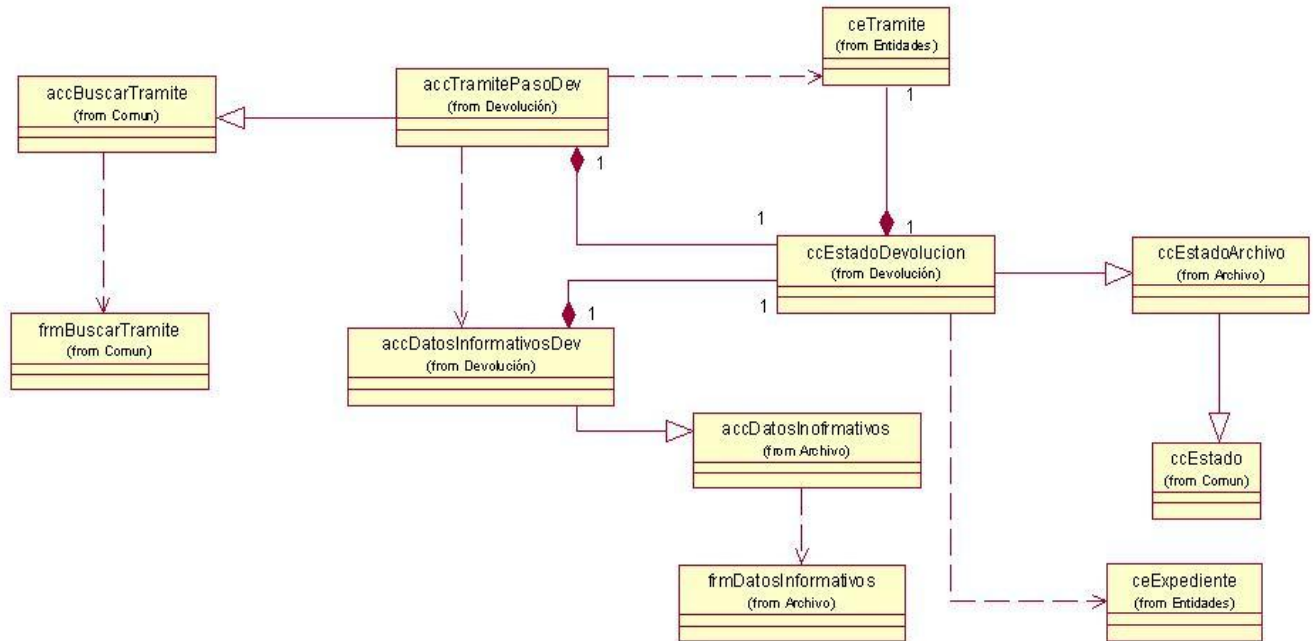


Figura 7. Diagrama de Clases. CU: Devoluciones de Expedientes.

Para más información de los diagramas de clases presentados en la Gestión de Expedientes (Ver [Anexo 1.](#))

3.1.2.2 Diagramas de Clases de Gestión de Sellado de Libros.

Los diagramas de clases de *Gestión de Sellado de Libros* se dividen cuatro (4) casos de usos, fundamentalmente como son:

- *Solicitar Sellado de Libro.*
- *Editar Sellado de Libro.*
- *Sellar Libros.*
- *Entrega de Libros.*

A continuación se muestra un diagrama de clases de los diferentes casos de usos que surgieron en este proceso, el cual refleja todas las clases con sus relaciones, ya sean clases

de entidades del negocio, clases controladoras de interfaz, clases de interfaz, y las clases gestoras de negocio.

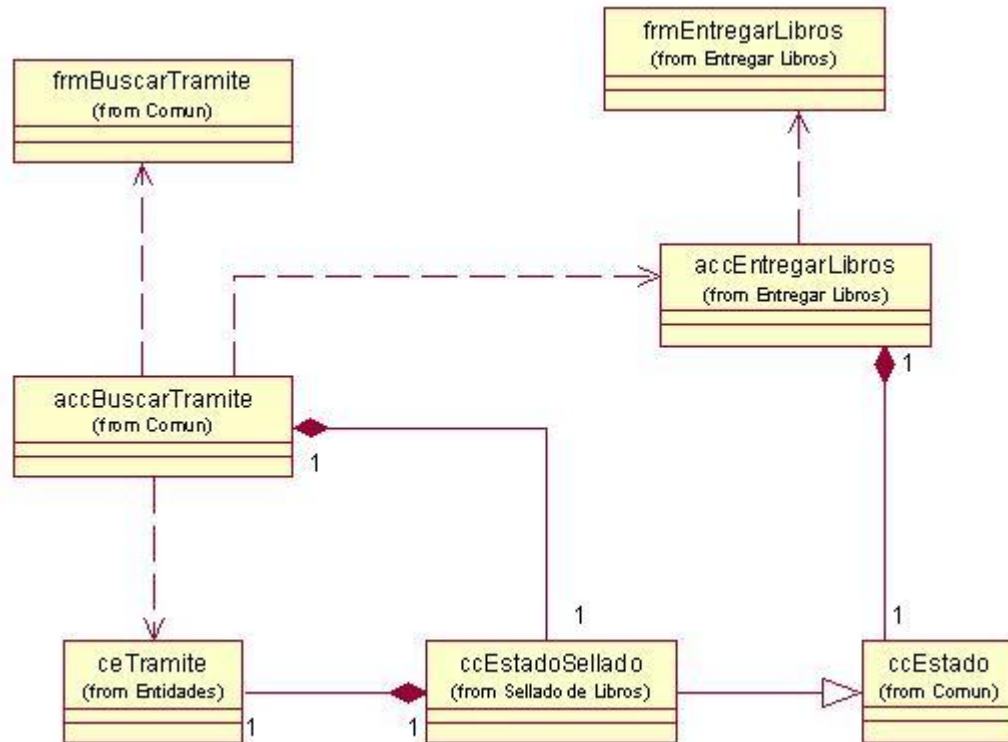


Figura 8. Diagrama de Clases. Entrega de Libros.

Para más información de los diagramas de clases presentados en la Gestión de Sellado de Libros ([Ver Anexo 2.](#)).

3.1.2.3 Diagramas de Clases de Gestión de Copias de Documentos.

Los diagramas de clases de *Gestión de Copia de Documentos* se dividen cuatro (4) casos de usos, fundamentalmente como son:

- *Solicitud de Copia de Documentos.*
- *Editar Solicitudes de Copia de Documentos.*
- *Solicitudes de Copia de Documentos Pendientes.*

- *Solicitudes de Copia de Documentos Listas.*

A continuación se muestra un diagrama de clases de los diferentes casos de usos que surgieron en este proceso, el cual refleja todas las clases con sus relaciones, ya sean clases de entidades del negocio, clases controladoras de interfaz, clases de interfaz, y las clases gestoras de negocio.

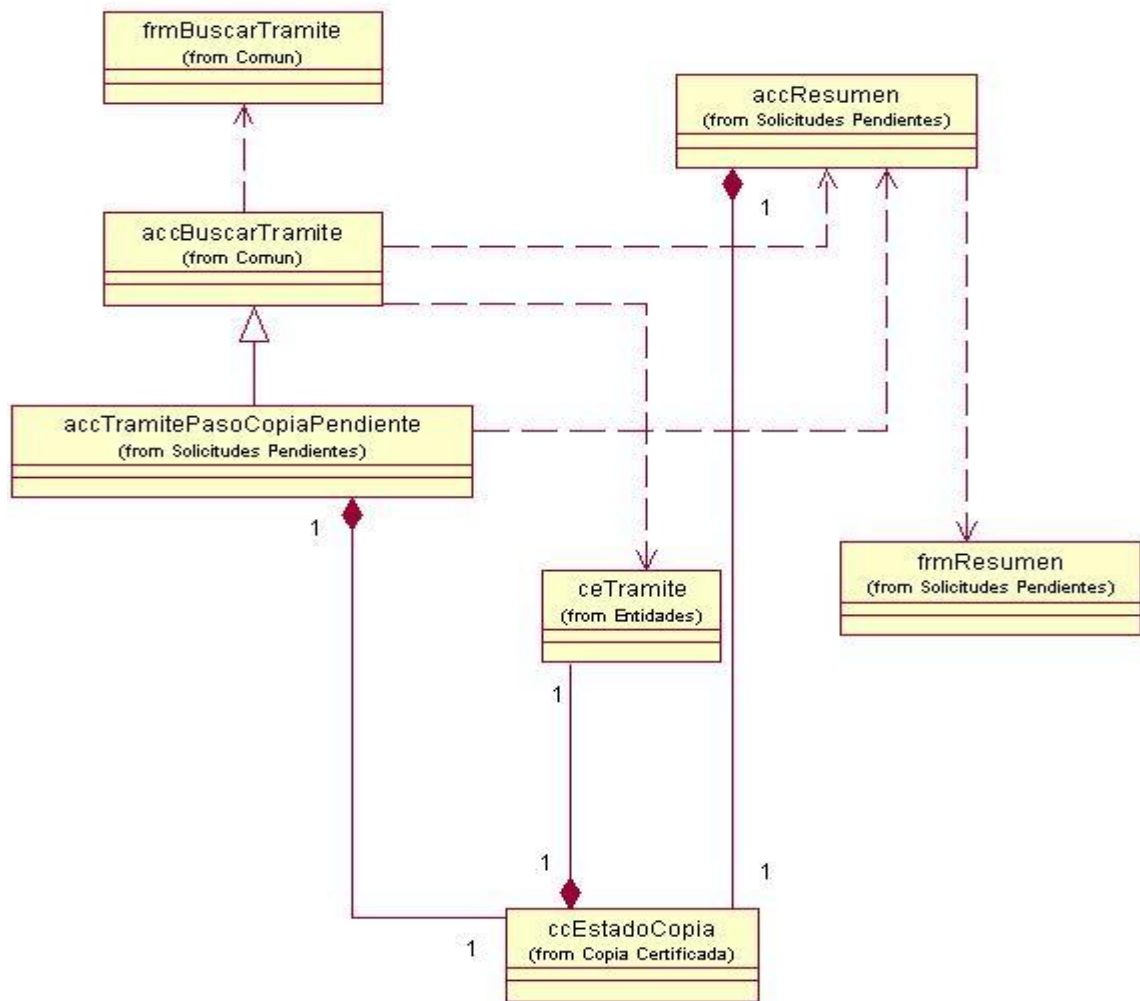


Figura 9. Diagrama de Clases del CU: Solicitudes de Copia de Documentos Pendientes.

Para más información de los diagramas de clases presentados en la Gestión de Copia de Documentos (**Ver [Anexo 3.](#)**).

3.1.3 Diagramas de Interacción.

A continuación se describen los diagramas de interacción, en este caso se representan mediante el Diagrama de Secuencia. Cada subepígrafe de los que a continuación se muestran, contiene el modelo de cada uno de los procesos antes mencionados.

3.1.3.1 Diagramas de Secuencias de Gestión de Expedientes.

Los diagramas de secuencia de *Gestión de Expedientes* se dividen en los dos procesos antes explicados, aquí se generan varios de estos artefactos, para la total realización de estos, llevando a una mejor implementación de la solución. (**Ver [Anexo 4.](#)**).

El presente diagrama de secuencia tienen como carácter general realizar un control de todos los expedientes que hayan sido accedidos en un Registro Mercantil, esto tiene un gran valor, sobre todo informativo y de control para ver quien, cuando y de qué forma se accedió a un expediente determinado.

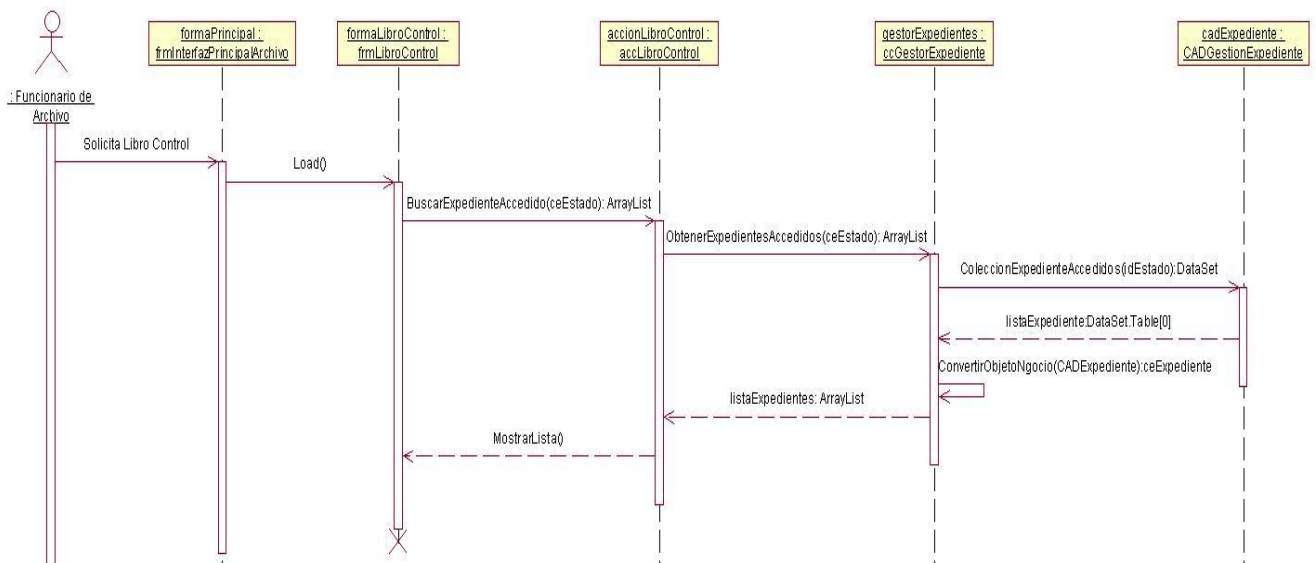


Figura 10. Diagrama de Secuencia del CU: Control de Expedientes Accedidos.

3.1.3.2 Diagramas de Secuencias de Gestión de Sellado de Libros.

Los diagramas de secuencia de *Gestión de Sellado de Libros* se dividen por casos de usos y cuando estos son muy grandes pues se tratan de dividir por escenarios, llevando esto a una mejor implementación de la solución y mayor entendimiento entre desarrolladores y diseñadores de software. (Ver [Anexo 5](#)).

El siguiente diagrama de secuencia es del caso de uso Entrega de Libros Sellados es uno de los pasos más importantes en este proceso porque es cuando se imprimen varios sellos que son los que le dan validez legal a los libros de las compañías, cabe señalar que la entrega de libros se realiza cuando los procesos anteriores a este han sido culminados exitosamente.

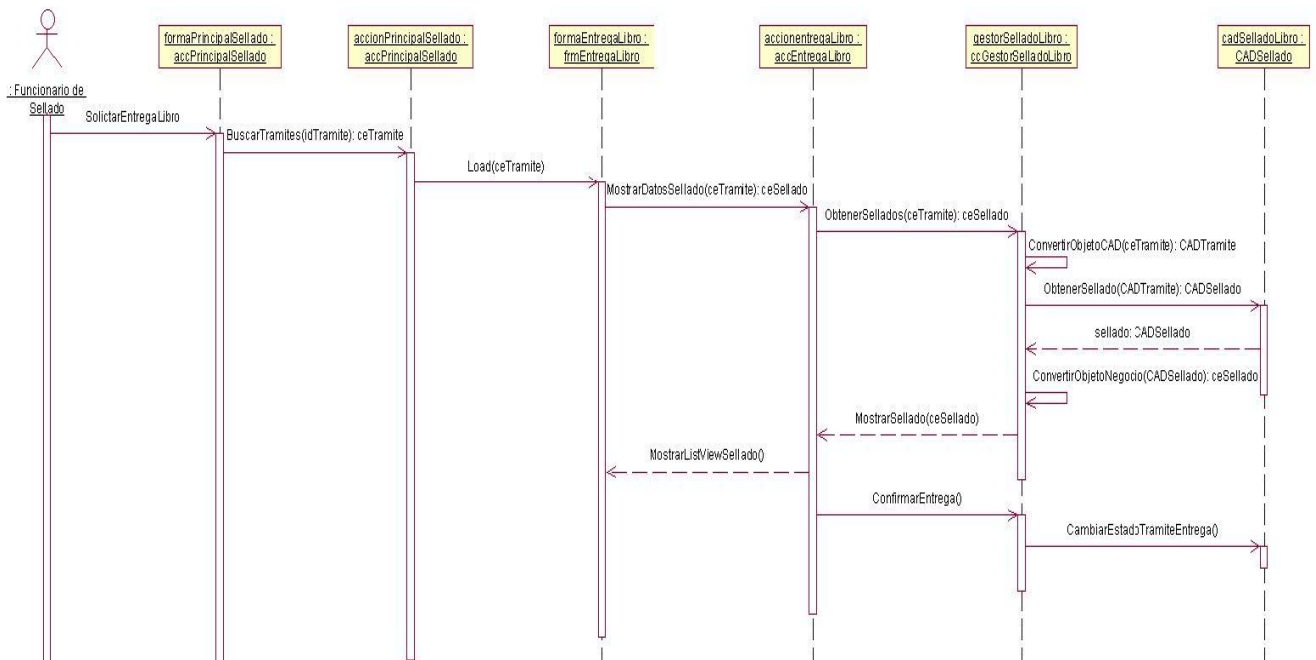


Figura 11. Diagrama de Secuencia del CU: Entrega Libros Sellado.

3.1.3.3 Diagramas de Secuencias de Gestión de Copia de Documentos.

Los diagramas de secuencia de *Gestión de Copia de Documentos*, se dividen por casos de usos y cuando estos son muy grandes pues se tratan de dividir por escenarios, llevando esto a una mejor implementación de la solución y mayor entendimiento entre desarrolladores y diseñadores de software. (Ver [Anexo 6](#)).

El siguiente diagrama de secuencia es del caso de uso Editar Copia Documento, este es un paso auxiliar en este proceso, no obstante no deja de ser importante para aplicación y mucho más para el usuario, porque esto genera montos que él tiene que ir a cancelar al banco, es por esto que si el funcionario se equivoca en la entrada de alguna copia, éste le brinda la oportunidad de cambiarla, veremos cómo es que interactúan todos los procesos en el diagrama que se le muestra a continuación.

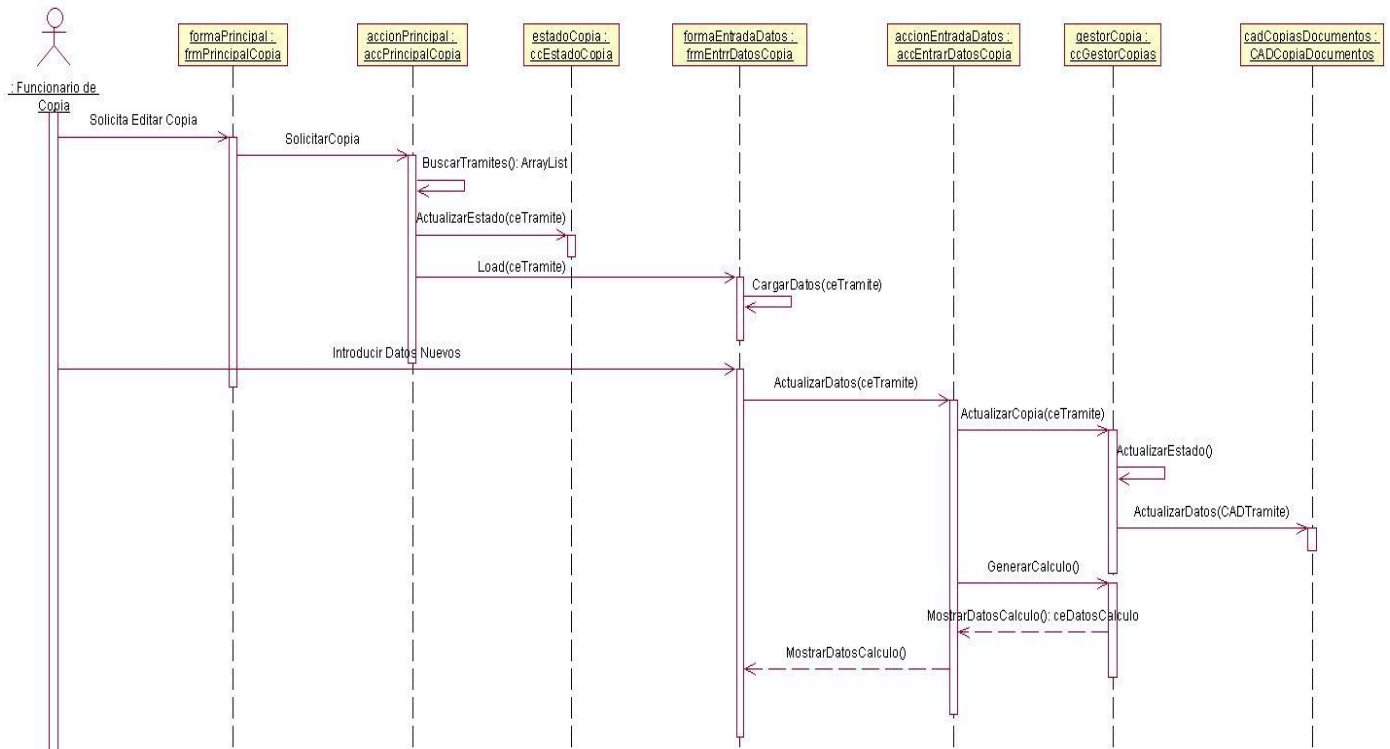


Figura 12. Diagrama de Secuencia del CU: Editar Solicitud de Copia de Documentos.

Después que se describieron de forma general los artefactos que genera el proceso de diseño, en el software se verá como éste influyó para la implementación del sistema, a continuación se muestra cómo se comportan los componentes en el modelo de implementación.

3.2 Modelo de Implementación.

En este modelo se implementa el sistema diseñado en términos de componentes, ficheros de código fuente, ficheros de códigos binarios, ejecutables, entre otros.

Según (Almenares, y otros, 2007), la implementación sigue varios propósitos consigo como se muestra a continuación:

- Planificar las integraciones necesarias del sistema en cada iteración. Se sigue un enfoque incremental dando lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignando componentes ejecutables a nodos en el Diagrama de Despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño. Las clases se implementan como componentes de ficheros que contienen código fuente.
- Probar los componentes individualmente y a continuación integrarlos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados, donde seguidamente se llevan a cabo las comprobaciones o prueba de unidad del sistema.

3.2.1 Diagramas de Componentes.

A continuación se presenta el Diagrama de Componentes propuesto para la solución del software, los componentes no son más que el empaquetamiento físico de los elementos de un modelo, como el Modelo de Diseño, donde presentan varios estereotipos como ficheros ejecutables, ficheros de código fuente, entre otros.

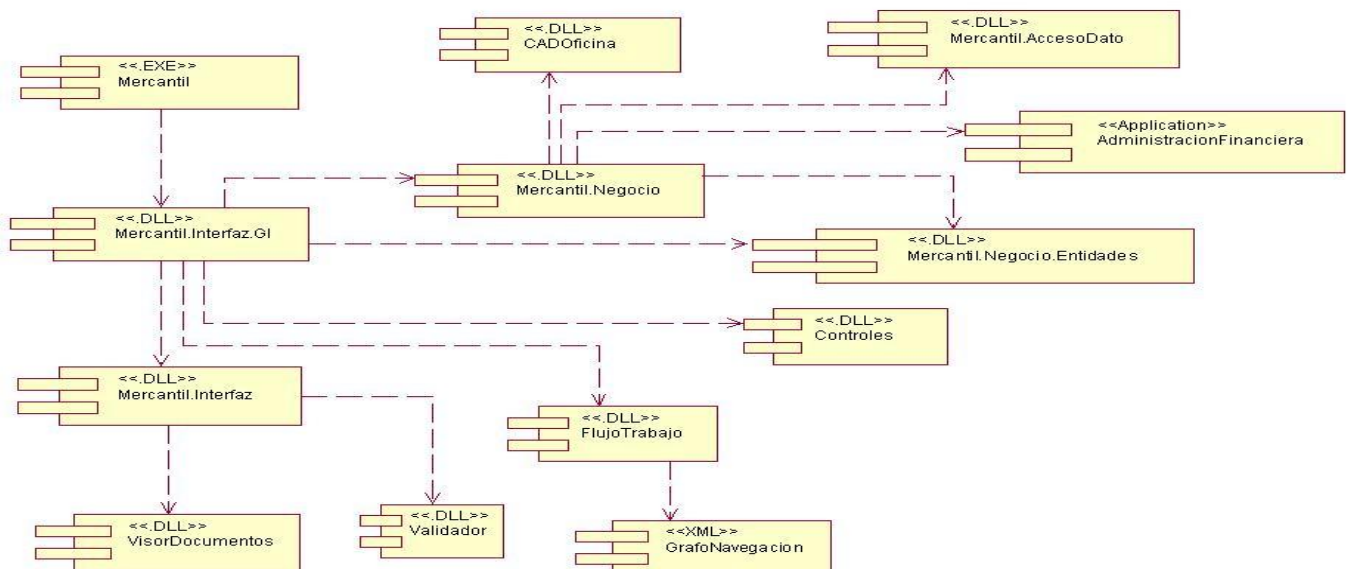


Figura 13. Diagrama de Componentes.

3.3 Diagrama de Despliegue.

Este modelo representa a través de sus nodos y relaciones entre ellos, el conjunto de dispositivos que interactuarían entre sí al desplegar el sistema propuesto, por ejemplo la comunicación entre computadoras (PC), escáner, impresoras, entre otros.

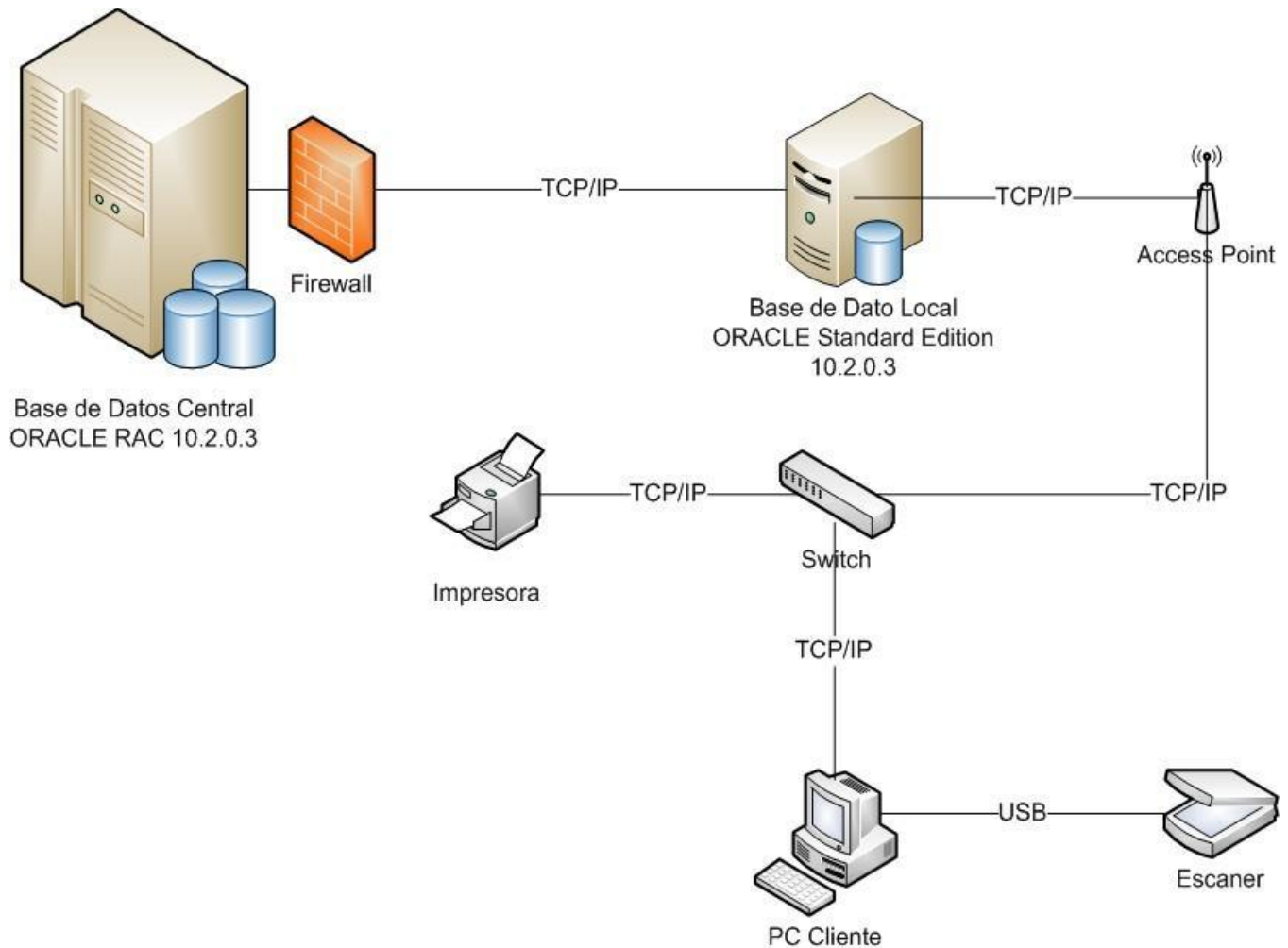


Figura 14. Diagrama de Despliegue para Registros Mercantiles.

3.4 Métricas Orientadas a Objetos Aplicadas al Diseño del Software.

El presente epígrafe hace un análisis de los resultados al aplicar las medidas para obtener la calidad del diseño del software. A nivel mundial se han definido métricas para medir las potencialidades y cualidades del diseño de un software.

Las métricas orientadas a objetos se han introducido para ayudar a un ingeniero del software a usar el análisis cuantitativo, para evaluar la calidad en el diseño antes de que un sistema se construya. El enfoque de las métricas orientadas a objetos está en las clases, piedra fundamental en la arquitectura orientada a objetos. (Pressman, 1998).

Uno de los conjuntos de métricas orientadas a objetos más ampliamente referenciados, han sido el propuesto por Chidamber y Kemerer, normalmente conocidas como la serie de métricas CK, estos autores han propuesto seis (6) métricas basadas en clases para sistemas orientados a objetos. (Pressman, 1998).

Para la validación del diseño del software se escogieron dos (2) de las series de métricas más conocidas y divulgadas a nivel mundial, la primera fue la serie de métricas CK, la cual de sus seis (6) métricas plantadas, se escogieron dos (2) en particular, por su gran importancia.

- **Árbol de Profundidad de Herencia (APH):** esta métrica se define como la máxima longitud del nodo a la raíz del árbol. (Chidamber, y otros, Junio de 1994).
- **Número de Descendientes (NDD):** las subclases inmediatamente subordinadas a una clase de la jerarquía de clases se denominan sus descendientes. (Chidamber, y otros, Junio de 1994).

La segunda serie que se escogió fue las métricas propuestas por Lorenz y Kidd, donde estos en su libro de métricas basadas en clases separan estas en cuatro (4) grandes categorías, de las cuales solo dos (2) de ellas se escogieron para la medición de este diseño.

- **Tamaño de Clases (TC):** las clases pueden medirse determinado el total de operaciones, tanto heredadas como privadas de la instancia que se encapsulan dentro de una clase, más el total de atributos, atributos tanto heredados como privados de la instancia encapsulados por la clases. (Pressman, 1998).

- **Número de Operaciones Redefinidas para una Sub-Clase (NOR):** es el caso de que una subclase reemplaza una operación heredada de su superclase por una versión especializada para su propio uso, a esto se le denomina redefinición. (Pressman, 1998).

A continuación se explicará el análisis de resultados del diseño planteado anteriormente por cada una de las series antes explicadas.

3.4.1 Árbol de Profundidad de Herencia (APH).

A medida que el APH crece, es posible que clases de más bajos niveles hereden muchos métodos, esto conlleva dificultades potenciales, cuando se intenta predecir el comportamiento de una clase. Una jerarquía de clases profunda, APH largo, también conduce a una complejidad del diseño mayor. Por el lado positivo, los valores de APH grandes implican un gran número de métodos que se reutilizarán. (Pressman, 1998).

Por su parte, algunos autores sugieren que un umbral de 6 niveles como indicador es un abuso en la herencia en distintos lenguajes de programación.

Resultado.

A partir de los datos obtenidos después de aplicar la métrica APH se obtuvo que el nivel más alto de herencia entre las clases del diseño es de 3, lo cual se encuentra dentro del umbral definido para determinar que el diseño no es complejo, existe un bajo acoplamiento entre las clases y no es difícil su mantenimiento.

3.4.2 Número de Descendientes (NDD).

A medida que el número de descendientes crece la reutilización se incrementa, pero además cuando el NDD crece, la abstracción representada por la clase predecesora puede diluirse. Esto significa que existe la posibilidad de que algunos descendientes no sean miembros realmente apropiados de la clase predecesora. A medida que el NDD crece, la cantidad de pruebas se incrementará también. (Pressman, 1998).

Al aplicar esta métrica al diseño propuesto se obtuvo el siguiente resultado:

Resultado

Después de haber aplicado la métrica de NDD, se llegó a que el sistema presenta como máximo nivel de descendientes 5, este nivel representa una media para los umbrales que proponen algunos autores en el campo de las métricas de diseño.

3.4.3 Tamaño de Clases (TC).

Según (Pressman, 1998), los valores grandes para esta métrica, indican que la clase debe tener bastante responsabilidad. Esto reducirá la reutilización de esta clase y complicará la implementación y las pruebas.

Se pueden calcular los promedios para el número de atributos y operaciones de clase. Cuando menor sea el valor del promedio para el tamaño será más posible que las clases dentro del sistema puedan ser reutilizadas.

Las medidas o umbrales para los parámetros de calidad han sido una polémica a nivel mundial en el diseño de sistemas. Algunos especialistas plantean umbrales para estas métricas según como se muestra a continuación, los cuales fueron aplicados al diseño propuesto anteriormente.

Número de Operaciones y/o Atributos	
Métrica TC	Umbral
Pequeño	≤ 20
Medio	> 20 y ≤ 30
Grande	> 30

Tabla 1. Umbrales para la Métrica TC.

La cantidad de atributos y operaciones que presenta los procesos de gestión de expedientes, gestión de sellado de libros y gestión de copia de documentos pueden verse en el [Anexo 7](#), donde en está fue aplicada la métrica TC.

Resultado.

Después de haber aplicado la métrica TC, se llegó a la conclusión que los procesos antes mencionados cuentan con 50 clases como se muestra en el [Anexo 7](#), para un promedio de cantidad de atributos de 3.62 y un promedio de la cantidad de operaciones de 7.02.

Dichos procesos presentan 1 clase de tamaño grande, 7 clases de tamaño mediano y 42 de tamaño pequeño, como se muestra en la tabla 2.

Umbral	Tamaño	Cantidad de Clases
≤ 20	Pequeño	42
> 20 y ≤ 30	Medio	7
> 30	Grande	1

Tabla 2. Cantidad de clases por tamaño

Otra forma representativa es el porcentaje de clases según los tamaño a partir de los umbrales definidos en la tabla 3, que se muestra a continuación, donde se puede observar que existe un 2% de clases grandes, un 14% de clases de tamaño mediano y un 84% de clases de tamaño pequeño.

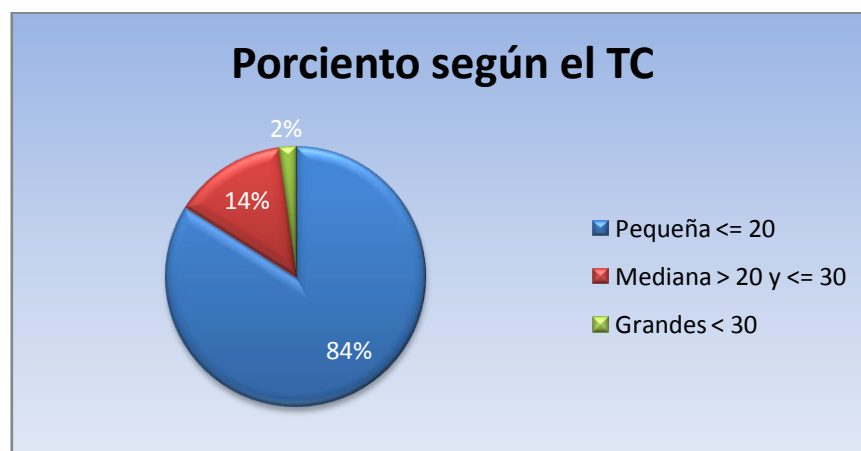


Tabla 3. Porcentos de clases por tamaños

Como se puede observar en las tablas que muestran los resultados, la mayor cantidad de clases están clasificadas entre pequeñas y medianas, y la minoría son clases de tipo grande, lo cual brinda un resultado positivo según los parámetros de calidad propuesto para la métrica tratada en este sub-epígrafe.

3.4.4 Número de Operaciones Redefinidas para una Sub-Clase (NOR).

Los valores grandes para el NOR, generalmente indica un problema en el diseño, o sea si el NOR es grande el diseñador ha violado la abstracción representada por la superclase. Esto provoca una débil jerarquía de clases y un software orientado a objetos, que puede ser difícil de probar y modificar. (Pressman, 1998).

Esta es una de las métricas que se aplican para medir la calidad del diseño propuesto para los Registros y Notarías de Venezuela.

Resultado.

A partir de los datos obtenidos después de aplicarle al sistema la métrica NOR se obtuvieron que de 64 clases que tiene el sistema solo 15 subclases reemplazan operaciones de las superclases como se muestra en la tabla.

Cantidad de clases del sistema	Total de subclases que reemplazan operaciones
64	15

Tabla 4. Total de clases que reemplazan operaciones.

Otra forma que se puede representar esta métrica es según el porcentaje que representa las subclases que reemplazan operaciones de su superclases, como se muestra en la tabla 5, que se muestra a continuación, donde se puede observar que el 23% de las clases reemplazan operaciones de su superclases y el 77% de las clases del sistema, no tienen que reemplazar ninguna operación.

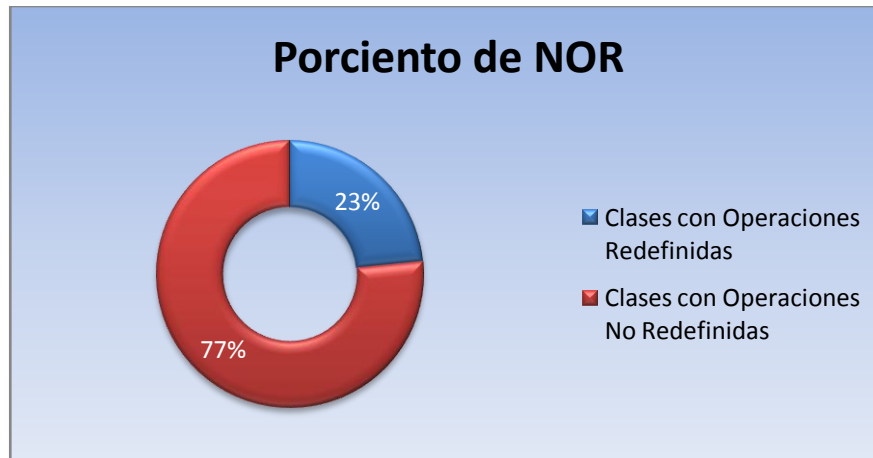


Tabla 5. Porcentaje aplicando NOR.

El porcentaje de subclases que reemplazan operaciones de las superclase es mínimo en relación al porcentaje de clases que no reemplazan operaciones en el sistema. Se puede notar que existe una jerarquía adecuada, y también esto permite que el software pueda ser fácil probado y modificado sin afectar tanto el tiempo que requieran los cambios.

3.5 Resultados Obtenidos de las Pruebas de Caja Blanca.

Las pruebas de caja blanca o pruebas de caja de cristal, es un método de diseño de casos de prueba que se usa la estructura de control del diseño procedimental para obtener los casos de pruebas. Según (Pressman, 1998), mediante los métodos de pruebas de caja blanca, los ingenieros del software pueden obtener casos de pruebas que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Para la evaluación de la calidad de la implementación del módulo de Mercantil, se realizaron dichas pruebas a algunos escenarios fundamentales de los procesos explicados en el capítulo 2, así como de

algunos controles usados en el desarrollo del software, para ello se tomó como herramienta de validación el NUnit Framework.

3.5.1 NUnit Framework.

Según (Guerrero Carreón), el NUnit es un framework de software libre que sirve para realizar pruebas unitarias para .NET; donde las pruebas unitarias es un procedimiento usado para validar que un módulo de código fuente funciona apropiadamente.

El NUnit Framework según (Guerrero Carreón), trae como beneficio principal que permite aislar segmentos del programa que pueden ser probados para verificar que funcionan correctamente.

3.5.2 Resultados obtenidos de las pruebas aplicadas a los controles.

Este es el método donde se declara el punto inicial de las pruebas de unidad, en el se declaran las inicializaciones para las futuras pruebas de los siguientes operaciones.

```
[SetUp]
public void Init()
{
    coleccionOriginal=new ColeccionDocumentosCopia();

    libros=new ColeccionLibroSellar();
    LibroSellar libro=new LibroSellar();
    libro.CantFolios = 500;
    libro.Descripcion = "Actas";
    libro.IdLibro = 1;
    libro.Repetible = true;
    libro.TipoLibro = "Libro de Actas";
    libros.Add(libro);
}
```

Figura 15. Método de inicio de las pruebas.

Este método valida a todos y cada uno de los controles que contiene el formulario en el que él se encuentra, se va desplazando por cada uno de ellos y si encuentra alguno cuyos datos no cumplan con los requerimientos establecidos devuelve el resultado en forma de un valor booleano.

```

[Test]
public void ValidarTodos()
{
    Assert.AreEqual(true,validador1.ValidateAll(),"Valor esperado");
}

```

Figura 16. Prueba al método de validación de todos los controles.

Este método se encarga de dado un control que recibe por parámetros, validar su contenido y construir un mensaje de error en caso de que no cumpla con los requerimientos de validación establecidos.

```

[Test]
public void ObtenerMensajeError()
{
    Assert.AreEqual(string.Empty,validador1.GetErrorMessage(txtCantCopias),"Valor esperado");
}

```

Figure 17. Prueba al método de validación de un control.

3.5.3 Resultados obtenidos de las pruebas aplicadas a las controladoras de interfaz.

Este método se encarga de validar, a la hora de conformar las combinaciones de las copias certificadas, que no se quede ningún documento sin adicionar a una combinación, teniendo en cuenta la cantidad de copias que se le asignó a cada documento.

```

[Test]
public void FaltaAlgunDoc()
{
    Assert.AreEqual(false,FaltaAlgunDocumento(),"Valor esperado");
}

```

Figura 18. Pruebas al método de conformación de combinaciones en las Copias Certificadas.

Este método recibe por parámetros una lista vacía, la cual se encargará de llenar con las descripciones de los errores que se encuentre al efectuar la validación de todo el formulario. En el caso de que el formulario se valide correctamente retornará verdadero, en caso contrario falso y la lista quedará vacía.

```
[Test]
public void ErroresDetectadosPrueba()
{
    StringCollection errores=new StringCollection();
    Assert.AreEqual(true,ErroresDetectados(errores),"Valor esperado");
}
```

Figura 19. Pruebas al método de validación de errores de un formulario.

3.5.4 Resultados obtenidos de las pruebas aplicadas a la Fachada en la capa de negocio.

Este método recibe por parámetros el identificador del expediente, el identificador del usuario que está autenticado en la aplicación y un parámetro de salida que es el estado en que se encuentra la solicitud. Con esto se encarga de verificar si ese usuario tiene en su poder el expediente en cuestión y de ser así guarda el estado de la solicitud en el parámetro de salida.

```
[Test]
public void UsuarioTieneExp()
{
    decimal estadoSol;
    Assert.AreEqual(false,UsuarioTieneExpediente(1,1,out estadoSol),"Valor esperado");
}
```

Figura 20. Prueba al método de usuario conectado por expediente.

Este método recibe por parámetros el identificador del expediente y se encarga de buscar los datos de la ubicación del mismo dentro del Archivo.


```

[Test]
public void CargarDatosExpediente()
{
    Assert.AreEqual(true, BuscarDatosExpediente(1), "Valor esperado");
}

```

Figura 21. Prueba al método cargar expediente del archivo.

Este método se encarga de extraer de la base de datos todos los nomencladores de los libros que se pueden sellar, de ellos carga el identificador, descripción del tipo de libro y si el tipo de libro se puede repetir.

```

[Test]
public void ObtenerNomencladoresLibros()
{
    Assert.AreEqual(true, ObtenerNomencladorTipoLibro(), "Valor esperado");
}

```

Figura 22. Prueba al método de obtener nomencladores de libros a sellar.

Este método dado un identificador del trámite que recibe como parámetro, devuelve una colección de los libros que tiene para sellar la solicitud contenida en el trámite, de estos devuelve su identificador, descripción, descripción del tipo de libro, si el tipo de libro se puede repetir y la cantidad de folios de cada libro.

```

[Test]
public void ObtenerLibrosPorTramite()
{
    Assert.AreEqual(true, ObtenerLibrosTramite(1100), "Valor esperado");
}

```

Figura 23. Prueba al método obtener libros a sellar por trámites.

Como resultado de las pruebas realizadas a los controles, controladoras de interfaz y a las fachadas del negocio especificadas, a continuación se puede observar la interfaz que brinda el NUnit como resumen de las pruebas realizadas a dichas operaciones, donde se muestra que las pruebas fueron realizadas satisfactoriamente.

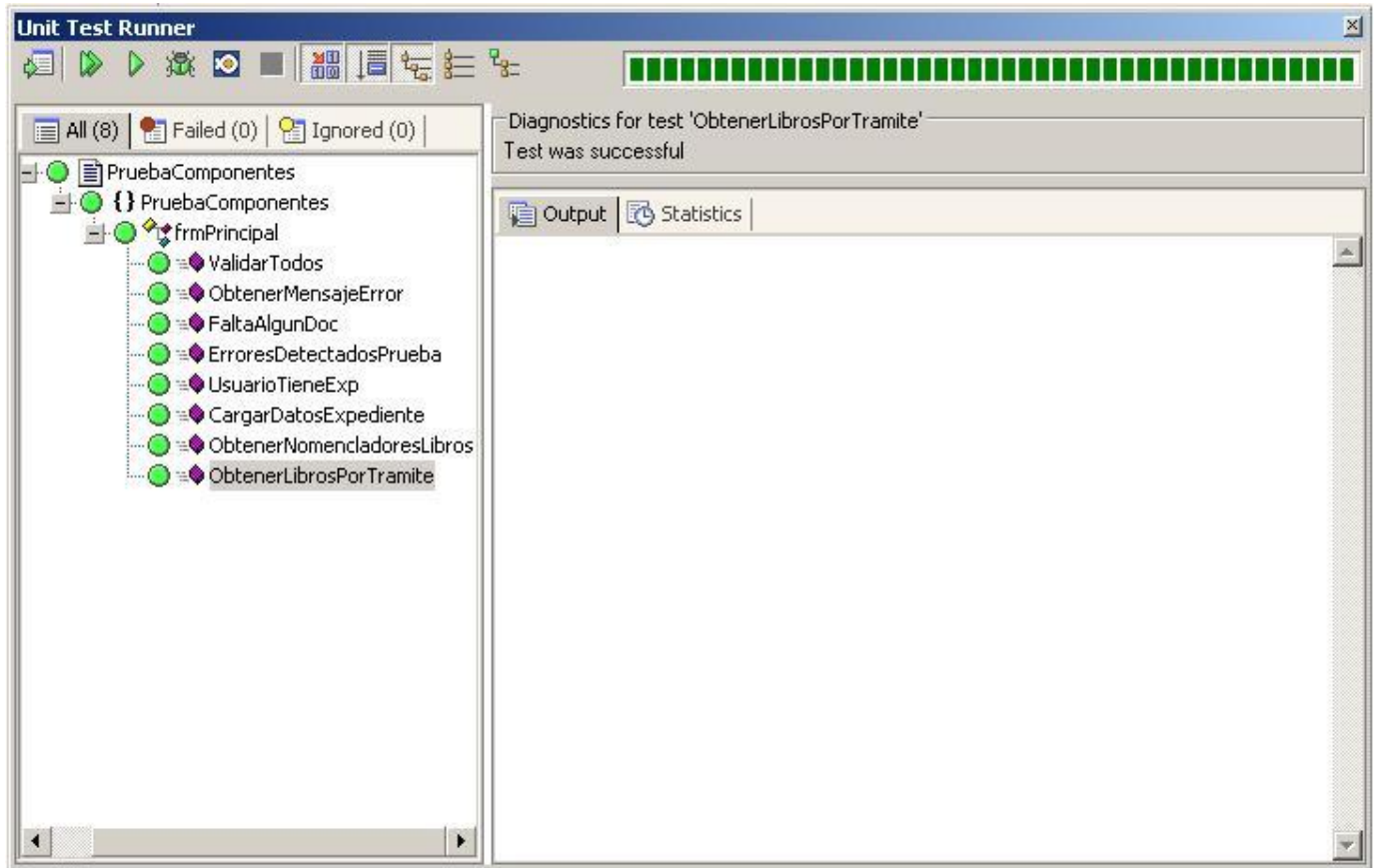


Figura 24. Pantalla de resumen del NUnit Framework.

3.6 Conclusiones.

En este capítulo se hizo se definió una solución de diseño, que responde completamente a la arquitectura planteada en el capítulo anterior, la cual permite que el sistema sea auditable y controlable para las situaciones que se plantean.

Se exponen los distintos tipos de clases que van a representar al diseño los cuales están formados por distintos paquetes, según como se explica en epígrafes anteriores, también se realizó el diseño de los diagramas de clases y diagramas de secuencia por cada uno de los procesos que competen al software que se presenta.

En este capítulo también se logró aplicar una serie de métricas propuestas para la validación del diseño propuesto, en el cual se determinó que este no presenta una alta complejidad estructural, de datos, ni del sistema en general, permitiendo que las pruebas no sean complejas y afecten en mínimo de tiempo algún cambio.

La métrica de Tamaño de Clases evidencia que la mayoría de las clases son reutilizables, la implementación no es complicada y las pruebas no son complejas, al igual que la profundidad de los niveles de herencia están acorde con los umbrales definidos por autores consultados permitiendo este un bajo acoplamiento y que el sistema no sea complejo.

Por último se logró aplicar las pruebas de unidad con el NUnit Framework en las cuales se demostró haber implementado los requisitos primarios, donde los resultados fueron satisfactorios, llevando esto a que el producto responda a todas las necesidades y propósitos de los clientes.

CONCLUSIONES GENERALES.

Como conclusiones generales de la investigación presente se tienen las siguientes:

- Quedó definida la metodología, tecnología y las herramientas, para la construcción del software a partir de un estudio previo de las funcionalidades de los procesos no registrales, mencionados en la investigación.
- Se analizó la arquitectura planteada por la dirección del proyecto, donde se llegó a la conclusión de que es la más adecuada para la implementación del sistema que se está proponiendo por todas las características antes planteadas.
- Se definió una estrategia para la construcción de un diseño que fuese flexible y escalable, a partir de la implantación de patrones y técnicas de diseño.
- Con el modelo de diseño de los procesos no registrales, se obtuvieron los artefactos necesarios, según la metodología de desarrollo de software definida (RUP), para la implementación, flujo que define esta metodología a continuación del diseño.
- Los resultados obtenidos a partir de los análisis del diseño y la implementación del sistema son positivos, tomando como argumento la aplicación de métricas y pruebas de caja blanca que permitieron evaluar el nivel de calidad que presenta el software.
- El sistema planteado facilita la organización y control de todos los procesos no registrales planteados anteriormente.
- El sistema se apoya totalmente en las Leyes Registrales para su correcto funcionamiento, evitando así que se pueda cometer errores, dando esto una gran confianza al ciudadano venezolano cuando vaya a realizar algún tipo de trámite en cualquier Registro Mercantil.
- El sistema brinda toda una facilidad y comodidad para los funcionarios de archivo en el proceso de solicitudes de expedientes, debido a todas las búsquedas y controles de dicho documento dentro del departamento.
- Todas las operaciones planteadas en un expediente de una compañía son accesibles y conocidas desde cada Registro, brindando esto una gran publicidad registral en todo momento.

RECOMENDACIONES.

- Migrar hacia otra herramienta CASE, la cual tenga incorporado todos los elementos necesarios para soportar la metodología escogida, e incorpore la potencialidad de soportar UML 2.1 u otro lenguaje de modelado más avanzado.
- Hacer un estudio y un análisis de alguna plataforma o lenguaje en software libre que soporte la migración del software, sin que tenga que emplearse un gran tiempo de ejecución para esta migración.
- Desarrollar los diagramas de diseño e implementación de los otros procesos que complementan al proceso de Inscripción que se ejecutan en los Registros Mercantiles.
- Documentar el Framework utilizado en la Capa de Presentación para que sea posible su utilización por otros desarrolladores y proyectos.

BIBLIOGRAFÍA.

ALBET, Ingeniería en Sistemas. Octubre 10, 2006. *Estándares de codificación.* Ciudad Habana, Cuba : s.n., Octubre 10, 2006.

Almenares, Kiosmy y de León, Rubén. 2007. *Diseño e implementación del proceso de inscripción del Módulo de Mercantil en las Oficinas Registrales de la República Bolivariana de Venezuela.* Caracas, Venezuela : s.n., 2007.

Artículo Nro 4, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 5, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 51, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DEL REGISTRO PÚBLICO Y NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 6, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 62, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 7, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 8, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

Artículo Nro 9, ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. *NUEVA LEY DE REGISTRO PÚBLICO Y DEL NOTARIADO DE VENEZUELA.* Caracas, Venezuela : s.n., 15 de Enero de 2007.

ASAMBLEA NACIONAL DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. 15 de Enero de 2007. NUEVA LEY DEL REGISTRO PÚBLICO Y NOTARIADO DE VENEZUELA. Caracas, Venezuela : s.n., 15 de Enero de 2007.

— **Noviembre 27, 2001.** Gaceta Oficial N° 37333. *Decreto con Fuerza de Ley de Registro y Notario.* Caracas, Venezuela : s.n., Noviembre 27, 2001.

Astarita, Emilio; Creative Commons. *Manual XHTML.*

Bass, Len, Clements, Paul y Kazman, Rick. 1998. *Software Architecture in Practice.* 1998.

Baufest, proyecto Exactas. 2006. *Resumen del seminario corto acerca de la metodología SCRUM.* Buenos Aires, Argentina : s.n., 2006.

Bipin, Joshi. Dot Net Bips. [En línea]
<http://www.dotnetbips.com/articles/61bcc693-292f-47a6-9729-75469c99ca23.aspx>.

Cano Ossa, Mauricio. Agosto 12, 2006. *EL DIOS DE LOS MONOS. Una guía para el desarrollador de GUIs en windows con el mono.* Medellín : s.n., Agosto 12, 2006.

Canós José A.; Letelier, Patricio; Panadés, María Carmen; DSIC -Universidad Politécnica de Valencia. *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.

Carballeira, Dr. Félix García. 2000. Revista Digital Universitaria. [En línea] 1 de Octubre de 2000.
<http://www.revista.unam.mx/vol.1/num2/art4/>.

Chidamber, S. R. y Kemerer, C. F. Junio de 1994. *A Metrics Suite for Object-Oriented Design.* s.l. : IEEE Trans. Software Engineering, Junio de 1994.

Clements, Paul. 1996. *A Survey of Architecture Description Languages.* Alemania : s.n., 1996.

CONGRESO DE LA REPÚBLICA DE VENEZUELA. Diciembre 21, 1955. *CÓDIGO DE COMERCIO, Gaceta extraordinaria No. 475.* Caracas, Venezuela : s.n., Diciembre 21, 1955.

Dass, Karen. Model-View-Controller (MVC) Architecture. *india Web Developers.* [En línea]
<http://www.indiawebdevelopers.com/technology/java/mvcarchitecture.asp>.

de Fuenmayor, Lola. 1995. *OPUS, Enciclopedia Jurídica. Volumen VII.* Caracas, Venezuela : s.n., 1995.

Gamma, E., y otros. 1995. *Design Patterns.* s.l. : Addison-Wesley, 1995.

Garlan, David y Shaw, Mary. 1994. *An introduction to software architecture.* Pittsburgh : CMU Software Engineering Institute Technical Report, 1994.

- Guerrero Carreón, José Raúl.** *Pruebas unitarias utilizando NUnit.* s.l. : Reunión de la Comunidad .NET D.F.
- Harvard Business Review.** Enero - Febrero, 1986. Enero - Febrero, 1986.
- Hensgen, Paul; Umbrello UML Modeller Authors.** October 15, 2003. *Umbrello UML Modeller Handbook.* October 15, 2003.
- Hernández, A.** 1998. *Curso de derecho mercantil.* 4ta Edición. 1998.
- Hernández, Jordi.** 2004. *Mono: mucho más que una implementación libre de .NET.* 2004.
- Jacobson, Ivar, Rumbaugh, James y Booch, Grady.** 2000. *El proceso unificado de desarrollo de software.* s.l. : Adison Wesley, 2000.
- Larman, Craig.** 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Upper Saddle River, EUA : Prentice Hall, 1999.
- Medina Pasaje, Julio Luis.** Marzo 7, 2006. *Metodología y herramientas UML para el modelado y análisis de sistemas de tiempo real orientados a objetos.* Madrid, España : s.n., Marzo 7, 2006.
- Microsoft Corporation.** June, 2003. *Microsoft Solutions Framework version 3.0 Overview. White paper.* June, 2003.
- Molpeceres, Alberto.** Diciembre 15, 2002. *Procesos de desarrollo: RUP, XP, FDD.* Diciembre 15, 2002.
- MSDN.** 2008. .NET Framework Developer Center. [En línea] Microsoft, 2008.
<http://msdn.microsoft.com/netframework/>.
- Pressman, Roger S.** 1998. *Ingeniería de Software. Un enfoque práctico.* 1998.
- Ramírez, J.** 1993. *Derecho General y Mercantil.* 3ra Edición. 1993.
- Rational Software Corporation.** 2002. *Product.* 2002.
- Real Academia de la Lengua Española.**
- Reynoso, Carlos y Kiccillof, Nicolás.** 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* s.l. : Universidad de Buenos Aires, 2004.
- Robles, Gregorio; Grupo de Sistemas y Comunicaciones.** Abril 4, 2003. *Programación eXtrema (y software libre).* Madrid : s.n., Abril 4, 2003.
- Seoane, Fabian.** 2004. LUGCIX. [En línea] Linux User Group Chiclayo, Diciembre de 2004.
<http://www.lugcix.org/tutoriales/mono/queesmono.php>.

Shaw, Mary y Clements, Paul. Abril de 1996. *A field guide to Boxology: Preliminary classification of architectural styles for software systems.* Pennsylvania, Estados Unidos de América : Computer Science Department and Software Engineering Institute, Carnegie Mellon University, Abril de 1996.

Shaw, Mary y Garlan, David. 1996. *Software Architecture: Perspectives on an emerging discipline.* Upper Saddle River : Prentice Hall, 1996.

Superintendencia de Bancos y otras instituciones financieras. 1998. *Curso Básico de Supervisión Bancario.* Caracas, Venezuela : s.n., 1998.

Trowbridge, David. 2003. *Enterprise Solution Patterns using Microsoft .NET.* s.l. : Microsoft Corporation, 2003.

Walsh, Norman, Muellner, Leonard y Stayton, Bob. Wed, 11 Oct 2006. *DocBook: The Definitive Guide.* s.l. : O'Reilly & Associates, Inc., Wed, 11 Oct 2006.

GLOSARIO DE TÉRMINOS.

Registro Mercantil: Es una institución de carácter público, autónomo sin personalidad jurídica que depende del Ministro de Interior y Justicia creada con el fin de suministrar en el interés de la colectividad y también en el del comerciante, datos importantes sobre la situación jurídica de estos, teniendo los documentos que allí se registren efectos contra terceros. Su principal función es satisfacer la obligación que tiene el estado en proteger los intereses individuales y colectivos de los ciudadanos proporcionándoles la posibilidad de que ellos por si mismos pueden obtener la información necesaria de una persona jurídica como sujeto de derecho y así tratar de evitar las competencias desleales, fraudes con respecto a terceros.

Expediente: Conjunto de todos los papeles correspondientes a un asunto o negocio. Se usa señaladamente hablando de la serie ordenada de actuaciones administrativas, y también de las judiciales en los actos de jurisdicción voluntaria.

Pieza: Constituye la unidad básica para organizar los expedientes en los Registros Mercantiles, éstos están conformados por piezas, en las que se guardan los diferentes documentos. Cada expediente tiene como mínimo una pieza y puede extenderse ese número según el tamaño del expediente y el criterio con respecto a cantidad de folios o documentos que contenga cada pieza.

Revisión Legal: Es el paso donde tiene mayor peso la verificación de los principales requisitos de forma y fondo, siendo este paso el que más aporta al principio de legalidad.

Principio de rogación: La presentación de un documento dará por iniciado el procedimiento registral, el cual deberá ser impulsado de oficio hasta su conclusión, siempre que haya sido debidamente admitido.

Principio de prioridad: Todo documento que ingrese al Registro deberá inscribirse u otorgarse con prelación a cualquier otro título presentado posteriormente, salvo las excepciones establecidas por la Ley.

Principio de especialidad: Los bienes y derechos inscritos en el Registro, deberán estar definidos y precisados respecto a su titularidad, naturaleza, contenido y limitaciones.

Principio de consecutividad: De los asientos existentes en el Registro, relativos a un mismo bien, deberá resultar una perfecta secuencia y encadenamiento de las titularidades del dominio y de los demás

derechos registrados, así como la correlación entre las inscripciones y sus modificaciones, cancelaciones y extinciones.

Principio de legalidad: Solo se inscribirán en el Registro los títulos que reúnan los requisitos de fondo y forma establecidos por la Ley.

Principio de publicidad: La fe pública registral protege la verosimilitud y certeza jurídica que muestran sus asientos. La información contenida en los asientos de los Registros es pública y puede ser consultada por cualquier persona.

Publicidad Registral: Es un medio a través del cual se puede obtener información sobre los bienes muebles o inmuebles o sobre las personas en general. Es el acceso a la información registral. Se clasifica en: publicidad registral notificativa, publicidad registral declarativa y publicidad registral constitutiva.

Application Programming Interface: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

GNU/Linux: GNU/Linux es, un Sistema Operativo. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium, Pentium Pro y Pentium II y superior, así como los clones AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0.

SmallTalk: Programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

Framework: Denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entornos de ejecución distribuido.

UML: Lenguaje Unificado de Modelado, es un lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

ANEXOS.

Anexo 1. Diagramas de Clases de Gestión de Expedientes.

En este anexo se presentan algunos de los diagramas de clases más importantes en la Gestión de Expedientes, ya sean las solicitudes internas o formales. En estos diagramas quedan reflejadas todas las clases que intervienen en dichos procesos. También se verán reflejados en el diagrama los patrones antes mencionados, utilizados en los casos de uso asociados al expediente porque se tiene que llevar un estricto control de todos los estados por los que puede pasar éste, debido a que si un expediente no se encuentra en Archivo pues eso impedirá que se pueda realizar cualquier otra operación en toda la aplicación.

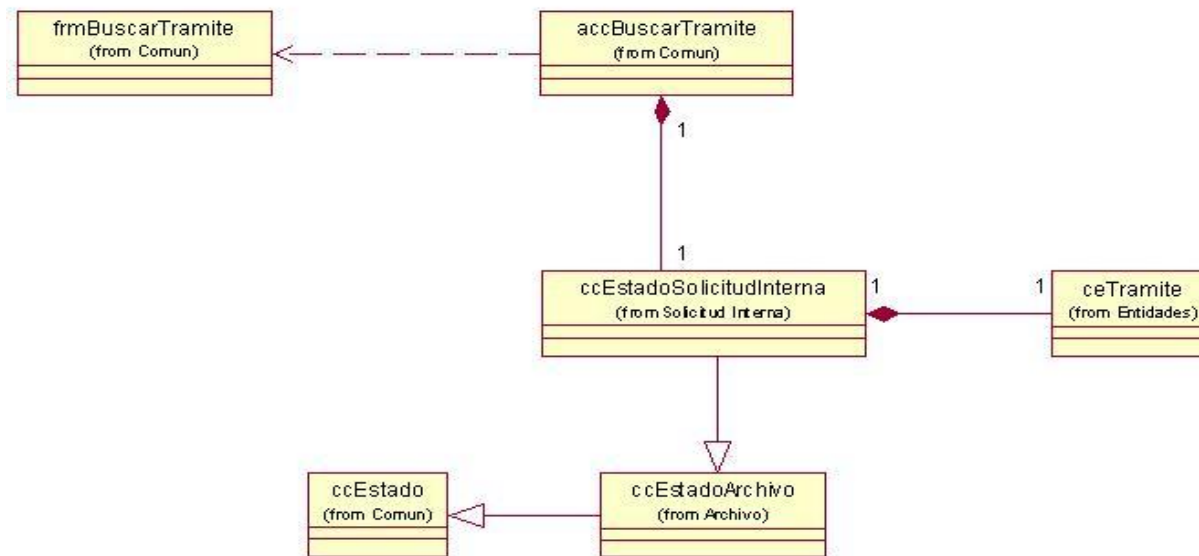


Figura 25. Diagrama de Clases del CU: Aceptar Solicitud Interna de Expedientes.

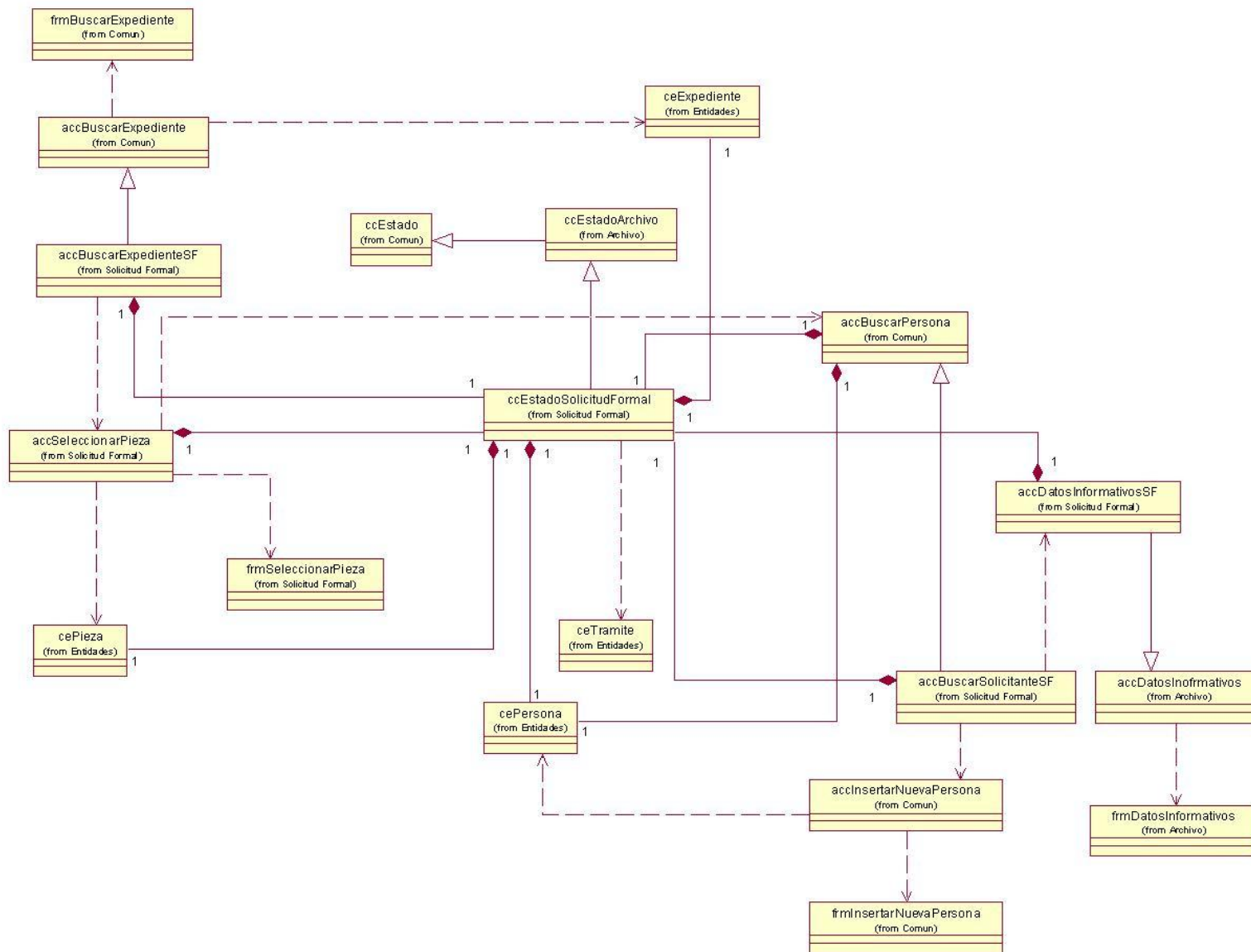


Figura 26. Diagrama de Clases del CU: Solicitud Formal de Expediente.

Para más información sobre los diagramas de clases de estos procesos, ver el archivo del Rational Rose “Procesos que complementan al proceso de Inscripción”, adjuntado al documento.

Anexo 2. Diagramas de Clases de Gestión de Sellado de Libros.

En este anexo se presentan algunos de los diagramas de clases más importantes en la Gestión de Sellado de Libros. En estos diagramas quedan reflejadas todas las clases que intervienen en dichos procesos.

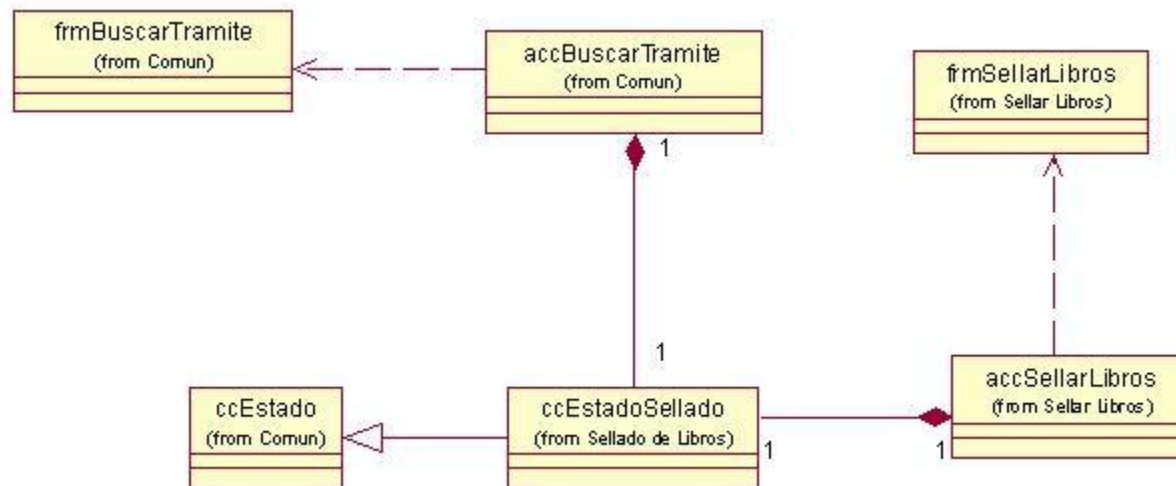


Figura 27. Diagrama de Clases del CU: Sellar Libros.

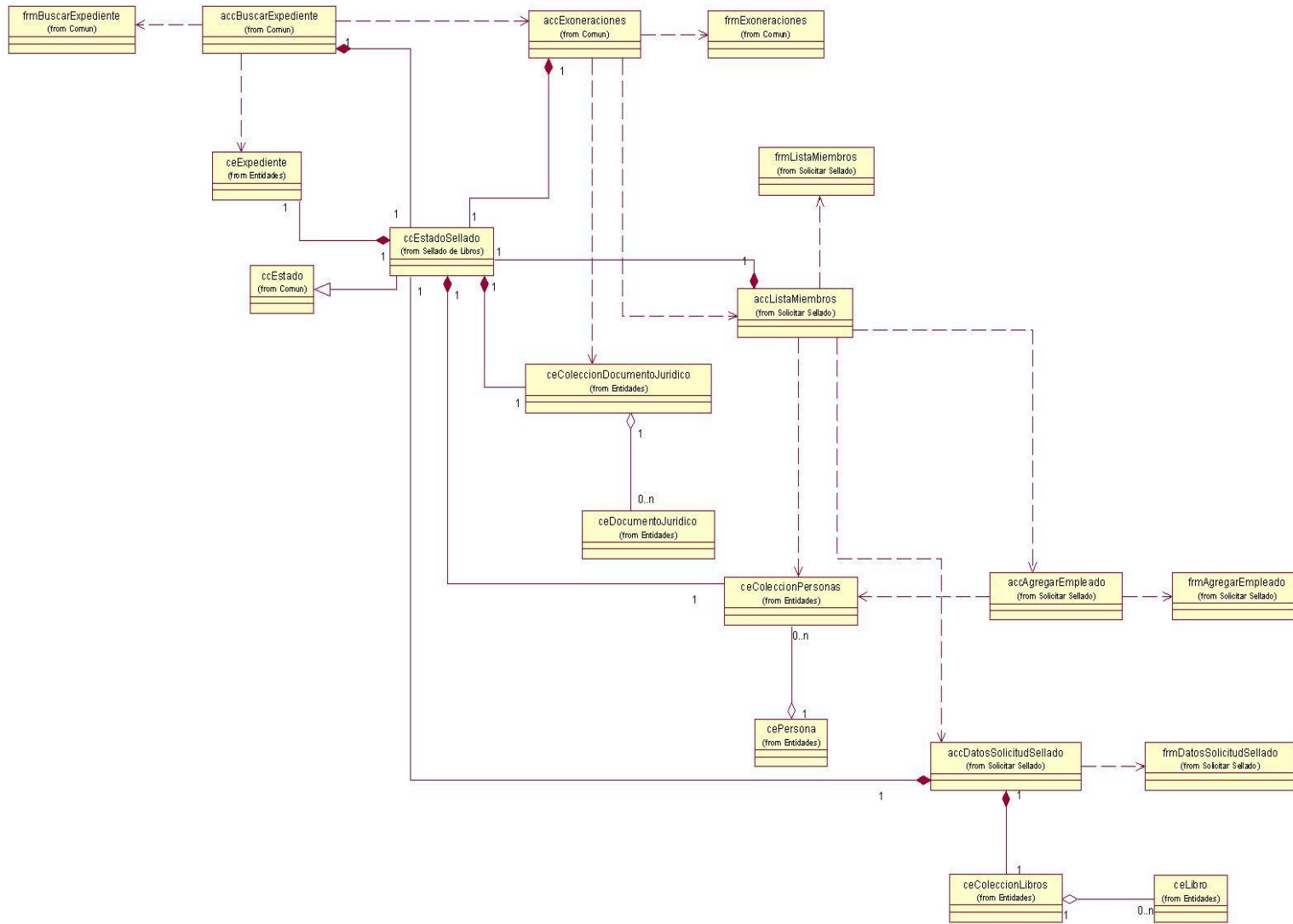


Figura 28. Diagrama de Clases del CU: Solicitar Sellado de Libros.

Para más información sobre los diagramas de clases de este proceso, ver el archivo del Rational Rose “Procesos que complementan al proceso de Inscripción”, adjuntado al documento.

Anexo 3. Diagramas de Clases de Gestión de Copia de Documentos.

En este anexo se presentan algunos de los diagramas de clases más importantes en la Gestión de Copia de Documentos. En estos diagramas quedan reflejadas todas las clases que intervienen en dichos procesos.

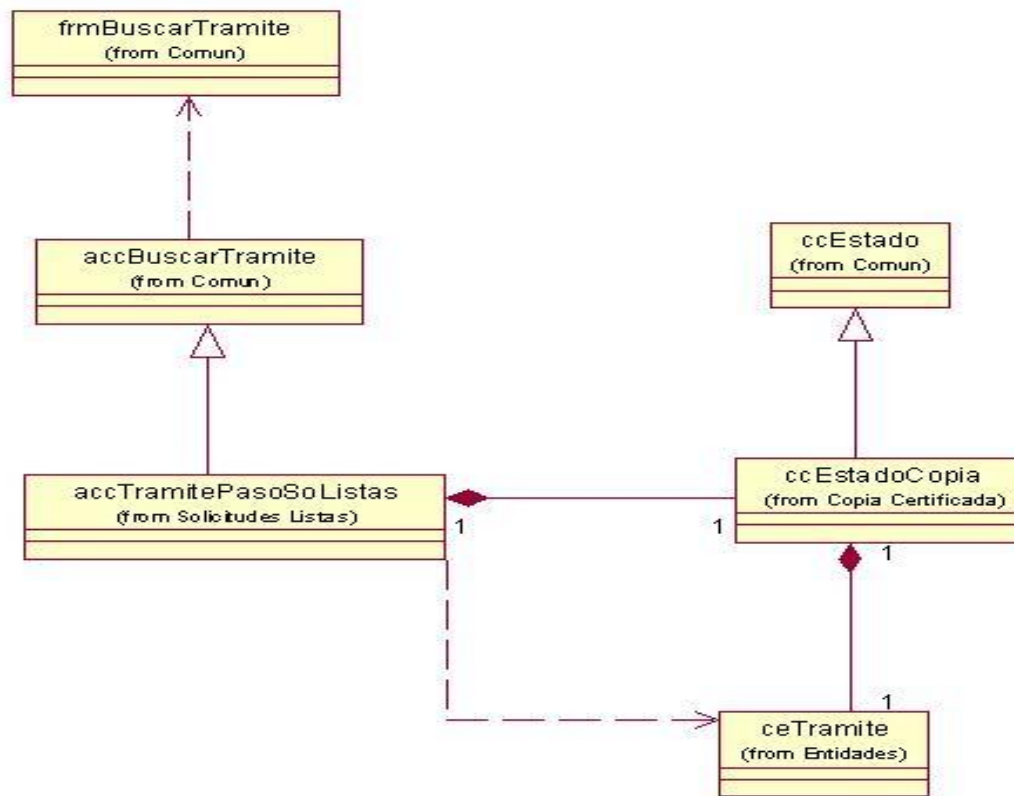


Figura 29. Diagrama de Clases del CU: Solicitudes de Copia de Documentos Listas.

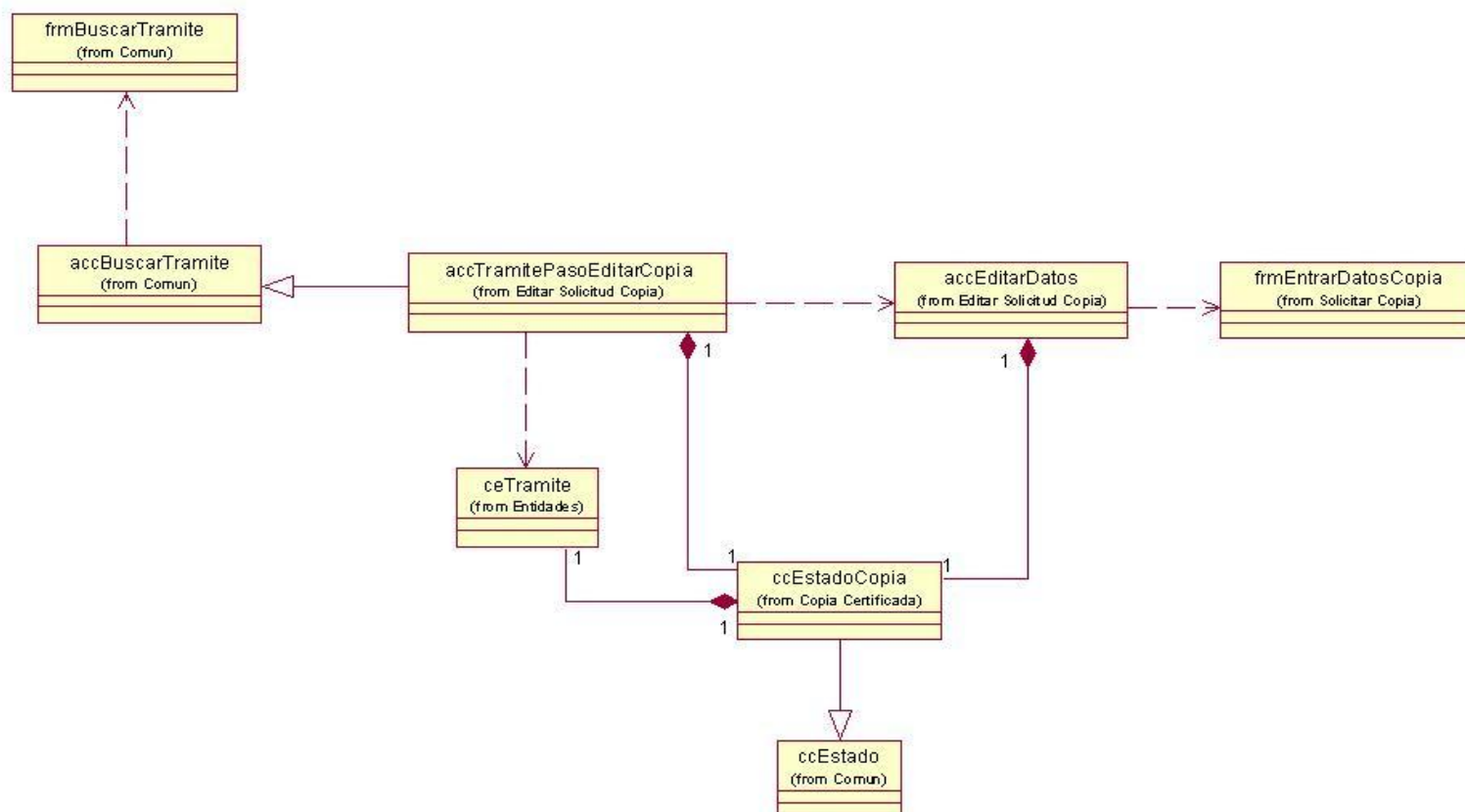


Figura 30. Diagrama de Clases del CU: Editar Solicitudes de Copia de Documentos.

Para más información sobre los diagramas de clases de este proceso, ver el archivo del Rational Rose “Procesos que complementan al proceso de Inscripción”, adjuntado al documento.

Anexo 4. Diagramas de Secuencias de Gestión de Expedientes.

En este anexo se presentan algunos de los diagrama de secuencia más relevantes para la Gestión de Expedientes en los Registros Mercantiles. En dichos diagramas quedarán reflejadas todas las operaciones que intervienen en dichos procesos, relacionándose y gestionando todo el negocio presentado.

El proceso de Devoluciones de Expedientes es uno de los procesos más importantes que existen en la Gestión de Expedientes, ya que es cuando el solicitante, tanto externo como interno del Registro Mercantil, debe entregar dicho documento y este tendría que informar a todos los procesos que quieren usar este expediente que ya se pueden hacer operaciones sobre el mismo.

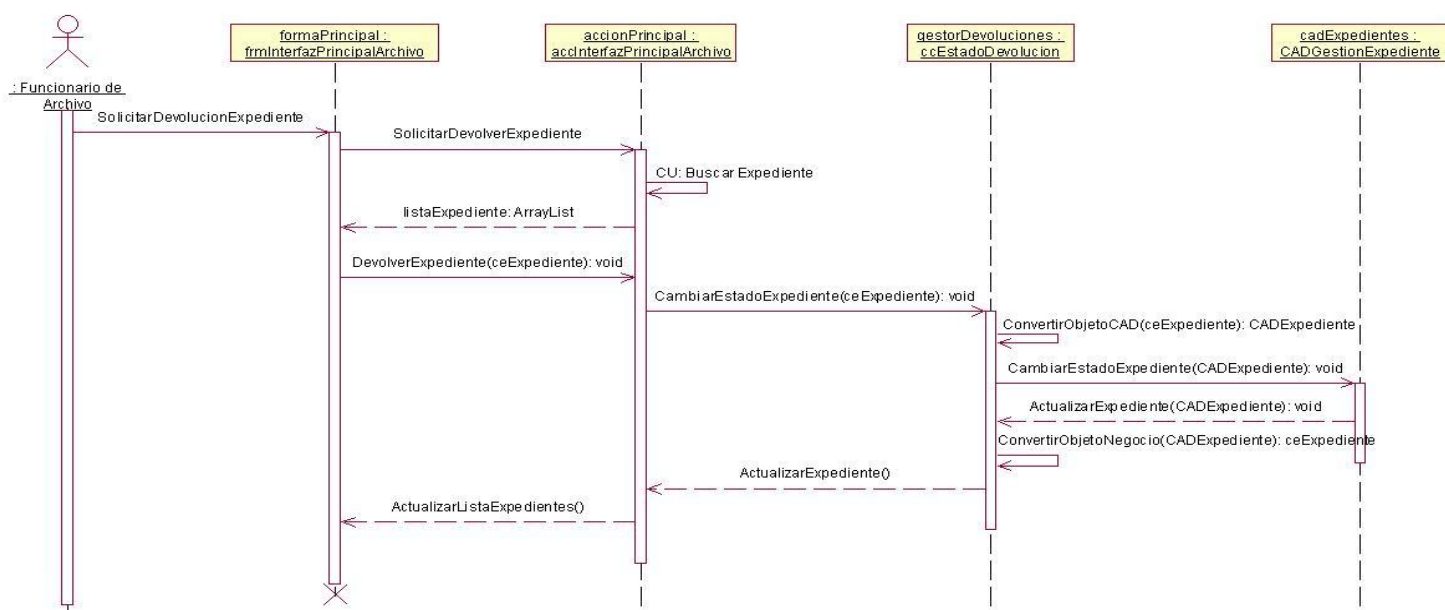


Figura 31. Diagrama de Secuencia del CU: Devolver Expedientes Solicitados Formales.

Como se muestra en la figura anterior este proceso llama a un caso de uso auxiliar, Buscar Expedientes, a continuación se muestra el diagrama de secuencia de cómo es que se obtienen todos los expedientes según una serie de parámetros que se pasan, para su búsqueda.

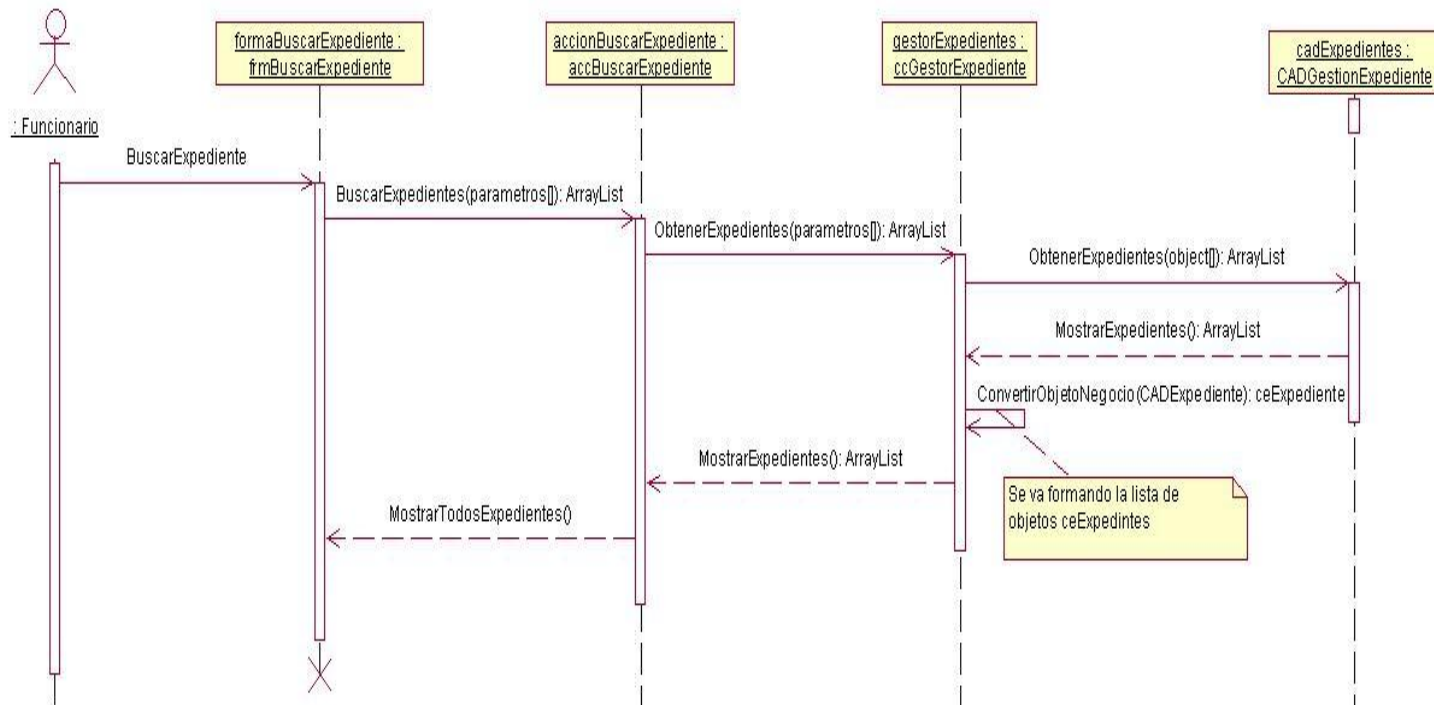


Figura 32. Diagrama de Secuencia del CU: Buscar Expedientes.

Para más información sobre los diagramas de secuencia de este proceso, ver el archivo del Rational Rose "Procesos que complementan al proceso de Inscripción", adjuntado al documento.

Anexo 5. Diagramas de Secuencias de Gestión de Sellado de Libros.

En este anexo se presentan algunos de los diagramas de secuencia más relevantes para la Gestión de Sellado de Libros de Comerciantes en los Registros Mercantiles. En dichos diagramas quedan reflejadas todas las operaciones que intervienen en dichos procesos, relacionándose y gestionando todo el negocio presentado.

El proceso de Sellar los Libros, es uno de los procesos más importantes que existen en la Gestión de Sellado de Libros, ya que va a realizar todo el proceso legal, al igual que la realización de las actualizaciones constantes de los libros de sellado, por el paso en que se encuentra su procesamiento.

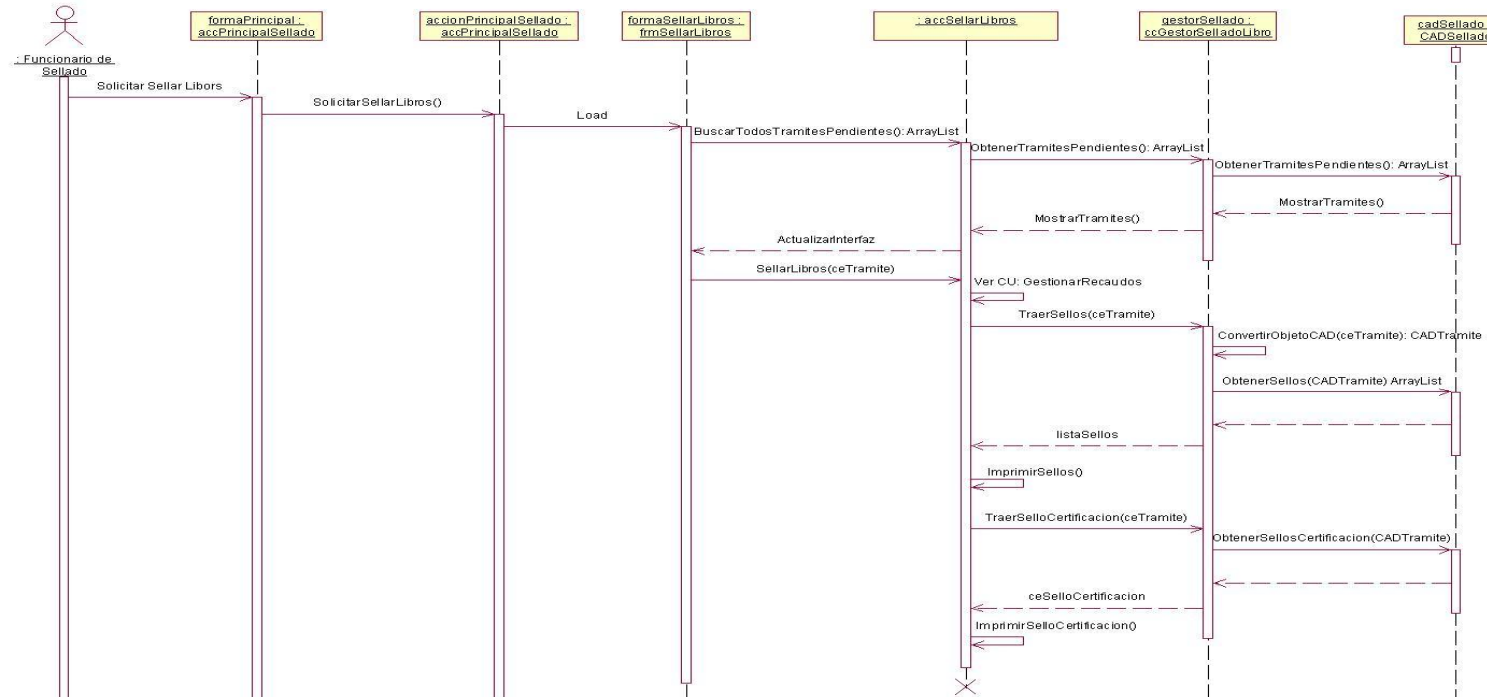


Figura 33. Diagrama de Secuencia del CU: Sellar Libros.

Como se muestra en la figura anterior este proceso llama a un caso de uso auxiliar, Gestionar Recaudos, a continuación se muestra el diagrama de secuencia de cómo es que se gestionan todos los recaudos asociados al trámite.

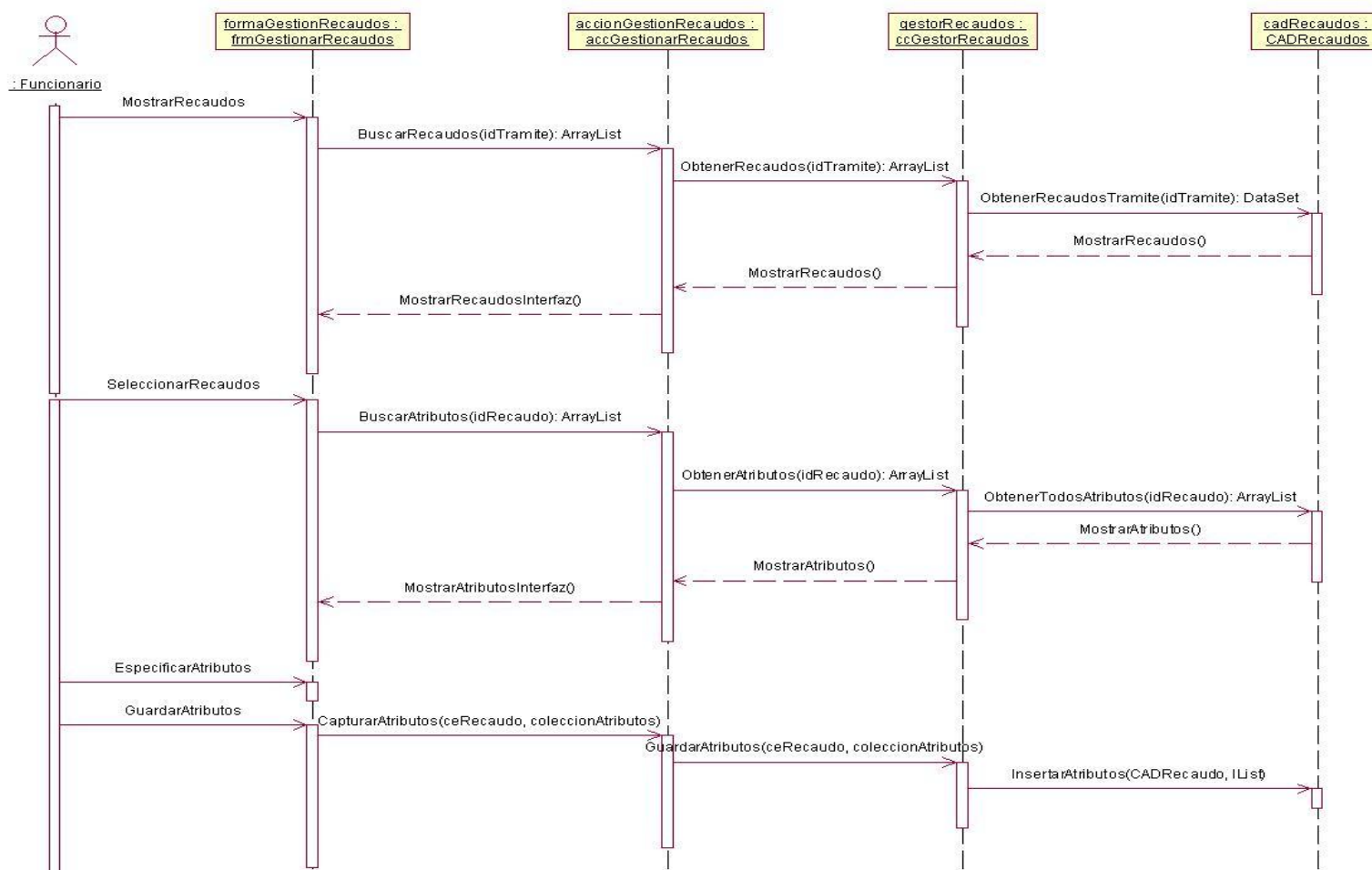


Figura 34. Diagrama de Secuencia del CU: Gestionar Recaudos.

Para más información sobre los diagramas de secuencia de este proceso, ver el archivo del Rational Rose “Procesos que complementan al proceso de Inscripción”, adjuntado al documento.

Anexo 6. Diagramas de Secuencias de Gestión de Copia de Documentos.

En este anexo se presentan algunos de los diagramas de secuencia más relevantes para la Gestión de Copia de Documentos en los Registros Mercantiles. En dichos diagramas quedan reflejadas todas las operaciones que intervienen en dicho procesos, relacionándose y gestionando todo el negocio presentado.

El proceso de solicitar copia de documentos es uno de los más importantes que existen en la Gestión de Copia de Documentos, ya que constituye el comienzo del proceso y es donde se entran los datos de los documentos a copiar, como son: cantidad de copias, tipo de copia, entre otros metadatos que se le piden al usuario.

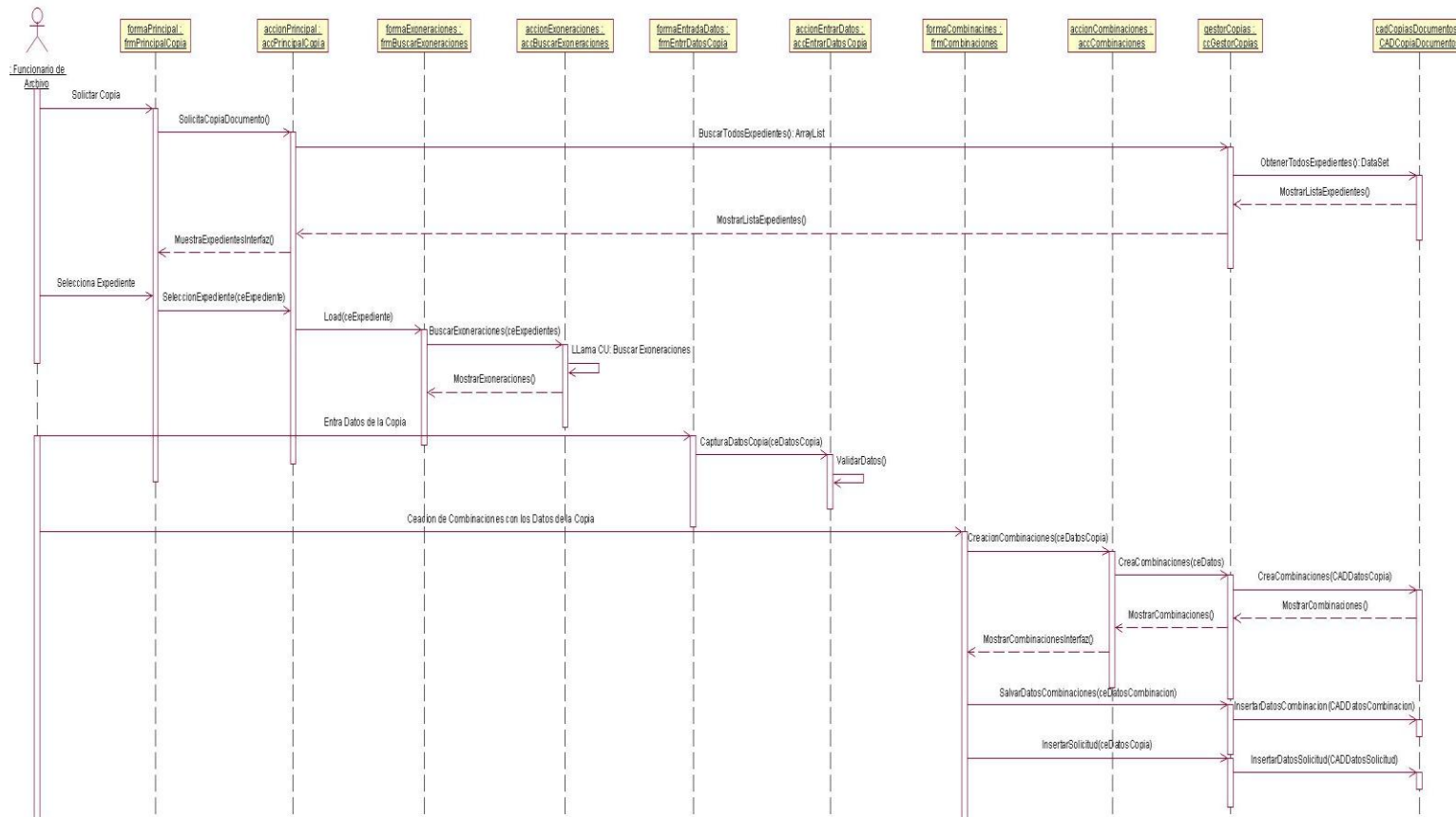


Figura 35. Diagrama de Secuencia del CU: Solicitud de Copia de Documentos.

Como se muestra en la figura anterior este proceso llama a un caso de uso auxiliar, Búsqueda de Exoneraciones, a continuación se muestra el diagrama de secuencia de cómo es que se hace dicha búsqueda, éste es uno de los procesos que son muy importantes para el cliente porque aquí se verá si este usuario tienen alguna exoneración o exención que lo exima del pago a un ente recaudador existente.

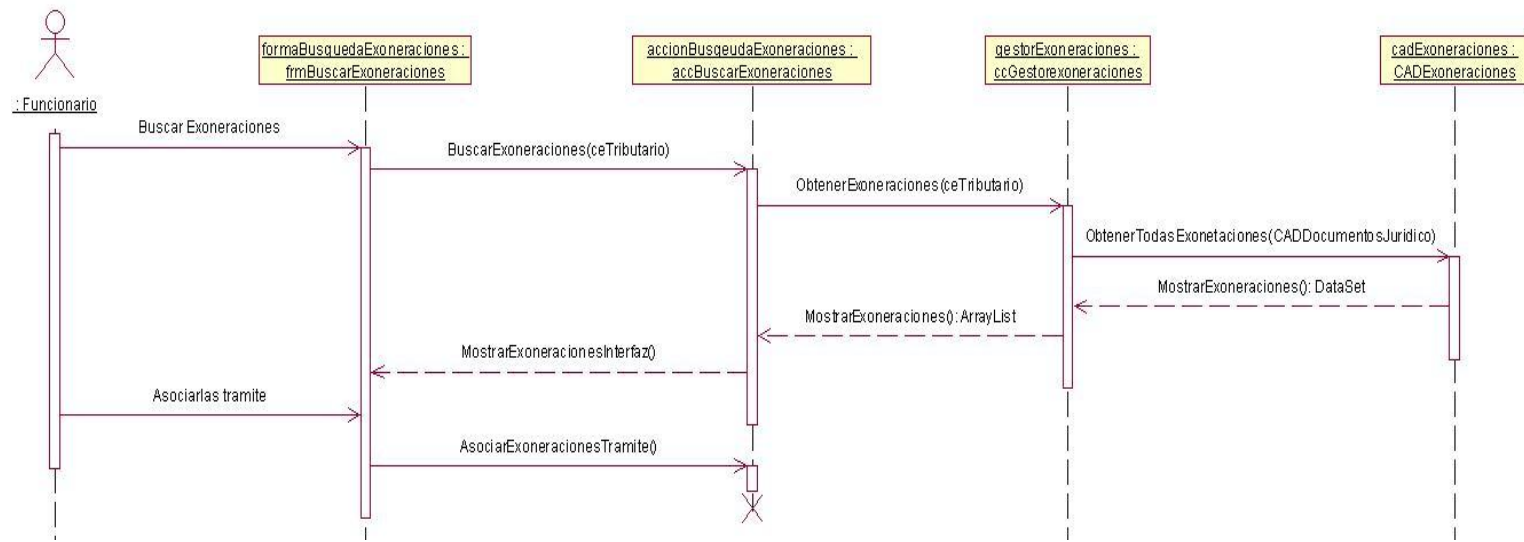


Figura 36. Diagrama de Secuencia del CU: Buscar Exoneraciones.

Anexo 7. Clases sus Atributos y Operaciones.

Las clases del sistema se encuentran identificadas en la siguiente tabla.

Número	Nombre de la Clase	Cantidad de Atributos	Cantidad de Operaciones	Tamaño de Clase
1	ETramite	14	14	Medio
2	EExpediente	14	14	Medio
3	EPersona	12	12	Medio
4	EPieza	3	3	Pequeño

5	EDocumentoJuridico	7	7	Pequeño
6	ELibro	6	6	Pequeño
7	EConcepto	10	10	Pequeño
8	EDocumento	12	12	Medio
9	GtrColeccionExpedientesAccedidos	1	4	Pequeño
10	GtrEstado	0	1	Pequeño
11	GtrEstadoArchivo	2	1	Pequeño
12	GtrEstadoDevolucion	2	1	Pequeño
13	GtrEstadoSolicitudFormal	5	2	Pequeño
14	GtrEstadoSolicitudInterna	2	1	Pequeño
15	GtrColeccionTramites	1	4	Pequeño
16	GtrColeccionExpedientes	1	4	Pequeño
17	GtrEstadoSellado	9	1	Pequeño
18	GtrColeccionDocumentoJuridico	1	4	Pequeño
19	GtrColeccionPersonas	1	4	Pequeño
20	GtrColeccionLibros	1	4	Pequeño
21	GtrEstadoCopia	9	7	Pequeño
22	GtrColeccionDocumento	1	4	Pequeño

23	GtrColeccionConcepto	1	4	Pequeño
24	SolicitudExpedienteFachada	0	9	Pequeño
25	SelladoFachada	0	4	Pequeño
26	CopiaFachada	0	4	Pequeño
27	accBuscarTramites	4	22	Medio
28	accTramitePasoDev	1	18	Pequeño
29	accDatosInformativos	0	3	Pequeño
30	accDatosInformativosDev1	1	1	Pequeño
31	accBuscarExpedientes	2	7	Pequeño
32	accBuscarExpedienteSF	2	8	Pequeño
33	accBuscarPersona	2	7	Pequeño
34	accBuscarSolicitante	1	4	Pequeño
35	accInsertarNuevaPersona	9	6	Pequeño
36	accSeleccionarPieza	2	5	Pequeño
37	accLibroControl	1	10	Pequeño
38	accDatosSolicitudSellado	3	5	Pequeño

39	accEntregarLibros	2	2	Pequeño
40	accSellarLibros	2	5	Pequeño
41	accExoneraciones	5	6	Pequeño
42	accAgregarEmpleado	4	3	Pequeño
44	accEditarDatos	5	7	Pequeño
45	accTramitePasoEditarCopia	3	20	Medio
46	accTramitePasoSoListas	2	20	Medio
47	accTramitePasoCopiaPendiente	5	26	Grande
48	accResumen	2	3	Pequeño
49	accEntrarDocumentoDuro	2	6	Pequeño
50	accBuscarSolicitanteCopia	3	4	Pequeño
51	accConfeccionarCombinaciones	3	12	Pequeño

Tabla 6. Clasificación de las clases del negocio.