

005.12
AM
D
TD-0215-07

TD-0215-07



Universidad de las Ciencias Informáticas

Facultad 3

Ingeniería Informática

*Diseño e Implementación del proceso de inscripción del
Módulo Mercantil en las Oficinas Registrales de la
República Bolivariana de Venezuela.*

Tipo de Informe: Productivo

Autores:

Kiosmy Almenares Herrera,
Rubén de León Becerra.

Tutores:

Ing. Lizandra Arza Pérez
Ing. Oscar Camacho Acosta

Resumen

Como parte de la estrategia de la Gobierno de la República Bolivariana de Venezuela de mejorar el servicio y la seguridad del ciudadano venezolano en lo referente a los procesos registrales en el territorio, se plantea el Proyecto de Modernización de los Registros y Notarías. Perteneciente a este proyecto se encuentra el módulo para la automatización de los Registros Mercantiles, refiriéndose el presente trabajo al Proceso de Inscripción en estos Registros.

El Proceso de Inscripción de Documentos trae como resultado el Otorgamiento de un documento y el archivo de este en los libros del Registro, constituyendo de esta forma la base para la realización de las demás operaciones que se realizan por ciudadanos o funcionarios de las Oficinas. El presente trabajo propone la Arquitectura del Sistema apoyándose en el estudio de los estilos arquitectónicos, las leyes y las características de la organización, brindando una solución de diseño e implementación, garantizando los parámetros de seguridad, flexibilidad y escalabilidad.

Con esta solución la Dirección de Registros y Notarías, el órgano que regula y controla la seguridad jurídica, contará con una herramienta que permita obtener la información de las actividades de los Registros de forma oportuna y eficiente a lo largo del territorio.

Índice

Agradecimientos	III
Resumen.....	V
Introducción.....	9
1. Capítulo 1.....	13
1.1. Sistema Registral.....	13
1.2. Leyes y principios que rigen los procesos registrales	14
1.3. Los Registros Mercantiles	17
1.4. Proceso de inscripción.....	19
1.4.1. ¿Qué es el proceso de inscripción?	19
1.4.2. Pasos en la inscripción de documentos	20
1.4.3. Búsqueda de denominaciones comerciales	23
1.4.4. Cálculo.....	24
1.4.5. Búsqueda de prohibiciones	25
1.5. Búsqueda fonética	26
1.5.1. ¿Qué es la búsqueda fonética?.....	26
1.5.2. Algoritmos.....	27
1.6. Plataformas para el desarrollo de sistemas de complejidad similar	28
1.7. Soluciones informáticas existentes.....	32
1.8. Metodologías y herramientas de modelado para el diseño de sistemas.....	32
1.8.1. Metodologías	33
1.8.2. Herramientas	35
2. Capítulo 2.....	37
2.1. Arquitectura	37
2.1.1. Plataforma	37
2.1.2. Estilo de arquitectura.....	38
2.1.3. Estilos de Flujos de Datos	38
2.1.4. Estilos Centrados en Datos	39
2.1.5. Estilos peer-to-peer	39
2.1.6. Estilos de Llamada y Retorno.....	40

2.1.7.	Framework.....	44
2.1.8.	Ventajas.....	45
2.1.9.	Desventajas.....	46
2.1.10.	Mejoras Propuestas.....	47
2.1.11.	Flujos de Trabajo.....	47
2.1.12.	Estándar de codificación.....	48
2.2.	Soluciones.....	48
2.2.1.	Algoritmo de búsqueda fonética.....	48
2.2.2.	Cálculo.....	50
2.2.3.	Búsqueda de Prohibiciones.....	51
2.3.	Patrones de diseño aplicados.....	52
3.	Capítulo 3.....	53
3.1.	Modelo del Diseño.....	53
3.1.1.	Diagramas de Interacción.....	53
3.1.2.	Búsquedas.....	53
3.1.3.	Cálculo.....	58
3.1.4.	Denominación Mercantil.....	58
3.1.5.	Presentación.....	64
3.1.6.	Revisión Legal.....	66
3.1.7.	Gestión Documental.....	70
3.1.8.	Diagramas de Clases.....	71
3.1.9.	Búsquedas.....	71
3.1.10.	Cálculo.....	72
3.1.11.	Denominación Mercantil.....	73
3.1.12.	Presentación.....	74
3.1.13.	Revisión Legal.....	76
3.1.14.	Modelo de Datos.....	77
3.2.	Modelo de Implementación.....	77
3.2.1.	Diagrama de Componentes.....	77
3.2.2.	Diagrama de Despliegue.....	79

Conclusiones.....	80
Recomendaciones	82
Bibliografía	83
Glosario de Términos.....	85
Anexos	88
Anexo 1	88
Anexo 2	90
Anexo 3	92
Anexo 4	94
Anexo 5	95
Anexo 6	96
Anexo 7	97

Introducción

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 los procesos registrales.

Bajo la Dirección de Registros y Notarías actualmente funcionan en el territorio venezolano 483 oficinas, laboran en las mismas aproximadamente 590 personas no vinculadas directamente desde el punto de vista legal y 6160 funcionarios (personal que labora en el procesamiento de documentos). Este conjunto de oficinas (según la existencia actual de las mismas) se encuentran clasificadas por sus funciones en 4 grupos, ellos son: Registros Civiles (21 oficinas), Mercantiles (47 oficinas), Públicos (207 oficinas) y Notarías (208 oficinas), llegando a abarcar el 98% de las transacciones por la gran demanda de sus servicios, los Registros Mercantiles y los Registros Públicos.

En 1993 surgió la Ley de Registro Público, la misma estaba encaminada a dar un respaldo legal a la iniciativa del gobierno de automatizar los registros y darle validez jurídica a los documentos en copias duras y digitales.

Algunas de las oficinas registrales que cuentan con recursos para esto, desarrollaron sistemas que automatizaran sus procesos. 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, que permita una auditoría y control de los procesos que se generan a nivel de Oficina, como son el tema de las tasas de impuestos y la precedencia (primero en tiempo, primero en derecho) que son de vital importancia y con la situación actual no se puede determinar rápido y oportunamente la información.

Entre los factores que afectan el correcto funcionamiento de los Registros y Notarías Públicas se encuentran los siguientes:

- La emisión de prohibiciones por parte de tribunales y órganos del estado, hacia personas jurídicas y naturales no llegan con la premura necesaria y en los registros donde no hay automatización estos controles pueden estar sujetos a errores personales ya que estos chequeos se hacen en libros de forma manual.

Introducción

- Los aranceles e impuestos son calculados en las oficinas y aunque existe una ley que respalde estos, cada registrador puede interpretarlos de forma diferente, como resultado los montos para las mismas operaciones pueden ser diferentes en cada territorio.
- No hay un sistema que verifique y controle el proceso por el que pasa un documento desde su presentación en el Registro hasta su Otorgamiento. De esta forma el proceso de inscripción no es auditable por parte de la Dirección.
- En la mayoría de los registros no hay un sistema que contemple la digitalización de los documentos y su validez e integración con el proceso registral.
- La Protocolización y Autenticación de documentos, para los ciudadanos y la Dirección, es hoy en día un proceso ineficiente y poco confiable.
- Particularmente, los Registros Mercantiles que son los responsables del registro de las personas jurídicas, no cuentan con la infraestructura que permita el otorgamiento de denominaciones comerciales únicas a nivel nacional. En el caso de aquellos registros que están informatizados por iniciativa propia, solo se verifica a nivel regional.
- No existe un control de todas las denominaciones mercantiles que ha tenido una compañía.
- El proceso de trasladar el conjunto de los documentos que forman el tracto documental de una compañía hacia otro registro, es ineficiente y en la mayoría de los casos implica para el usuario el volver a hacer sus trámites en el nuevo Registro. De esta forma las compañías pueden trasladar y cambiar el nombre de su compañía con el fin de evadir sanciones jurídicas.

Lo antes expuesto muestra en parte la situación vigente en los Registros y Notarías, particularmente los Registros Mercantiles, en la República Bolivariana de Venezuela, aunque esta situación está particularizada para este país, este contexto de autonomía a nivel de oficina y la falta de control a nivel superior se puede extrapolar hacia otros países. La necesidad de una automatización de estos procesos, para brindar seguridad y facilidad de uso a los usuarios, funcionarios, registradores y dirección traen consigo el planteamiento del siguiente problema: ¿Cómo lograr, mediante un sistema informático, una rápida gestión de los procesos de inscripción en los Registros Mercantiles, así como posibilitar el control local y centralizado de este proceso, garantizando de esta forma la seguridad jurídica?

Introducción

Estos procesos registrales que se llevan a cabo en las oficinas mercantiles, son tareas que desde el punto de vista legal tienen un elevado grado de importancia, además de la complejidad de cada uno de sus pasos. Dentro de los procesos que se realizan tienen vital importancia los procesos de inscripción de documentos. Teniendo en cuenta la no existencia de un sistema informático uniforme para todas las oficinas con un carácter centralizado, que gestione, garantizando la seguridad, cada uno de los pasos de los procesos registrales en la República Bolivariana de Venezuela; se llevó a cabo la confección de un Sistema Registral Mercantil automatizado, flexible, que sea capaz de soportar los cambios en leyes, tasas de aranceles e impuestos y nuevos tipos de documentos a inscribir. Todo ello controlado a nivel central y garantizando de esta forma la seguridad jurídica. Dentro de los principales aportes concretados durante el proceso de inscripción se encuentran los siguientes:

- Algoritmo para la búsqueda fonética, relacionado con la actividad de búsqueda de denominaciones.
- Solución dinámica y configurable para la realización de los cálculos.
- La solución para el tratamiento de las prohibiciones, relacionado con el paso de Revisión Legal.

El objetivo general de la investigación es diseñar e implementar una solución informática para el proceso de inscripción de documentos en los Registros Mercantiles de la República Bolivariana de Venezuela.

De este se derivan los siguientes objetivos específicos:

- Estudiar las leyes que rigen a los Registros Mercantiles de República Bolivariana de Venezuela.
- Estudiar cómo tienen lugar los procesos en los Registros Mercantiles de la República Bolivariana de Venezuela.
- Estudiar el modelo del proceso de inscripción de documentos
- Estudio y análisis de la arquitectura definida para la implementación de la solución.
- Dar una solución de diseño que permita que el sistema sea auditable y controlable a nivel central y de oficina garantizando la seguridad jurídica.
- Diseñar la solución para el proceso de inscripción de documento en los Registros Mercantiles

- Implementar la solución diseñada.

Entonces, según lo antes expuesto se puede afirmar que al diseñar e implementar un sistema que permite la inscripción de documentos en los Registros Mercantiles, se posibilita la agilización, organización y control a nivel local y de Dirección del proceso de inscripción. Contribuyendo a la publicidad registral y la seguridad jurídica.

Capítulo 1.

Este capítulo comienza refiriéndose a las principales definiciones y principios registrales que se deben tener en cuenta para la realización de un sistema automatizado, explica la función de los Registros Mercantiles y el proceso de inscripción que constituye el flujo más importante, además se hace un estudio de las plataformas y metodologías para el desarrollo de software, explicando sus características y contribuyendo a la explicación de la selección.

1.1. Sistema Registral

“Se entiende por Sistema Registral al conjunto de normas y procedimientos que tienen por objeto, la calificación e inscripción de los actos jurídicos y documentos que conforme al orden legal de cada país, deban registrarse, así como su ordenación en folios, integración y conservación de sus Archivos.” (Dra. Yarina Amoroso Fernández).

La verificación de las normas y procedimientos que rigen el Sistema Registral están a cargo de la Administración Pública, por lo que es importante conocer qué se entiende por Administración Pública.

Administración Pública: “Organización ordenada a la gestión de los servicios y a la ejecución de las leyes en una esfera política determinada, con independencia del poder legislativo y el poder judicial. Conjunto de organismos encargados de cumplir esta función.” (Real Academia Española). Una vez vista la definición de Administración Pública se puede ver la definición de Registro. *Registro:* “En las diversas dependencias de la Administración Pública, departamento especial donde se entrega, anota y registra la documentación referente a ellas.” (Real Academia Española).

En la República Bolivariana de Venezuela los Registros u Oficinas se clasifican en cuatro tipos: Registros Civiles, Registros Públicos, Notarías y Registros Mercantiles, su función está bien delimitada y cada uno se encarga de anotar y registrar la documentación que le

corresponde. La Dirección General de Registros y Notarías es la encargada de la gestión y control de los procesos que se efectúan en cada dependencia, siendo estas entidades miembros de la Administración Pública.

1.2. Leyes y principios que rigen los procesos registrales

La solución que se dará informatiza procesos de una de las oficinas del Sistema Registral, que además de ser un sistema de gestión y control la solución va a estar enmarcada en un contexto jurídico, por lo tanto se debe ajustar a la Ley y no puede violar ninguno de los principios que esta establece, siendo la seguridad jurídica un eslabón fundamental en el objetivo del desarrollo de este sistema.

La Ley que rige el funcionamiento de los registros es La Ley de Registros y Notarías (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006) identificándose dentro de ella los principios por los cuales trabajan las Oficinas o Registros. A continuación se relacionan los principios y la interpretación que se le debe dar para el buen entendimiento de cada uno de ellos, 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 (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006).

Con este principio se hace referencia a que el centro de atención en un Registro es el Documento. El proceso registral comienza con la presentación de este, una vez que cumple con los requisitos iniciales y es debidamente admitido para procesarse, con su aceptación se está asumiendo su validez para impulsar el proceso hasta su conclusión. Cabe destacar que siendo el documento el ente más importante, los datos que se recojan por el sistema estarán asociados a este 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 presentado posteriormente, salvo las excepciones establecidas en esta Ley. (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006)

La prioridad es un principio muy importante cuando se habla de temas jurídicos y registrales, en otras palabras este principio significa: primero en tiempo, primero en derechos. En caso de algún litigio jurídico el documento que mayor peso tiene es el presentado con mayor antelación. Particularmente para el caso de los Registros Mercantiles (Véase 1.3 Los Registros Mercantiles) este tema tiene mucho peso en la solicitud de denominaciones comerciales, si dos personas solicitan el mismo nombre tiene mayor prioridad aquel que lo haya solicitado primero, igualmente pasa si por algún error se ha otorgado la misma denominación a instituciones diferentes, en caso de algún litigio se aplica el principio de prioridad.

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

Como se mencionó anteriormente los registros u oficinas en la República Bolivariana de Venezuela se clasifican en cuatro tipos: Registros Civiles, Registros Públicos, Notarías y Registros Mercantiles. El principio de especialidad establece que los bienes y derechos a ser inscritos estén bien definidos y precisados por cada Registro, de manera tal que ninguno pueda asumir las funciones de otro. Esto no significa que un Documento solo pueda ser inscrito en un Registro porque los documentos pueden variar su naturaleza de acuerdo al contexto, por ejemplo: un Acta de Matrimonio puede inscribirse en un Registro Civil, pero también puede inscribirse en un Registro Mercantil porque su naturaleza es demostrar el vínculo existente entre personas relacionadas con una compañía. Tampoco significa que dos Oficinas no puedan compartir el mismo espacio físico, pero en materia jurídica y registral su espacio estará 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.

Para entender en qué consiste este principio es necesario definir brevemente que es un asiento registral.

Asiento registral: Inscripción o anotación en un registro público (Real Academia Española) Las titularidades del dominio se refieren a los títulos o documentos que pertenezcan al dominio del Registro que se esté analizando, en el caso del 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 estas inscripciones.

El principio de consecutividad 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. Más adelante se explica el modo de organización de esta información lo que viene a ser el Archivo en los Registros Mercantiles (Véase 1.4 Proceso de inscripción). 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: Sólo se inscribirán en el Registro los títulos que reúnan los requisitos de fondo y forma establecidos por la ley.

Los documentos que se presentan en el Registro deberán ser revisados por la persona o personas calificadas para la revisión de estos requisitos de fondo y forma (Véase 1.4 Proceso de inscripción) verificando de esta forma el cumplimiento de la 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.

Principio de publicidad: La fe pública¹ 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.

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.

Una vez explicados los principios que rigen el funcionamiento de los Registros se puede analizar el objeto de los Registros Mercantiles, todos estos principios en mayor o menor medida estarán presentes en los procesos u operaciones que se deberán inscribir o anotar en un Registro Mercantil.

1.3. Los Registros Mercantiles

Según La Ley de Registro Público y del Notariado de Venezuela (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006) los Registros Mercantiles tienen por objeto los siguientes puntos:

- 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.

¹ Fe Pública: 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 Española).

- 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 se debe comprender que el Registro Mercantil se refiere en otras palabras a todos los llamados comerciantes individuales, compañías y sociedades anónimas, además de todas las personas jurídica que señale la ley, siendo este término el que se emplea para referirse a todas estas entidades.

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 Española). De esta definición se observa la diferencia entre la persona natural y la persona jurídica, la primera es exclusivamente una y puede ser aquella que lleva personalmente los documentos al registro; la segunda puede ser 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, por ejemplo: Ventas, Disoluciones, Aumentos de Capital. Estas actuaciones quedan anotadas en los registros de acuerdo a un proceso legal, que se describe más adelante, (Véase 1.4 más adelante 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.

Una vez analizados los principios que rigen el proceso registral y ver el objeto del Registro Mercantil se explicará el 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. Durante el proceso de

inscripción se ponen de manifiesto los principios explicados anteriormente, se hará notar sobre estos y se destacarán los puntos a tener en cuenta para la realización del sistema.

1.4. Proceso de inscripción

1.4.1. ¿Qué es el proceso de inscripción?

En materia legal el proceso de inscripción es una formalidad contenida en el Código de Comercio² 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 (principio de publicidad), 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. Esto se ve reforzado por el hecho de que no solo se inscriben estos actos, el Registro entre otras cosas legaliza los libros de los comerciantes y registra los estados contables (Véase el objeto de los Registros Mercantiles en 1.3 Los Registros Mercantiles).

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.

En conclusión los actos y documentos sujetos a ser inscritos en el Registro Mercantil, no producirán efectos respecto de terceros sino a partir de la fecha de su inscripción.

A continuación se describen los pasos por el que debe pasar un Documento para inscribirse en el Registro Mercantil.

² (Congreso de la República Bolivariana de Venezuela, 1955)

1.4.2. Pasos en la inscripción de documentos

En el Registro Mercantil se inscriben varios tipos de documentos, de acuerdo a los estudios hechos de la Ley, la naturaleza de los Documentos que se inscriben en el Registro y los resultados que provocan producto de su inscripción, se pueden clasificar en 3 grandes grupos:

- **Constitutivos:** Traen como resultado de su inscripción la constitución de una Persona Jurídica apta para operar en territorio venezolano.
- **No Constitutivos:** Su objetivo es actualizar las actividades tanto internas (Los propios de la vida interna de la sociedad, o sea los que se refieren al comportamiento de los socios entre sí y con la sociedad como persona jurídica distinta de ellos mismos individualmente considerados) y externa (Los que surgen en la vida externa de la sociedad, o sea entre la sociedad y los terceros con quienes ella contrata).
- **Otros Actos:** Son aquellos que no están directamente vinculados a una persona jurídica y que por lo tanto no se archivan (más adelante se explica el proceso de archivar documentos) junto al conjunto de los documentos relacionados con alguna persona jurídica.

Siendo el proceso de inscripción de documentos constitutivos el más extenso y abarcador por lo que se explicará exhaustivamente para su mejor comprensión, los otros procesos salvo algunas particularidades que se explicaran más adelante son similares. Este proceso consta de cinco pasos fundamentales:

- Denominación Mercantil
- Presentación de Documentos.
- Revisión Legal
- Otorgamiento
- Archivo

Los pasos anteriormente mencionados pueden estar compuestos de pequeños pasos para llegar a su objetivo final, a continuación se analizaran y explicaran cada uno de ellos.

El objetivo de la **Denominación Mercantil** es reservar para el usuario el nombre de una compañía que posteriormente se utilizará para la legalización de todos los documentos

referentes a esta. Este paso comienza con la personación en el Registro de un usuario y la solicitud por su parte de un nombre que deberá ser único en todo el territorio de la República Bolivariana de Venezuela. Con la solicitud de este nombre se desata para el Registro la tarea de realizar la búsqueda de esta denominación y verificar que en todo el territorio no exista una persona jurídica con un nombre similar (Véase 1.4.3 Búsqueda de denominaciones comerciales). Una vez aceptado el nombre como válido por parte del Registro y la cancelación por parte del usuario de los impuestos necesarios (Véase 1.4.4 Cálculo) se procede a la reserva de la denominación quedando lista para la presentación de los documentos.

La **Presentación de Documentos** tiene como objetivo la captura de los primeros metadatos asociados al documento y las primeras verificaciones de forma del mismo, los funcionarios de los Registros le han dado nombre a cada uno de estos datos y entre estos se encuentran los siguientes:

- Presentante: Persona que presenta el documento en el Registro.
- Otorgantes: Una o más personas que firmarán el documento en Otorgamiento.
- Recaudos: Otros documentos que acompañan al Documento y que sirven para validarlo, tales como: identificación del presentante y los otorgantes entre otros.
- Otros datos asociados al documento.

Una vez capturados estos datos y realizada la cancelación de los impuestos (Véase 1.4.4 Cálculo) se procede a la firma del libro diario donde queda constancia de la fecha y hora en la que se completa el paso de presentación, quedando listo para el próximo paso.

La **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 (Véase 1.2 Leyes y principios que rigen los procesos registrales). El principal objetivo de este paso es dar el visto bueno del Documento desde el punto de vista legal, dejándolo listo para Otorgamiento, pero además como en este paso es donde se hace la revisión exhaustiva se capturan otros datos que no se puedan capturar en la presentación, entre los que están: información de los socios, modificaciones a la compañía que deban quedar asentados como metadatos de esta y personal relevante como administradores o gerentes. Como parte del

proceso también se realiza una verificación de prohibiciones (Véase 1.4.5 Búsqueda de prohibiciones).

El **Otorgamiento** es el paso donde culmina para el usuario el proceso de inscripción, aquí se entrega el documento original firmado por el Registrador 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 prohibiciones. Una vez culminado este paso el usuario puede hacer uso de este documento ante terceros culminado de esta forma el proceso de inscripción para el usuario aunque el proceso en el Registro continúa.

El paso de **Archivo** tiene como objetivo la consecuente anotación en los libros del Documento que fue asignado a estos en el paso de Otorgamiento, el archivo en los Registros Mercantiles se organiza en protocolos A, B y C, estos a su vez están compuestos por tomos, o sea libros que contienen un conjunto de documentos. Los tomos y los documentos dentro de los tomos están enumerados de forma consecutiva, poniéndose de manifiesto el principio de consecutividad (Véase 1.2 Leyes y principios que rigen los procesos registrales). Existe otra agrupación de documentos en el Archivo que son los expedientes, existe uno por cada persona jurídica que inscriba sus actos o contratos en el Registro. Los expedientes están divididos en piezas las cuales son un conjunto de documentos, tanto las piezas dentro del expediente y los documentos dentro de las piezas se encuentran enumerados, poniéndose nuevamente de manifiesto el principio de consecutividad.

Las particularidades para el proceso de inscripción de actos no constitutivos es que estos no comienzan por una Denominación Mercantil sino que se hacen sobre una persona jurídica ya inscrita, al igual que los constitutivos van a quedar anotados en el protocolo y expediente correspondientes.

Los otros actos tienen como particularidad que no se anotan en un expediente porque no pertenecen o constituyen a ninguna persona jurídica. Entre estos están los Poderes, Autorizaciones Judiciales y Sentencias de Divorcio.

Una vez analizado el proceso de inscripción se analizarán aquellos pasos que afectan el proceso pero que es necesario tener en cuenta otras consideraciones a la hora de dar una

solución automatizada, además, su contribución a la garantía jurídica es importante y su solución puede acarrear una complejidad mayor.

1.4.3. Búsqueda de denominaciones comerciales

Partiendo del principio de que no pueden existir denominaciones repetidas en el territorio nacional, es de vital importancia el entendimiento del tratamiento que se da en el Registro a la búsqueda de coincidencia en las denominaciones mercantiles en estos momentos.

Este paso comienza una vez que se ha hecho la solicitud de una denominación comercial por parte de un usuario al Registro, después de esta solicitud el funcionario tiene que hacer una búsqueda de las denominaciones solicitadas u otorgadas anteriormente, si esta búsqueda arroja resultados similares el funcionario denegará la solicitud, en caso contrario aceptará la denominación como válida y el nombre quedará listo para ser otorgado a un usuario. El funcionario es el responsable de que la negación o aceptación de una denominación como válida esté acorde a los requisitos establecidos, por lo tanto su criterio es el último y es quien toma la decisión. Con el otorgamiento de este nombre se está asumiendo su unicidad a lo largo del territorio venezolano, por esta razón este paso es de vital importancia.

El sistema que automatice esta búsqueda deberá arrojar como coincidencias aquellas denominaciones de compañía cuyo nombre sea similar al buscado, por ejemplo: Coca Cola y Cola Coca, son nombre similares al igual que el siguiente, Sonidos **Ercaloza** y Audios **Calosa**. Por la variedad de posibilidades que se pueden encontrar el sistema podría dejar la decisión final al funcionario y su objetivo sería ayudar en el proceso de búsqueda.

Otras consideraciones que se deben tener en cuenta es el hecho de que estas denominaciones deben ser únicas en toda Venezuela. Además si dos personas solicitan el mismo nombre aunque ninguno haya sido otorgado tendrá más derechos aquel que haya hecho su solicitud con antelación, con el objetivo de garantizar el principio de prioridad (Véase 1.2 Leyes y principios que rigen los procesos registrales).

A continuación se explicará otro de los pasos que se utilizan en el proceso de inscripción al cual se le debe prestar también especial atención, este es el paso relacionado con el cálculo de los aranceles o impuestos.

1.4.4. Cálculo

El cálculo es un paso que se hace en varios lugares dentro del proceso de inscripción, su objetivo es calcular los aranceles o impuestos para las operaciones, actos o servicios que se efectúan en el Registro. Durante el proceso de inscripción se deberán pagar los siguientes impuestos:

- Solicitud de Denominación Comercial: Estos impuestos van al ente recaudador Servicio Autónomo de Registros y Notarías.
- Reserva de Denominación Comercial: Estos impuestos van al ente recaudador Fisco.
- Presentación de Documentos: Se cobran impuestos que van a ambos entes recaudadores.

Entre las consideraciones que se deben tener en cuenta en la realización de este proceso están las siguientes:

- El cálculo es regido por la Ley de Registro Público y del Notariado de Venezuela (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006) Ejemplo: Artículo 86 Numeral 1. Por la inscripción de cualquier tipo de sociedades, firmas personales, y asociaciones de cuentas en participación, dos unidades tributarias (2 U.T.), más una décima de unidad tributaria (0,1 U.T.) por cada folio que contenga el documento o actuación.
- La unidad tributaria es un valor que se fija cada año y en base a esta se calculan los impuestos.
- Las leyes son documentos que quedan derogados y que pueden cambiar por modificaciones o emisiones de nuevas leyes, por lo tanto artículos que se calculaban de una forma podrán pasar a calcularse de otra.

Por último se analiza otro de los pasos que se utilizan en el proceso de inscripción en este caso en más de una ocasión, es de vital importancia porque contribuye garantizar el principio de legalidad y está relacionado con aquellas prohibiciones que pueden impedir la inscripción de un Documento.

1.4.5. Búsqueda de prohibiciones

Las prohibiciones son documentos emanados de tribunales que se circulan a nivel territorial y pueden afectar a personas naturales, jurídicas o inmuebles, su objetivo es prohibir alguna acción sobre alguna de estas entidades y por consiguiente para lograr el principio de legalidad (Véase 1.2 Leyes y principios que rigen los procesos registrales) se deben chequear durante el proceso de inscripción.

El chequeo de prohibiciones se hace en dos pasos durante el Proceso de Inscripción, en la Revisión Legal de los documentos y antes de realizar el Otorgamiento, su objetivo es verificar que las personas que están involucradas en el documento no tengan ninguna limitación impuesta por tribunales. Por ejemplo si una compañía está en quiebra se podría emitir por parte de un tribunal una orden de que esta no puede realizar inscripciones de ningún tipo.

Entre las características significativas de este proceso y que se deben tener en cuenta para la realización del sistema están las siguientes:

- La naturaleza de las prohibiciones es variada y puede estar afectada por los cambios en las leyes. Las personas más capacitadas para determinar si un documento es afectado por alguna prohibición son los abogados.
- Las prohibiciones pueden afectar las diferentes personas, jurídicas y naturales, que están envueltas en el proceso de inscripción, la compañía en cuestión, el presentante, otorgante(s), accionista(s) u otra persona que se mencione en el documento.
- Cuando a un documento se le asocia una prohibición este queda paralizado hasta tanto no quede derogada la prohibición o el afectado resuelva su caso a nivel central si ese fuera el caso.

Como se mencionó anteriormente esta revisión se hace en dos pasos, Revisión Legal y Otorgamiento, el primero es el que hace la revisión exhaustiva de todas las prohibiciones que se han emitido y el segundo se hace con el principal objetivo de chequear si en el tiempo transcurrido entre la Revisión Legal y el Otorgamiento se emitió otra prohibición que pudiera impedir el Otorgamiento del documento.

Una vez analizado el proceso de inscripción y dentro de este la búsqueda de coincidencias se observa que este último se refiere a encontrar sonidos similares o compañías que su nombre

sea similar, por lo tanto es necesario abundar en estos temas, su significado y algoritmos más utilizados, lo cual veremos en el siguiente epígrafe.

1.5. Búsqueda fonética

Hace muchos años surgió la necesidad de identificar las variantes³ de nombres personales. Esto conformó una situación problemática que diversas aplicaciones han tratado, tales como los sistemas de recuperación de información (SRI), las bibliotecas digitales, las bases de datos de pacientes en un hospital, los sistemas de reservas aéreas, o los sistemas de censo. La identificación y uso de las diferentes variantes de palabras que pueden asemejarse desde el punto de vista de su pronunciación, es a lo que se le llama comparación fonética.

1.5.1. ¿Qué es la búsqueda fonética?

Este tipo de búsquedas consiste en la codificación fonética⁴ de palabras a través de un algoritmo, obteniendo códigos fonéticos que no son más que una combinación de números y letras formando una cadena para luego realizar una comparación entre estos resultados y no directamente con las palabras iniciales. La codificación fonética es el resultado de transformar a una forma común aquellos nombres personales que son similares en cuanto a su pronunciación, haciendo más sencilla la comparación de una cadena con otra, debido a que se almacena el código generado en lugar del nombre completo (véase el epígrafe 1.5.2 para más información sobre códigos fonéticos). Sin embargo, la principal limitante en sistemas de este tipo de búsquedas es que son dependientes del lenguaje utilizado, lo que hace necesario la realización de modificaciones de acuerdo al idioma que se va a emplear. Cada palabra está compuesta por caracteres que a su vez son fonemas del idioma, los cuales tienen el nivel más elevado de importancia cuando se trata de evaluar y comparar pronunciaciones en una lengua. Los fonemas son una unidad fonológica (indica el modo de pronunciación) diferenciadora, indivisible y abstracta.

³ Diferentes combinaciones de palabras que fonéticamente suenan de la misma manera cuando se pronuncian.

⁴ Rama de la lingüística que estudia la producción, naturaleza física y percepción de los sonidos de una lengua.

- **Diferenciadora:** Porque cada fonema se delimita dentro del sistema por las cualidades que se distinguen de los demás y además es portador de una intención significativa especial. A modo de ejemplo, tenemos que *Pota* y *Bota* son dos palabras que se distinguen semánticamente debido a que “b” se opone a “p” por la sonoridad.
- **Indivisible:** No se puede descomponer en unidades menores.
- **Abstracta:** No son sonidos, sino modelos o tipos ideales de sonidos.

Se puede concluir entonces que la búsqueda fonética depende de la correcta identificación de uno u otro algoritmo⁵ (dependiendo en gran medida de la lengua donde se usara) para la obtención de códigos fonéticos aceptados, ya que luego solo restaría comprar estos códigos resultantes por el método convencional para comparar de dos palabras.

1.5.2. Algoritmos

Los algoritmos para el proceso de búsquedas y comparaciones fonéticas parten de la codificación basada en la similitud fonética de los nombres personales. Esto se aplica principalmente a los nombres y apellidos para reducirlos a una forma común. Los procedimientos de codificación fonética más conocidos se encuentran los sistemas Soundex (Odell & Russell, 1918), Daitch-Mokotoff Soundex (Daitch & Mokotoff, 1985), Phonix (Gadd, 1988, 1990), Metaphone (Philips, 1990) y NYSIIS (Taft, 1970).

De los antes enumerados varios son variantes ajustadas al llamado Soundex, del cual se explica a continuación su principio de funcionamiento:

Este algoritmo se hizo para reducir particularmente apellidos ingleses, a un código de cuatro caracteres. El primer carácter es una letra mayúscula y los tres restantes son dígitos. Knuth (1973) describe el procedimiento utilizado por Soundex por medio de una función que consiste en:

- La conversión de caracteres a un código fonético, tal y como aparece en la tabla 1.
- Un algoritmo que sustituye todos los caracteres, excepto el primero, por su correspondiente código fonético.
- La eliminación de cualquier repetición consecutiva de caracteres.

⁵ Es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema.

- La devolución únicamente de los primeros cuatro caracteres de la cadena resultante.

Tabla # 1 - Códigos fonéticos de Soundex

Código	Caracteres
0	a e h i o u w y
1	b f p v
2	c g j k q s x z
3	d t
4	L
5	M n
6	R

La siguiente tabla 2 se muestra un ejemplo de la representación en códigos fonéticos basados en Soundex de algunos nombres ingleses.

Tabla # 2 - Resultado de la aplicación de Soundex

	Apellidos	Variantes	Códigos
Coincidencias	Appelt	Apelt	A143, A143
	Hobbs	Hubbs	H120, H120
No Coincidencias	Appelt	Appell	A143, A140
	Hobbs	Hobds	H120, H130

El sistema Soundex se usa actualmente por el National Archives and Records Administration (NARA) de EE.UU. Existen actualmente en el mundo disímiles variantes que resuelven problemáticas de este algoritmo, como es el caso de la Daitch-Mokotoff que permite la aplicación de este algoritmo a diferentes apellidos judíos, germánicos o eslavos. También la llamada Phonix, que permite la ordenación entre cadenas similares y finalmente entre otras se encuentra Metaphone que es muy semejante a los anteriores pero especialmente diseñado para el inglés americano. Para conocer una información más detallada de estas diferentes variantes consultar el **Anexo 1**.

1.6. Plataformas para el desarrollo de sistemas de complejidad similar

Desde el enfoque de la realización de sistemas informáticos, se le denomina plataformas a los diferentes ambientes creados para el desarrollo de software. Actualmente en el mundo existen

diferentes tipos de plataformas de desarrollo para aplicaciones electrónicas y de escritorio. En los últimos años los ambientes de programación han ido dirigiéndose especialmente al bienestar del desarrollador y a la unificación en un mismo entorno de desarrollo, 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, lenguajes de programación (con su ambiente visual en algunos casos) para desarrollar sistemas. Por citar algunos ejemplos se encuentran el Borland Delphi , Borland C++ y el Visual Basic. En estos momentos con la aparición de la plataforma .Net de Microsoft los conceptos del desarrollo de aplicaciones han sufrido un cambio radical por el simple hecho de unificar en un único ambiente de desarrollo múltiples lenguajes de programación representados por un Framework (marco de trabajo)⁶ común. De las diferentes plataformas existentes en el mundo para el desarrollo de aplicaciones de escritorio se encuentran:

- **Java y su Máquina Virtual (*Virtual Machine*⁷):** Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que un programa escrito para Linux puede ser ejecutado en Windows. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas. El código que generan los compiladores de Java no es particular de una máquina física, sino de una máquina virtual. Aún cuando existen múltiples implantaciones de la Máquina Virtual Java, cada una específica de la plataforma sobre la cual subyace, existe una única especificación de la máquina virtual, que proporciona una vista independiente del hardware y del sistema operativo sobre el que se esté trabajando.

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

⁶ Denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

⁷ Es un 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.

una máquina virtual para soportar abstracciones orientadas a objetos de alto nivel, sobre las máquinas subyacentes.

- **.Net Framework:** Es una plataforma de software que conecta información, sistemas, personas y dispositivos. La plataforma .NET conecta una gran variedad de tecnologías de uso personal y de negocios, de teléfonos celulares a servidores corporativos, permitiendo el acceso a información importante, donde y cuando se necesiten. Desarrollado con base en los estándares de Servicios Web XML⁸, .NET permite que los sistemas y aplicaciones, ya sea nuevos o existentes, conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo. 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:
 - **Código administrado:** El CLR⁹ (*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 (MSIL¹⁰).
 - **Compilación just-in-time:** El compilador JIT¹¹ 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.

⁸ Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones

⁹ Entorno común de ejecución para los diferentes lenguajes del .Net Framework.

¹⁰ Es un código de bites que la Tecnología .NET de Microsoft utiliza para lograr independencia de la plataforma y seguridad en ejecución.

¹¹ Es una técnica para mejorar el rendimiento de sistemas de programación en las tecnologías .NET.

- **Recolector de basura:** El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (Garbage Collector¹²). 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.
- **Mono:** Es un proyecto de código abierto impulsado por Novell para crear un grupo de herramientas libres, basadas en GNU/Linux ¹³y compatibles con .NET según lo especificado por el ECMA¹⁴. Entre los principales componentes de los que dispone como plataforma están los siguientes:
 - Una máquina virtual de lenguaje común de infraestructura (CLI) que contiene un cargador de clases, un compilador en tiempo de ejecución (JIT), y unas rutinas de recolección de memoria.
 - Una biblioteca de clases que puede funcionar en cualquier lenguaje que funcione en el CLR (Common Language Runtime).
 - Un compilador para el lenguaje C#, MonoBasic (la versión para mono de Visual Basic), Java y Python.
 - Es un proyecto independiente de la plataforma. Actualmente Mono corre sobre Linux, FreeBSD, UNIX, Mac OS X, Solaris y plataformas Windows.

¹² Mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación

¹³ Uno de los variados sistemas operativos libres.

¹⁴ Organización internacional basada en membrecías de estándares para la comunicación y la información.

Esta plataforma presenta una importante ventaja, y es que 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.

1.7. Soluciones informáticas existentes

Actualmente existen en la República Bolivariana de Venezuela diferentes aplicaciones informáticas que brindan servicios en el proceso de inscripción de los Registros Mercantiles. La gran mayoría de estas aplicaciones son módulos separados entre los cuales no existen vínculos. Estas aplicaciones difieren unas de otras cuando se trata de diferentes registros. Es decir, cada oficina registral cuenta con uno o varios módulos diferentes de software que contribuyen a la realización del Proceso de Inscripción en estas oficinas. Se quiere llevar a cabo la realización de una solución informática contenedora de todos los pasos de dicho proceso, con un carácter centralizado y controlado donde toda la información estará relacionada.

1.8. Metodologías y herramientas de modelado para el diseño de sistemas

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. Existen en el mundo 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. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de 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. Yourdon dice que consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc.
- **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 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. De manera similar, alrededor de 2/3 de toda la ingeniería civil, arquitectura y trabajo de construcción es dar mantenimiento.

1.8.1. Metodologías

Existen dos grupos en los cuáles se dividen estas metodologías (los métodos tradicionales y los procesos ágiles). Todas contribuyen a un buen y organizado desarrollo de software aunque tengan sus marcadas diferencias. A continuación una breve descripción de dos de estas metodologías:

- **Rational Unified Process (RUP):** Esta metodología divide en 4 fases el proceso de desarrollo de un software. Comienza por la fase nombrada *Inicio* (donde se determina la visión de todo el proyecto), le sigue la *Elaboración* (determinar la arquitectura óptima), la *Construcción* (lograr la obtención de la capacidad operacional inicial), y finalmente la *Transición* (lograr obtener una versión del software en cuestión).

Estas etapas se llevan a cabo usando un ciclo de iteraciones, y los objetivos de una iteración se establecen en función de la evaluación de las iteraciones anteriores. Además, vale destacar que el ciclo de vida que se desarrolla por cada iteración es aplicado bajo dos disciplinas, una es de desarrollo y la otra es la de soporte. En el **Anexo 2** se puede ver una descripción más detallada de los procesos que define esta metodología.

- **XP (Extreme Programming):** Es la metodología más utilizada de los procesos ágiles del desarrollo de software. Esta, a diferencia de los métodos tradicionales pone más énfasis en la adaptabilidad que en la previsibilidad. El principio de este proceso es el hecho de tener en cuenta la posible y natural aparición de cambios de requisitos durante el desarrollo del software, por lo cual esta metodología presenta los mecanismos necesarios para adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:
 - ✓ El cliente define el valor de negocio a implementar.
 - ✓ El programador estima el esfuerzo necesario para su implementación.
 - ✓ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
 - ✓ El programador construye ese valor de negocio.
 - ✓ Se regresa al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. Para más información sobre esta metodología XP léase el **Anexo 3**.

La metodología RUP (Rational Unified Process), fue la seleccionada por la dirección del proyecto Registro y Notarias como la adecuada para llevar a cabo la confección de dicho

proyecto, en el cuál se incluye el Módulo Mercantil al cual nos referimos en este documento. Esta metodología además de varias de sus características antes mencionadas, se identificada como perteneciente a los procesos de desarrollo del tipo pesados. Se le llama de esta forma por la extensión en tiempo de desarrollo, así como el gran número de especialistas necesarios para la confección del Software. A diferencia de las metodologías ágiles, dentro de las cuáles se encuentra XP (Extreme Programming), que funcionan con un número inferior de iteraciones y artefactos¹⁵, así como un menor número de especialistas y finalmente el tiempo de desarrollo es relativamente corto. Además, y de gran importancia destacar, el uso de metodologías ágiles requiere de un contacto prácticamente constante con el cliente, es decir, el cliente forma parte del proceso de desarrollo del software. Lo cual es una precondition para cumplir con los plazos y buen desarrollo del software que se desea. Muy difícilmente las diferentes metodologías pueden ser comparadas en el mayor número de los casos, ya que cada una de ellas es aplicada a uno u otro ambiente de desarrollo determinado por variables como lo son: magnitud de lo que se desea desarrollar, tiempo con el cual se cuente, número del personal con el cual se trabajara, complejidad de los diferentes módulos.

Se considera que el uso de la metodología RUP para la confección de este proyecto fue correcto, según la magnitud y extensión del funcionamiento de su negocio, el gran número de especialistas involucrados, así como su proceso de desarrollo a distancia, no se determino aplicar una metodología más ágil y si una correcta y robusta captura de requisitos, lo cual es exigido como premisa fundamental por la metodología RUP.

1.8.2. Herramientas

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, calculo de costes, implementación de parte del código automáticamente con el

¹⁵ Herramientas organizativas identificadas para la confección organizada y correcta del Software en cuestión.

diseño dado, compilación automática, documentación o detección de errores entre otras. 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 en el mundo actualmente:

- **Rational Rose.**
- **Visual Paradigm for UML.**
- **Umbrello.**
- **ArgoUML.**

Capítulo 2

En este capítulo se hace una valoración de las posibles arquitecturas a escoger para la realización del sistema, de cada una se explica por qué no es apropiada para la realización del sistema y se exponen los argumentos de la seleccionada. Se explican las soluciones más complejas, importantes y flexibles a las que debía adaptarse la solución, además se exponen los patrones más utilizados para la realización del software.

2.1. Arquitectura

2.1.1. Plataforma.

La dirección del proyecto para la Modernización de los Registros y Notarías de Venezuela escogió para el desarrollo, la **plataforma .NET**. Hay varios elementos que se tuvieron en cuenta para esta decisión:

- El tiempo de desarrollo para la primera entrega del proyecto era corto, se disponía de 2 meses para entregar un prototipo funcional y ser mostrado a la parte Venezolana.
- No se tenía experiencia por el equipo de desarrollo en construcción de proyectos de complejidad similar.
- El número de personas que tenían dominio sobre la plataforma .NET en la universidad era superior a los de la plataforma Java.
- Existía un precedente por parte de otro equipo de desarrollo en la construcción de un proyecto también para la República Bolivariana de Venezuela, hecho en .NET.
- La plataforma Mono no estaba concluida y no se consideraba un Framework estable y maduro, además, los entornos de desarrollo no ofrecían las mismas ventajas y velocidad de trabajo que en el Visual Studio .NET.

En base al estudio de plataformas realizado (Véase **Error! Reference source not found. Error! Reference source not found.**) y los factores que se mencionaron. La plataforma Java no era una buena opción, existía poco conocimiento sobre esta y para el desarrollo del proyecto en corto período de tiempo era vital que el equipo de desarrolladores estuviera preparado, lo mismo sucedía con la plataforma Mono que se veía agravado por el hecho de no contar un entorno de desarrollo tan avanzado como Visual Studio .Net

Por todas estas razones se considera la elección de la plataforma .NET una buena decisión, el entorno de desarrollo posibilitó dar una respuesta mucho más eficiente en cuanto a tiempo a lo

que en otras plataformas hubiera significado más horas de construcción. Se contaba entonces con una plataforma estable que resolvía algunos temas como la administración de memoria (Recolector de basura), administración del código y seguridad de acceso a este. Además permitió la reutilización de componentes libres y otros desarrollados por proyectos de la Universidad. Una vez escogida la plataforma comenzaba la tarea de decidir qué estilo de arquitectura escoger para la construcción de la solución, lo cual veremos en el próximo epígrafe.

2.1.2. Estilo de arquitectura.

Antes de escoger la arquitectura a utilizar para el desarrollo de software es necesario hacer un recorrido por los diferentes estilos de arquitecturas, no es objetivo el análisis en su totalidad, sino solo aquellos que por su estructura y solución podrían considerarse para la construcción del sistema, viendo las ventajas y desventajas que poseen para determinar si deben ser o no ser adoptados para el desarrollo de la solución.

2.1.3. Estilos de Flujos de Datos

La familia de Estilos de Flujos de Datos, esta enfatiza la reutilización y la modificabilidad, es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos, ejemplos de ella podemos encontrar las Arquitecturas de Tuberías y Filtros y las de Proceso Secuencial en Lote.

“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.” (Reynoso, y otros, 2004)

Ninguna de las arquitecturas que ofrece esta familia de estilos se recomienda para el desarrollo de un sistema con las características que se necesita. A pesar de que enfatiza la reutilización, la modificabilidad y que en el proceso de inscripción el documento pasa por determinados pasos hasta llegar a la culminación de su procesamiento en el Registro, este documento no recibe transformaciones sino que se le van agregando metadatos a lo largo de él flujo. Además entre las desventajas del uso de este estilo están las siguientes:

- “El patrón puede resultar demasiado simplista, especialmente para orquestación de servicios que podrían ramificar la ejecución de la lógica de negocios de formas complicadas.” (Reynoso, y otros, 2004) El negocio en los Registros Mercantiles en un proceso complejo, puesto que es necesario que el sistema garantice la seguridad jurídica, donde se deben realizar muchas validaciones tanto a nivel de usuario como de negocio.

- “No maneja con demasiada eficiencia construcciones condicionales, bucles y otras lógicas de control de flujo. Agregar un paso suplementario afecta la performance de cada ejecución de la tubería.” (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.” (Reynoso, y otros, 2004) Tal y como sucede con los procesos por los que pasa un documento.

2.1.4. Estilos Centrados en Datos

Otro estilo arquitectónico es el Centrado en Datos, esta familia enfatiza la integración de los datos. Entre los subestilos característicos están los Repositorios, Bases de Datos y Arquitecturas de Pizarra.

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

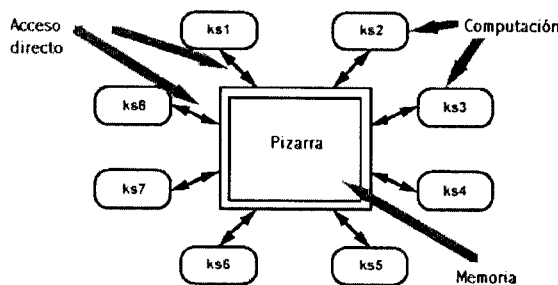


Imagen 1: Arquitectura de Pizarra o Repositorio.

Este estilo de arquitectura “se utiliza para aplicaciones que requieran complejas interpretaciones de procesos de señales (reconocimiento de patrones, reconocimiento de habla) o sistemas que involucren acceso a datos con agentes débilmente acoplados.” (Reynoso, y otros, 2004)

No se considera apropiada la arquitectura anteriormente expuesta para la realización del sistema, esta enfatiza los datos y no el modelo, además los conceptos de reutilización y flexibilidad no son sus objetivos. En el entorno del sistema o negocio del Registro Mercantil no existen los componentes individuales que actúan sobre la pizarra o repositorio.

2.1.5. Estilos peer-to-peer

Los estilos peer-to-peer enfatizan 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.

Entre los estilos que componen esta familia se encuentran, las Arquitecturas Basadas en Eventos y Arquitecturas Orientadas a Servicios.

A pesar de que los eventos se utilizan para manejar las interfaces de usuario, desde el punto de vista arquitectónico suponen que los métodos o procedimientos no se llaman de forma directa sino que uno o más de estos métodos puede ser disparado por un evento, de esta forma los módulos ofrecen tanto una interfaz de métodos como una interfaz de eventos. Es importante destacar que las operaciones que suceden mediante eventos son engorrosas de depurar. El estilo más bien se utiliza en ambientes de integración de herramientas, interfaces de usuario o cuando se desee que no exista un acoplamiento entre el emisor del evento y el receptor.

Entre las desventajas que presenta este estilo están las siguientes:

- El estilo no permite construir respuestas complejas a funciones de negocios. (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 surgir problemas de performance global y de manejo de recursos cuando se comparte un repositorio común para coordinar la interacción. (Reynoso, y otros, 2004)

Por estos motivos no se escogió esta arquitectura en un negocio que tiene sus propias reglas y donde es muy importante el rendimiento. Destacando la flexibilidad y la reusabilidad.

La Arquitectura Orientada a Servicios no se aplica en este contexto por ser la Oficina o Registro un ente de cierta forma independiente y que debe funcionar pese a problemas de conectividad.

2.1.6. Estilos de Llamada y Retorno.

Los estilos de Llamada y Retorno son los más utilizados en sistemas a gran escala, estos enfatizan la modificabilidad y la escalabilidad. Dentro de ellos se encuentran el Modelo-Vista-Controlador (MVC), Arquitectura Orientada a Objetos, Arquitectura Basada en Componentes y Arquitecturas en Capas.

El **Modelo-Vista-Controlador** es un estilo de arquitectura ampliamente utilizado, ha sido propio de las aplicaciones de Smalltalk ¹⁶al menos desde 1992 (Reynoso, y otros, 2004), se conoce también como un patrón de diseño y se le han dado numeras implementaciones en C++ y ASP.NET, Implementando Modelo-Vista-Controlador en ASP.NET (Trowbridge, y otros, 2003).

La idea fundamental de este estilo de arquitectura es separar sus tres componentes, el modelo, la vista y el controlador de manera tal que puedan ser reutilizados y desarrollados de forma paralela por personal con diferentes aptitudes. Para el desarrollo del sistema se escoge un Framework existente en la Universidad el cual se basa en el MVC, (Véase 2.1.7 Framework.) de esta forma se aprovechan las ventajas de este estilo o patrón, pero desde el punto de vista de arquitectura la elección sería diferente.

La **Arquitectura Orientada a Objetos o Basada en Objetos** como también se le conoce, es una de las más difundidas, sobre todo con el advenimiento de la Programación Orientada a Objetos después de la Programación Estructurada. En este tipo de arquitectura el centro son los objetos y las interacciones entre estos sin importar su naturaleza local o remota, ellos son los responsables de mantener su integridad y por lo tanto su representación interna no debe ser accesible desde otros objetos.

El principal problema de este tipo de arquitecturas es que en sistemas de gran tamaño, las interacciones entre los objetos y su volumen se pueden volver inmanejables, es necesario tener un nivel de abstracción superior que agrupe estos objetos de modo tal que sean más fáciles de entender y desarrollar en paralelos por diferentes grupos de desarrolladores.

No se seleccionó este tipo de arquitectura pero el hecho de haber escogido la plataforma .NET para el desarrollo se considera una buena elección. (Véase 2.1.1 Plataforma.) En ella se aprovechan las ventajas de la Programación Orientada a Objetos y se deja pie para escoger una arquitectura que no necesariamente sea basada en objetos y aportará más para el desarrollo de la solución.

La **Arquitectura Basada en Componentes** es una tendencia que surgió después de la Programación Orientada a Objetos, debido a la necesidad creciente de la reutilización de código, pues aunque los objetos se puedan reutilizar, el componente podría entregarse o venderse como un paquete, exponiendo una interfaz y manteniendo su integridad. El desarrollo de una Arquitectura Basada en Componentes lleva un esfuerzo por parte del equipo de desarrollo, si se tiene en cuenta que se está en presencia de un negocio que tiene propias

¹⁶ **Smalltalk**: Es un 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.

reglas, no sería apropiado emplear este tipo de arquitectura; lo cual no excluye la posibilidad de desarrollar componentes para aquellas funcionalidades que podrían emplearse en el desarrollo de otros sistemas.

Por último se analizará la Arquitectura en Capas, esta fue la escogida para el desarrollo del sistema, por lo que se profundiza en las ventajas y desventajas que esta ofrece.

La **Arquitectura en Capas** podría decirse que su finalidad es abstraer las funcionalidades de una capa, de manera tal que esta pueda ser totalmente reemplazada. La Arquitectura de Capas más común es la que está compuesta por tres, Presentación, Modelo o Reglas del Negocio de la Empresa y Acceso a Datos. De esta forma se podrá cambiar cualquiera de estos sin afectar a las restantes. Aunque tres capas es lo más común, a medida que aumenta la complejidad de los sistemas, las capas crecen. A su vez cada capa puede estar compuesta por subcapas y una capa o subcapa puede estar compuesta por una o más clases del diseño.

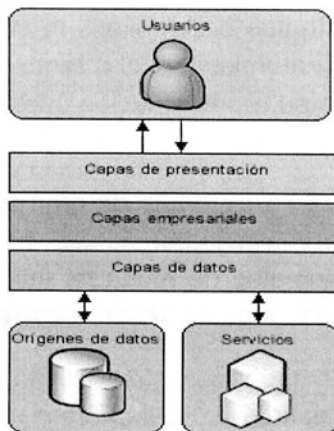


Imagen 2 Arquitectura en tres capas.

Esta característica de la arquitectura en capas permite implementar las reglas del negocio en una capa aparte, de modo que estas reglas puedan ser usadas por otros sistemas o por servicios que necesiten de estas para su funcionamiento, evitando la duplicación de código y una mayor organización.

Existen dos formas de concebir una arquitectura en capas, de manera rígida y de manera flexible, en la primera de estas una capa solo puede interactuar con su capa inmediatamente inferior, en la segunda, una capa superior puede interactuar con capas inferiores. El escoger una u otra constituyen un balance entre flexibilidad y rendimiento, mientras la primera mantiene un bajo acoplamiento puede implicar impactos negativos en el rendimiento, el modo relajado aumenta el rendimiento pero implica menos flexibilidad.

En el libro Enterprise Solution Patterns (Trowbridge, y otros, 2003) se refieren a las ventajas y desventajas de una arquitectura en capas:

Ventajas

- El mantenimiento y las mejoras a 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.
- 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.
- Distribuir las capas a lo largo de múltiples capas físicas puede mejorar la escalabilidad, tolerancia a errores y rendimiento.
- 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.

La primera de estas ventajas es importante para el desarrollo del sistema, pues el sistema deberá permitir cambios ya sea por modificación de los requisitos o agregaciones o mejoras de las funcionalidades. El hecho de que se puedan reutilizar las funcionalidades permite la incorporación de una capa de servicios y el desarrollo de un portal a nivel nacional para algunos servicios que brindan los Registros. Como bien enfatiza el tercer punto el desarrollo se puede realizar de forma paralela y posibilitando la especialización de los desarrolladores e incrementando la productividad. Una capa física puede estar compuesta por una o más computadoras, si estas capas se pueden distribuir el rendimiento aumenta y de esta forma el desempeño, durabilidad y escalabilidad del sistema. Por último las pruebas, permitiendo que estas puedan realizarse a nivel de capas, sobre todo al comienzo del desarrollo del sistema cuando todavía no existe una infraestructura de acceso a datos y la estructura de la Base de Datos se está definiendo, permitiendo la sustitución de esta capa y comenzar las pruebas en el resto.

Desventajas

- 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 capas evita el uso de componentes de interfaz de usuario que interactúan directamente con la Base de Datos.
- El uso de capas ayuda a controlar y encapsular la complejidad de aplicaciones grandes, pero agrega complejidad a las aplicaciones simples.

- Cambios en las interfaces de las capas inferiores tienden a propagarse a los altos niveles, especialmente si el modo relajado es usado.

El primer y el cuarto punto constituyen aspectos a tener en cuenta para el desarrollo propio del sistema, más bien constituye una desventaja dependiendo en gran medida del diseño que se le dé. El segundo punto se ve mitigado por la reutilización de componentes que interactúan con clases del modelo, algunos de los cuales fueron desarrollados por proyectos previos en la Universidad o por el uso de la plataforma; en .Net la mayoría de los componentes interactúan con un DataSet¹⁷ que es una entidad separada de los datos. El tercer punto más allá de ser una desventaja se convierte en una ventaja, por ser un sistema complejo y por lo tanto necesita esta encapsulación.

Estos puntos ponen de manifiesto las razones por las cuales se escogió esta arquitectura dentro de las restantes que se estudiaron. Es de destacar que existen otros estilos arquitectónicos además de los enumerados previamente, ejemplo de los cuales son: Estilos de Código Móvil (Arquitecturas de Máquinas Virtuales), Estilos Heterogéneos (Sistemas de Control de Procesos, Arquitecturas Basadas en Atributos y Otros.) Las clasificaciones de estos estilos y arquitecturas son variadas como se puede constatar en el documento (Reynoso, y otros, 2004) pero sus fundamentos permanecen constantes más allá de su clasificación.

Como se mencionó anteriormente para el desarrollo del sistema se reutilizó un Framework desarrollado previamente en la Universidad, a continuación se expondrán algunas de sus características, así como las ventajas y desventajas que este ofrece para el desarrollo del sistema.

2.1.7. Framework.

“En general, un framework es una estructura real o conceptual que pretende servir como soporte o guía para la construcción de algo que expande su estructura en algo usable.

En sistemas computarizados, un framework es con frecuencia una estructura que indica cuales tipos de programas pueden o deben ser construidos y como estos se interrelacionaran. Algunos pueden además incluir programas existentes, interfaces de programación específicas u ofrecer herramientas de programación para usar el framework. Un framework puede ser para un conjunto de funciones dentro de un sistema y como ellos se interrelacionan, las capas de un sistema operativo, de un subsistema de aplicación, como la comunicación debe ser estandarizada en algún nivel de la red de trabajo. Es generalmente más comprensivo que un protocolo y más prescriptivo que una estructura.” (TechTarget)

¹⁷ Dataset: Una réplica de una base de datos en memoria. Al igual que una base de datos puede contener tablas, relaciones y restricciones (Bipin).

Básicamente un framework evita construir una aplicación desde cero, los frameworks orientados a objetos están estructurados en librerías de clases y ahorran de cierta forma el trabajo largo y a veces tedioso de la programación, se debe tener en cuenta que para comenzar a desarrollar en un framework se debe conocer sus especificaciones.

El framework escogido como se mencionó anteriormente fue el desarrollado por un proyecto de la Universidad, el cual tiene elementos del patrón Modelo-Vista-Controlador (MVC) (Trowbridge, y otros, 2003). Estas ideas quedan encapsuladas en una capa de presentación que queda compuesta por los elementos del patrón. El uso del framework no cambia el estilo de arquitectura pues la arquitectura es basada en capas habiéndose empleado o no este framework.

La intención de este patrón es separar estos tres componentes de modo tal que pueda ser variados y desarrollados en paralelo, en el sistema la vista sería el conjunto de formularios de Windows la cual muestra la información necesaria por el usuario, el controlador es el encargado de responder a todos los eventos de usuario, las entradas del teclado y del mouse, informa al modelo de los cambios en la vista y viceversa. El modelo el conjunto de estructuras, clases e interacciones entre estos que son modelados sobre la arquitectura y soportan las reglas del negocio.

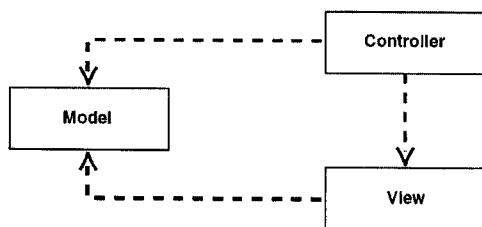


Imagen 3 Modelo-Vista-Controlador

Ahora se analizarán las ventajas y las desventajas de este framework, algunas son propias de del MVC y otras de la implementación en concreta que se le dio.

2.1.8. Ventajas.

Entre las ventajas del Framework escogido están las siguientes:

- **Múltiples vistas usando el mismo modelo:** Dado que la vista 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, et al., 2004) (Dass) (Kotec) Consecuentemente los componentes del modelo de una aplicación son fáciles de implementar, probar y mantener, debido a que todo el acceso al modelo se hace a través de estos componentes. (Dass) En el framework el

controlador y el modelo 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) (Kotec) Importante si alguna vez se decide migrar el sistema a un entorno de 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 la interfaz común es mantenida. (Dass)
- **Distribuir:** Con conjunto de proxys (Gamma, y otros, 1994) se puede distribuir una aplicación MVC, solamente alterando el método de inicio de la aplicación. (Dass)
- **Seguridad a nivel de aplicación:** El framework sienta las bases para brindar seguridad a nivel de aplicación, mediante un fichero de configuración donde se definen roles y en base a estos se gestiona el acceso a las acciones (los controladores) y la visualización de las opciones de la interfaz que le pertenecen a cada rol.
- **Colores de Interfaz Configurable:** se puede cambiar el color de toda la aplicación con unos pocos cambios en la configuración.
- **Configuración del Menú de Opciones:** Mediante un fichero 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 lo conoce, sino que es 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.

2.1.9. Desventajas.

Entre las desventajas del Framework escogido están las siguientes:

- **Complejidad:** introduce nuevos niveles de indirección y por tanto aumenta la complejidad de la solución. Incrementa además la naturaleza basada a eventos del código de la interfaz de usuario, lo cual puede ser más difícil de depurar. (Trowbridge, y otros, 2003)
- **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 cambios del código pueden ser difíciles. (2006)
- **Interfaz:** el framework posee una interfaz al estilo Web donde en la parte derecha se muestra un 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 no es posible.

- **Interoperabilidad con Bloques de Aplicación (Applications Blocks):** debido a sus características únicas no se puede utilizar el Bloque de Aplicación para Procesos de Interfaz de Usuario (User Interface Process) que se integra con los demás Bloques de Aplicación como un paquete a través de la Librería para Empresas (Enterprise Library Application Block) (Active Software Professional Alliance) en la versión para el .NET Framework 1.1 y en caso de una migración para el .Net Framework 2.0 y 3.0

2.1.10. Mejoras Propuestas.

Las mejoras que se le incluyeron al framework son las siguientes.

- **Manejador de excepciones:** incluir dentro del framework una forma de manejar las excepciones que se detecten por la aplicación de manera tal que se pueda configurar el lugar donde son guardadas, base de datos o fichero en formato XML.
- **Versionamiento:** Controla la versión del framework que se está utilizando, lo cual mejora además la seguridad de la aplicación.
- **Flujo de Navegación:** incluir una infraestructura que permita definir flujos de navegación, de forma similar al Bloque de Aplicación Interfaz de Usuario (User Interface Process Application Block). En el marco del proyecto se desarrolló un componente para estos flujos de trabajo lo cual se analizará en el siguiente epígrafe.

2.1.11. Flujos de Trabajo.

A menudo las aplicaciones que automatizan los procesos empresariales o de servicios, están compuestas por flujos de trabajo, estos flujos están compuestos por una serie de vistas que muestran y capturan información del usuario, registrándose al final del proceso los cambios como una transacción atómica.

En el sistema existen varios de estos flujos de trabajo, ejemplo: Presentación, Revisión Legal (Véase **Error! Reference source not found. Error! Reference source not found.**) En su mayoría los pasos del proceso de inscripción son flujos que deberán capturar información y registrarla como un todo.

Entre las **ventajas** que reporta el uso de este componente están las siguientes:

- **Navegación Configurable:** permite cambiar el modo en el cual se hace la navegación y por consiguiente el orden en el cual se muestra la información. En numerosas ocasiones los clientes desean que se siga mostrando la misma información pero solamente variando el orden en el que se muestra o se capturará esta.

- **Extensible:** es fácil agregar nuevas vistas y sus controladores, una vez programado el controlador y diseñada la vista solo es necesario incluirlos en la configuración para que se comience a interactuar con estos.
- **Información Centralizada del Flujo:** cada flujo del navegador posee un estado, el cual tiene toda la información que se cambió o se insertó durante el flujo, de esta forma mirando el estado se conoce que información deberá ser actualizada en la base de datos.

2.1.12. Estándar de codificación.

El estándar de codificación fue definido por el proyecto, se define el formato para los siguientes puntos:

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

Las buenas prácticas de programación y la uniformidad en el código permiten que un programa sea legible y que parezca que fue escrito por un solo programador.

Una vez visto y explicado las bases sobre las cuales se construyó el sistema se presentarán las soluciones que por su importancia y flexibilidad son destacan en el desarrollo del sistema.

2.2. Soluciones.

2.2.1. Algoritmo de búsqueda fonética.

Para la búsqueda fonética las denominaciones solicitadas se deben buscar en las denominaciones reservadas o solicitadas previamente en un lapso de 3 días. Existen dos modos de búsqueda, Texto Completo o Separado por Palabras.

El modo Texto Completo se refiere a buscar exactamente la frase o palabra entrada en las denominaciones que ya estén solicitadas o reservadas.

Ejemplo:

Utilizando como criterio de búsqueda la frase AERO PUERTO se mostraría como coincidencia:

- **AEROPUERTO INTERNACIONAL DEL CENTRO**
- **LIMPIADORA Y SERVICIOS AEROPUERTOS**

Utilizando como criterio de búsqueda la frase CA LO SA:

- **SONIDOS ERCALOZA SCALA**
- **TÉCNICA LOZANO**
- **INVERSIONES CALOZAN**

Por último, utilizando como criterio de búsqueda la frase PROYECTOS TORONCA traería como coincidencia:

- **PROYECTOS TORON CA**

La decisión de si la denominación será aceptada o denegada deberá ser tomada por el funcionario, el sistema mostrará los elementos de coincidencia que soportarán la toma de decisiones.

El otro criterio de búsqueda es Separar Palabras, se basa en el mismo principio de búsqueda pero en este caso se buscarían cualquiera de las palabras contenidas en la frase.

El algoritmo escogido para la implementación es el Soundex (Véase 1.5.2 Algoritmos) los códigos fonéticos en este caso se adaptan a las reglas del idioma español. Algunas de las transformaciones que se hacen son las siguientes:

- cambio de C por S para los fonemas CE, CI que suenan como SE, SI.
- cambio de Q por K para sonidos QUI, QUE
- cambio de U por W para los sonidos UA, UE por WA, WE

El objetivo de estas transformaciones es convertir los fonemas iguales a la misma cadena para una vez hecho esto, hacer las comparaciones sobre estas, como se puede observar su basamento es el Soundex, solo que las transformaciones no se hacen a dígitos, sino a letras.

El conjunto de transformaciones que se le aplican a la palabra o frase son las siguientes:

- Se lleva la frase a mayúscula.
- Se eliminan las vocales con tildes transformándolas a vocales comunes.
- Eliminación de la H cuando suena muda.
- Se aplican las reglas de transformación del idioma.
- Se eliminan los espacios.

Como resultado se obtiene una cadena, la cual se utilizará para todas las futuras comparaciones.

A continuación se analizará otras de las soluciones más importantes, el cálculo de los aranceles o impuestos.

2.2.2. Cálculo.

El cálculo de los aranceles o impuestos deberá ser una solución muy flexible, partiendo de que están basados y regulados por la Ley de Registros y Notarías (Asamblea Nacional de la República Bolivariana de Venezuela, 22 de Diciembre de 2006) Esta ley puede cambiar según lo estimen conveniente los órganos pertinentes, además la organización puede hacer ajustes en estos aranceles, la misma estipula que la Unidad Tributaria, la base de todos los cálculos deberá incrementarse anualmente lo que significa que estas variaciones no pueden implicar cambios en la implementación.

Estos impuestos son entregados a diferentes tributarios, existiendo dos en estos momentos, Servicio Autónomo que es la entidad que representa a la Dirección de Registros y Notarías y a todos los Registros en sí, así como el SENIAT, que es el órgano que regula las tasas que pertenecen al gobierno de los distintos estados y el central. La existencia de estos tributarios bien definidos no implica que estos no puedan variar sus nombres, esto puede ocurrir por el cambio de alguna Ley o incluso la creación de alguno nuevo, en todos estos casos el cálculo de impuestos en el sistema debería asumir esos cambios.

Las expresiones de cálculo de impuestos pueden estar definidas por expresiones aritméticas, condicionales y puede que determinados aranceles se calculen a través del cálculo de otros.

Teniendo en cuenta estas características se plantearon las siguientes ideas:

- El cálculo estaría definido por conceptos de pago, estos conceptos serían entidades calculables o editables, o sea se podrían calcular en base a alguna expresión o ser valores de datos que introduciría el usuario que servirían para realizar determinados cálculos. Ejemplo de conceptos calculables serían los diferentes artículos de la Ley (Artículo 86 Numeral 1. Por la inscripción de cualquier tipo de sociedades, firmas personales, y asociaciones de cuentas en participación, dos unidades tributarias (2 U.T.), más una décima de unidad tributaria (0,1 U.T.) por cada folio que contenga el documento o actuación. Expresión $2*UT+0.1UT*folios$). En este caso folios y UT serían dos conceptos, el primero editable y el segundo solo editable desde el Servicio Autónomo.
- Se construiría un componente, que permitiera mostrar estos conceptos de forma dinámica, de esta forma si agregara algún concepto no sería necesario cambiar interfaces ni reglas del negocio.
- Los conceptos tendrían atributos que brindarían la información de cómo deberían ser mostrados en este componente, incluyéndose reglas de validación. Ejemplos de estos atributos serían: editable o no, visible, mostrable en alguna planilla de salida,

variable (la cual se utilizaría en las fórmulas para el cálculo de otros conceptos) y nombre, el cual sería el que identificaría el concepto en la interfaz.

- Los tributarios estarían compuestos por conceptos, de esta forma la suma de todos los conceptos que tributen al monto final a pagar serían el total a pagar.
- Los tributarios podrán contener exoneraciones de pago, que son documentos emanados desde el Servicio Autónomo y que implican el no pago de estos derechos en base a algún tributario.
- Todo debería encapsularse de tal forma que pueda ser reutilizado para una capa de servicios.

Así funcionaría el sistema, haciendo configurables estos conceptos y que las modificaciones en la base de datos por el Servicio Autónomo impliquen las correspondientes modificaciones en los Registros y por consiguiente una uniformidad en cuanto al modo de cálculo y el control por parte de esta entidad. De esta forma se garantiza la seguridad jurídica y se facilitan las auditorías, toda la información referente a estos temas estaría en la base de datos central, permitiendo ser consultada de forma oportuna y eficiente.

Una vez analizado el cálculo se expondrá otra de las soluciones dadas en el sistema, la búsqueda de prohibiciones.

2.2.3. Búsqueda de Prohibiciones.

La búsqueda de prohibiciones como se menciona anteriormente (Véase **Error! Reference source not found. Error! Reference source not found.**) es una revisión sobre los documentos que fueron emanados de algún tribunal. Estos documentos son enviados al Servicio Autónomo para que sean divulgados a todos los Registros del territorio Venezolano, siendo la distribución de forma automática en el sistema.

Para la búsqueda de prohibiciones de personas jurídicas se aplica la búsqueda fonética de compañías. El funcionario determina por el nombre si la prohibición afecta a la compañía de la cual se está efectuando la búsqueda; pero para el caso de las personas que pueden estar afectadas por estos documentos se implementó una solución diferente.

El principal problema que debe enfrentar esta búsqueda, es que los documentos que llegan al Servicio Autónomo pueden estar incompletos en cuanto a datos, que la información que contengan no sea suficiente para identificar una persona natural o que aún no se encuentre en el sistema porque aún no ha sido actualizada la base de datos central. Por estas razones no se puede seleccionar una determinada persona para asociarla a una prohibición, sino que se deben entrar los datos referentes a estas y después en la búsqueda de prohibiciones hacerla por los atributos que sean similares. Estos atributos pueden estar presentes o no en el documento.

El algoritmo consiste en asignarle a todos los atributos por los que se puede realizar búsqueda un determinado peso en dependencia de lo que aporte como coincidencia, de esta forma el número de cédula (documento que identifica a las personas naturales venezolanas) tendría el mayor peso, mientras que el segundo nombre de una persona tendría poco peso. El peso a partir del cual se considera una ocurrencia como coincidencia (umbral), devolviéndose solo las ocurrencias superiores a este, sería configurable, al igual que el resto de los pesos de los atributos.

Estas son algunas de las soluciones que más se destacan en el desarrollo del software, ahora explicaremos los patrones de diseño más aplicados.

2.3. Patrones de diseño aplicados.

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

El **Patrón Fachada** se usó 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. La intención del patrón fachada es proveer una interfaz unificada para un conjunto de interfaces en el subsistema, haciendo a este fácil de usar. (Gamma, y otros, 1994) Las fachadas se ven con más facilidad para el traspaso de datos y la encapsulación a través de las capas.

El **Patrón Mediador** se utilizó para la transformación de los datos que se devuelven como estructuras de datos desde la base de datos (generadas por el Tier Developer) a las Entidades del Negocio que son con las cuales interactúan las capas superiores. Su intención es encapsular la interacción entre objetos, promoviendo el bajo acoplamiento de manera tal que estos puedan variar independientemente (Gamma, y otros, 1994) De esta forma las entidades que generara el Tier Developer podrían variar sin afectar a las clases del modelo.

El **Patrón Método Plantilla** es uno de los más utilizados para la reutilización de una misma vista por uno o más controladores (acciones según las define el framework). Su intención es definir el esqueleto de un algoritmo dejando las especificidades a las subclases y permitiendo que estas redefinan ciertos pasos del algoritmo. (Gamma, y otros, 1994) De esta forma el controlador base podría definir el comportamiento para las operaciones comunes y dejaría a las subclases que respondieran con sus propias implementaciones a las que deberían realizar.

De manera general estos fueron los patrones más utilizados, pero ello no excluye que se utilizaran patrones como, Estrategia, Métodos de Fábricas y Estado.

Una vez visto estos aspectos se define el capítulo 3 la solución de diseño para la realización del sistema donde se podrán observar expresados estos patrones y las soluciones de diseño a todo lo que se ha expuesto en este capítulo.

Capítulo 3

El diseño de sistemas se ocupa de desarrollar las directrices propuestas durante el análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados tanto desde el punto de vista funcional como del no funcional. El proceso de diseño de un sistema complejo se suele realizar de forma descendente:

- Diseño de alto nivel (o 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 los objetivos establecidos en el análisis.
- Desarrollo según la especificación.
- Prueba.
- Integración de todos los subsistemas.
- Validación del diseño.

En el transcurso de este capítulo se expondrán una serie de diagramas pertenecientes al diseño del sistema.

3.1. Modelo del Diseño

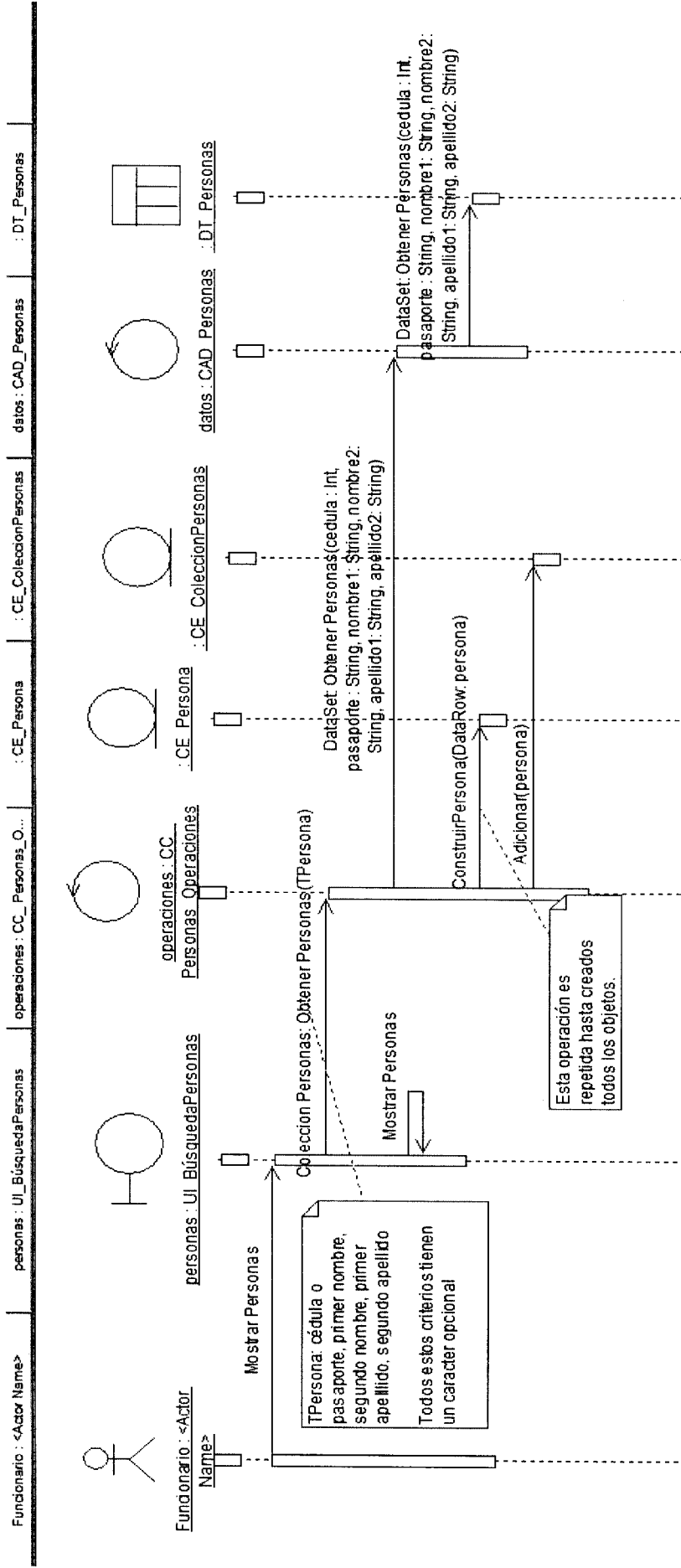
Como parte del Modelo de Diseño del proceso de Inscripción de este módulo de software, se realizaron los diagramas de interacción de los flujos de operaciones por pasos de este proceso, representados con el Diagrama de Interacción. También se encuentran los Diagramas de Clases y el Modelo de Datos pertenecientes a este proceso. Tanto los diagramas de clases como los de interacción fueron agrupados por grupos, estos grupos son: Búsquedas, Denominación Mercantil, Presentación y Revisión Legal.

3.1.1. Diagramas de Interacción

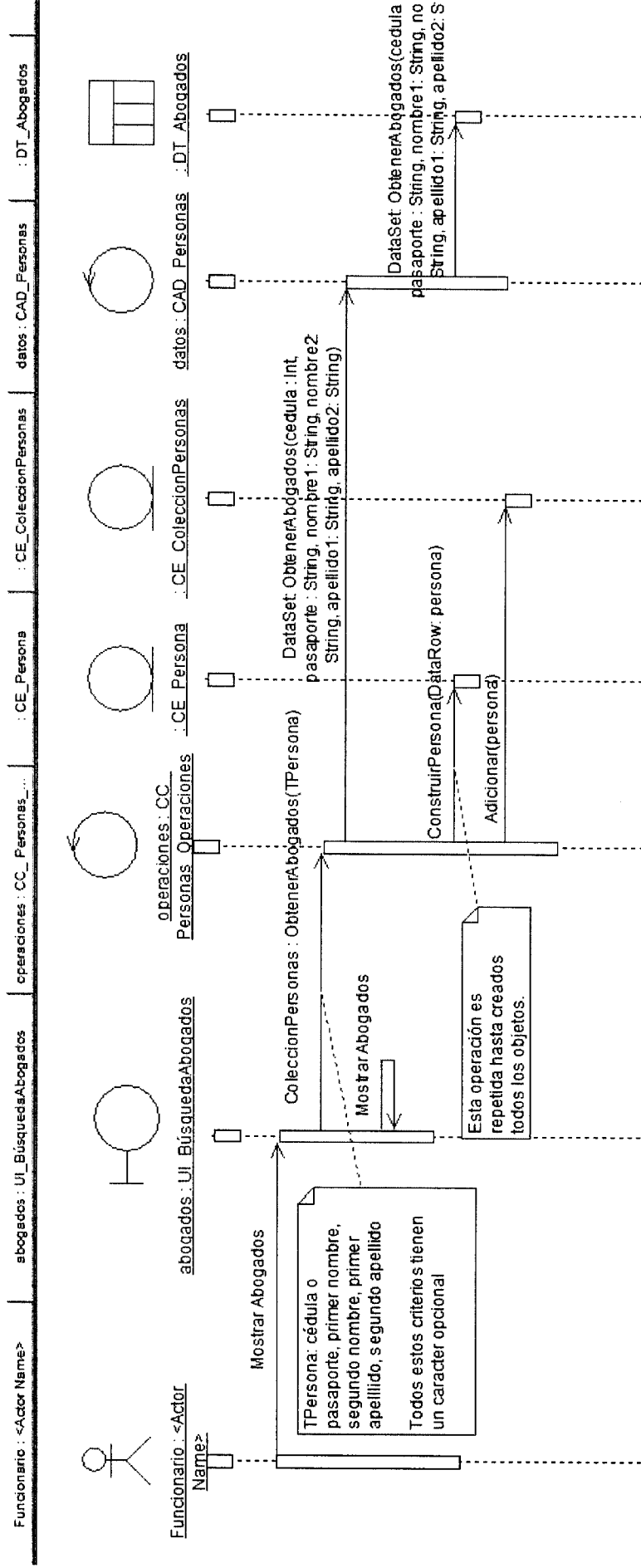
A continuación se describen los diagramas de interacción, en este caso se representan mediante el Diagrama de Secuencia. Cada sub epígrafe de los que a continuación se muestran contiene el modelo de cada uno de los grupos antes mencionados.

3.1.2. Búsquedas

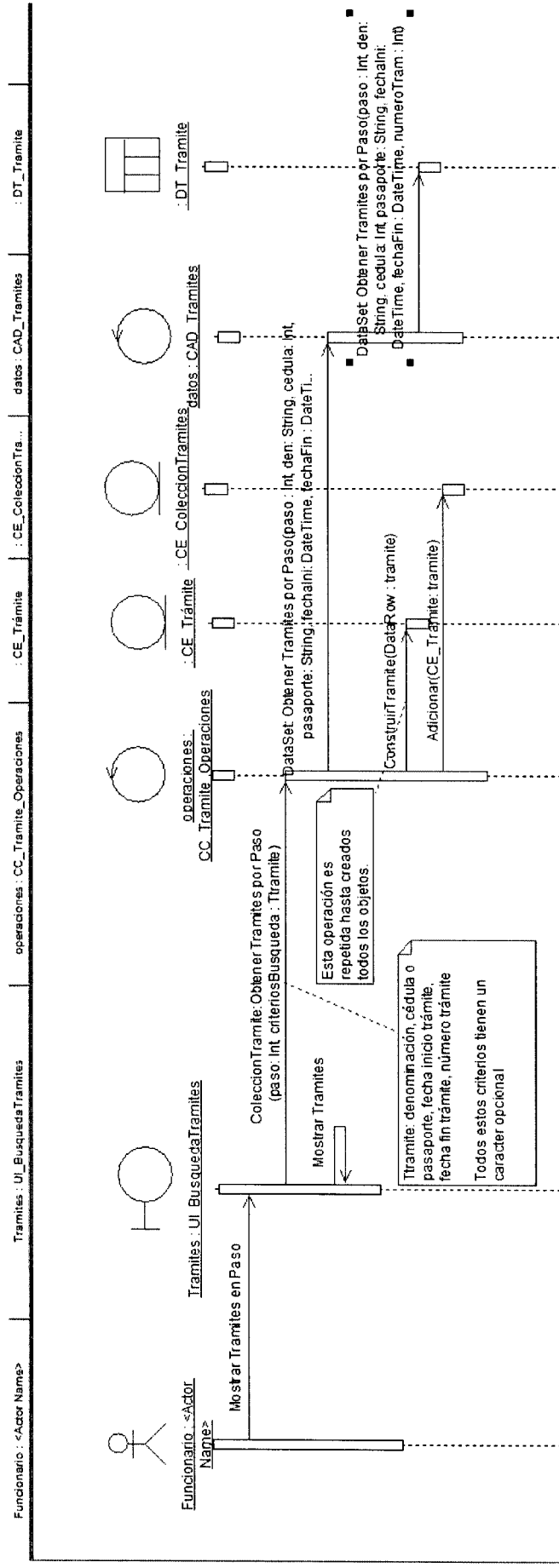
Búsqueda de Personas. Este diagrama es de carácter general y es usado por otros diagramas.



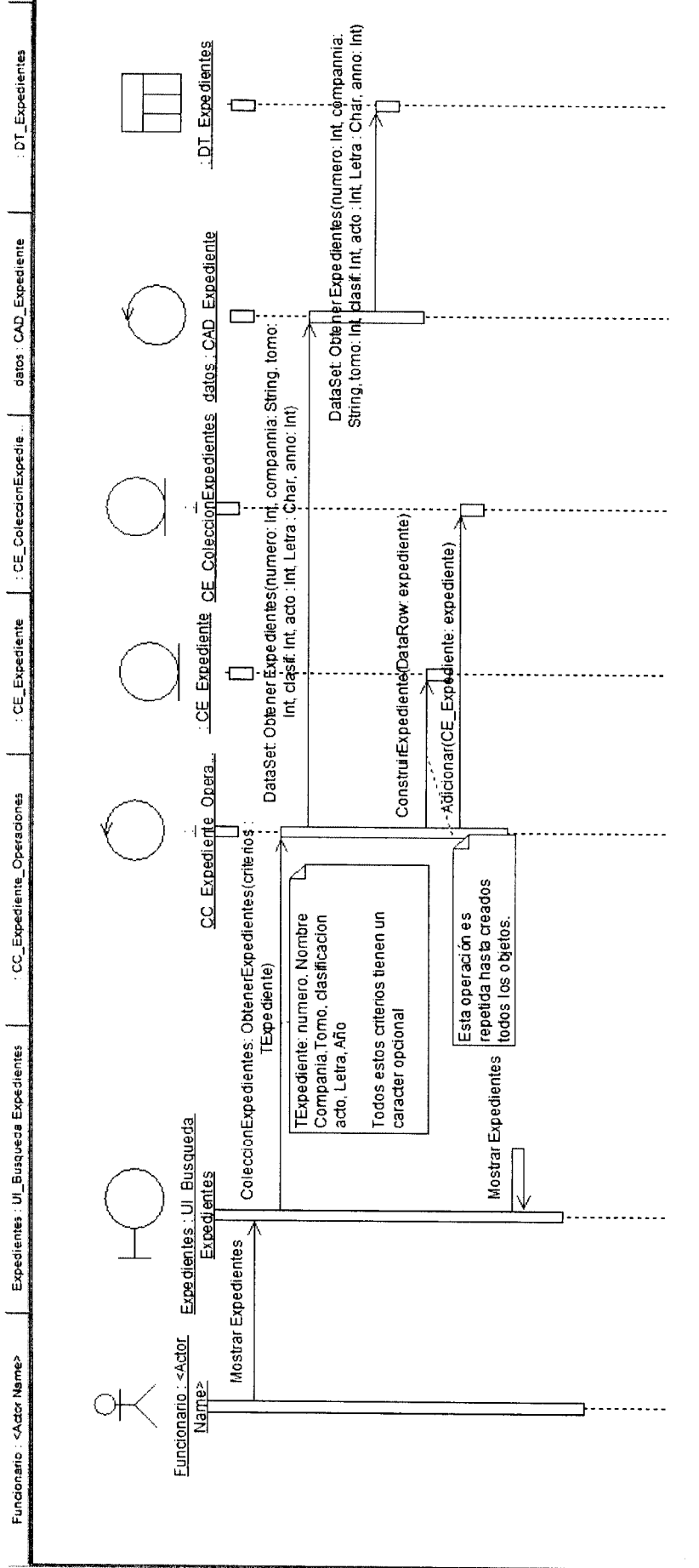
Búsqueda de Abogados. Este diagrama es usado para la búsqueda de abogados redactores, los encargados de redactar los documentos presentados.



Búsqueda de Trámites en Paso. Este diagrama es usado para la búsqueda de los trámites pendientes en cada uno de los pasos por los que pasa el trámite, dígame pasos los flujos de operaciones relacionadas a un sub proceso asociado al documento presentado.



Búsqueda de Expedientes. Este diagrama es usado para la búsqueda de los expedientes con los que cuenta el Registro, estos expedientes identifican cada una de las compañías inscritas.



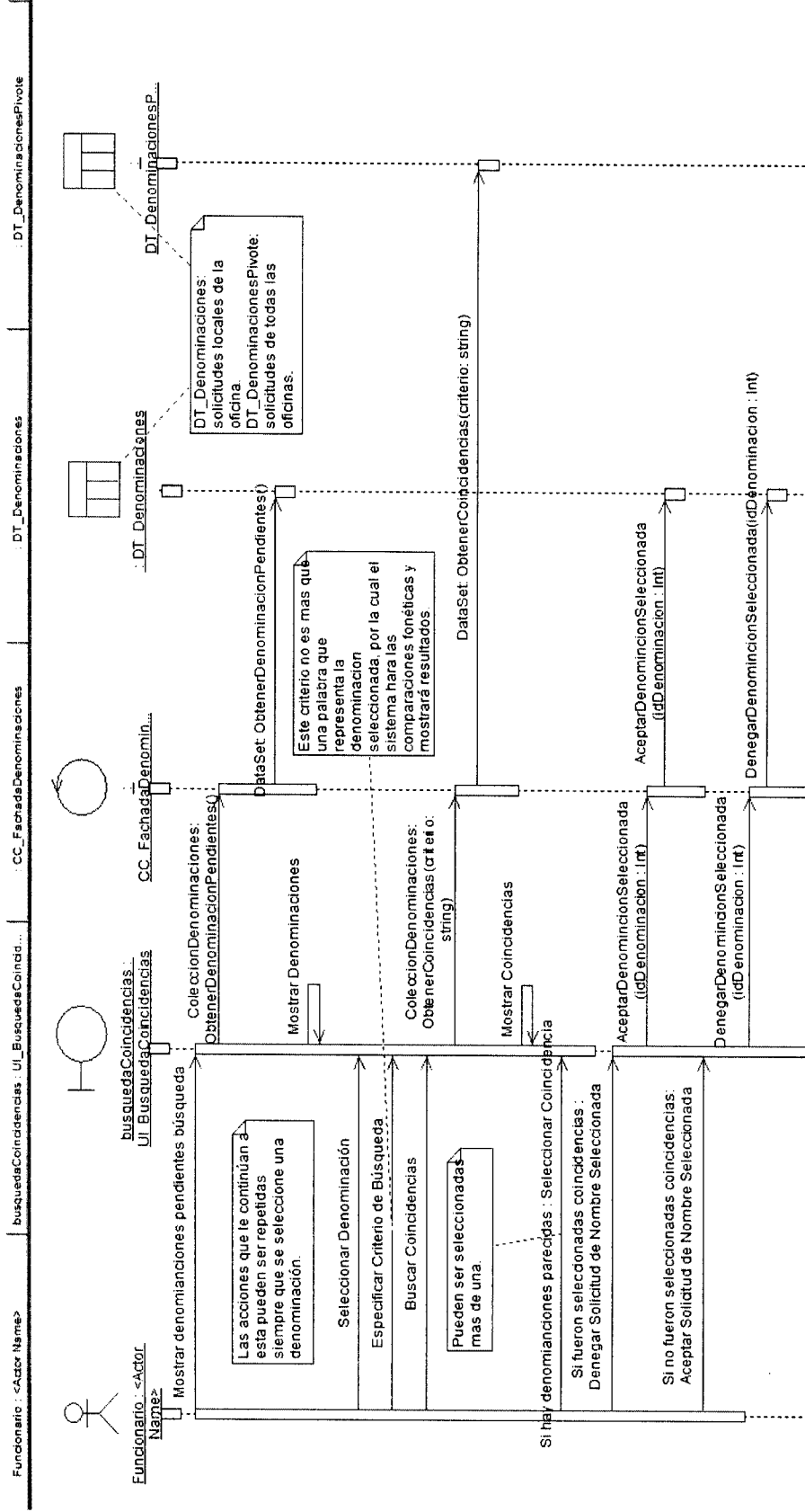
3.1.3. Cálculo

El cálculo de gastos es uno de los pasos más importantes del proceso de inscripción. Este diagrama esta descrito en el **Anexo # 4**.

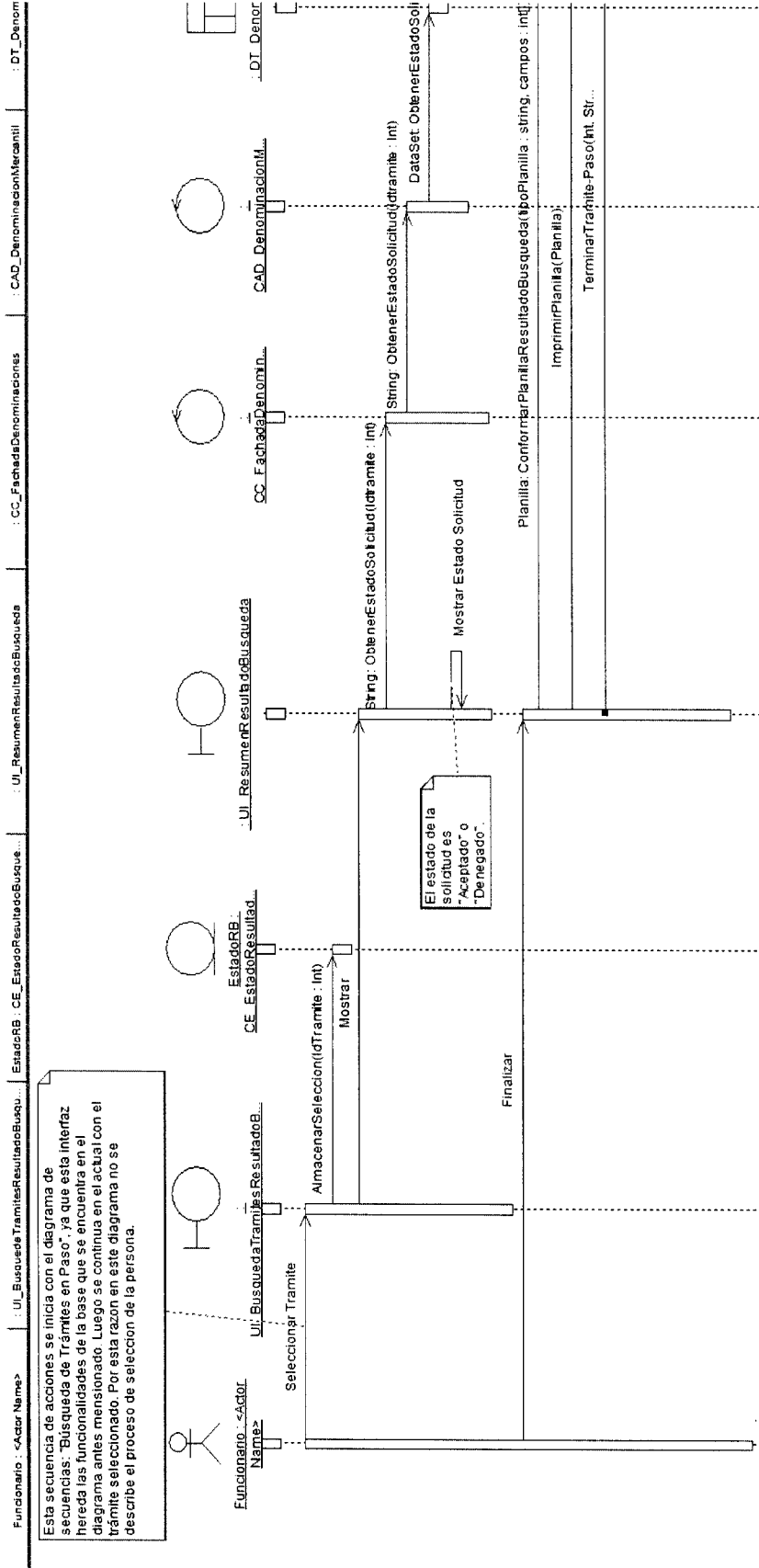
3.1.4. Denominación Mercantil

Solicitud de Nombre. Este diagrama describe las primeras operaciones que se llevan a cabo cuando se desea constituir un trámite. Es el flujo donde se selecciona por parte del usuario interesado un nombre que representara legalmente su empresa. Véase el **Anexo # 5** para consulta el diagrama.

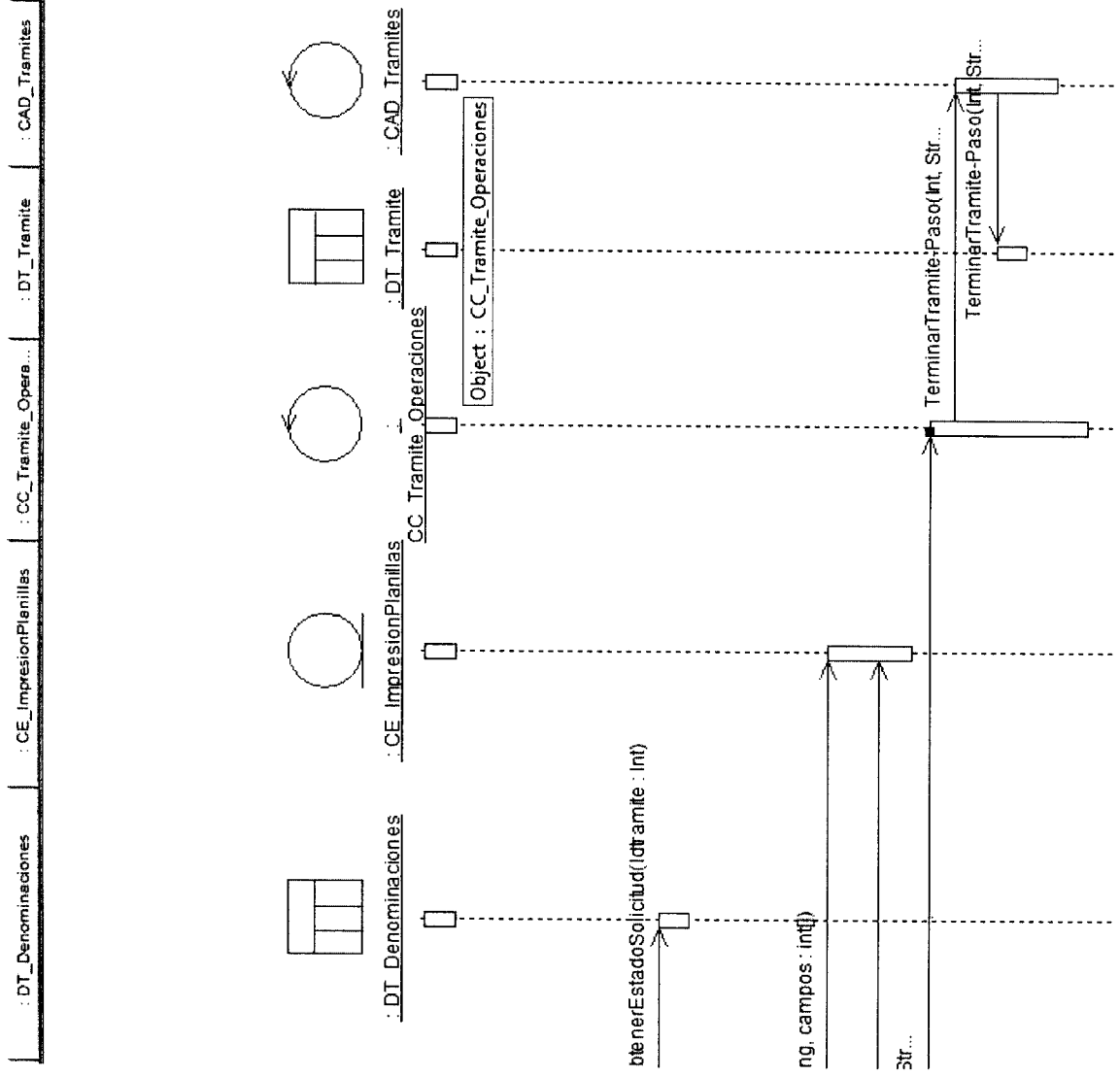
Chequeo de Coincidencias. El diagrama siguiente representa el proceso de aceptación ó denegación que lleva acabo el funcionario, con relación al nombre solicitado por el usuario.



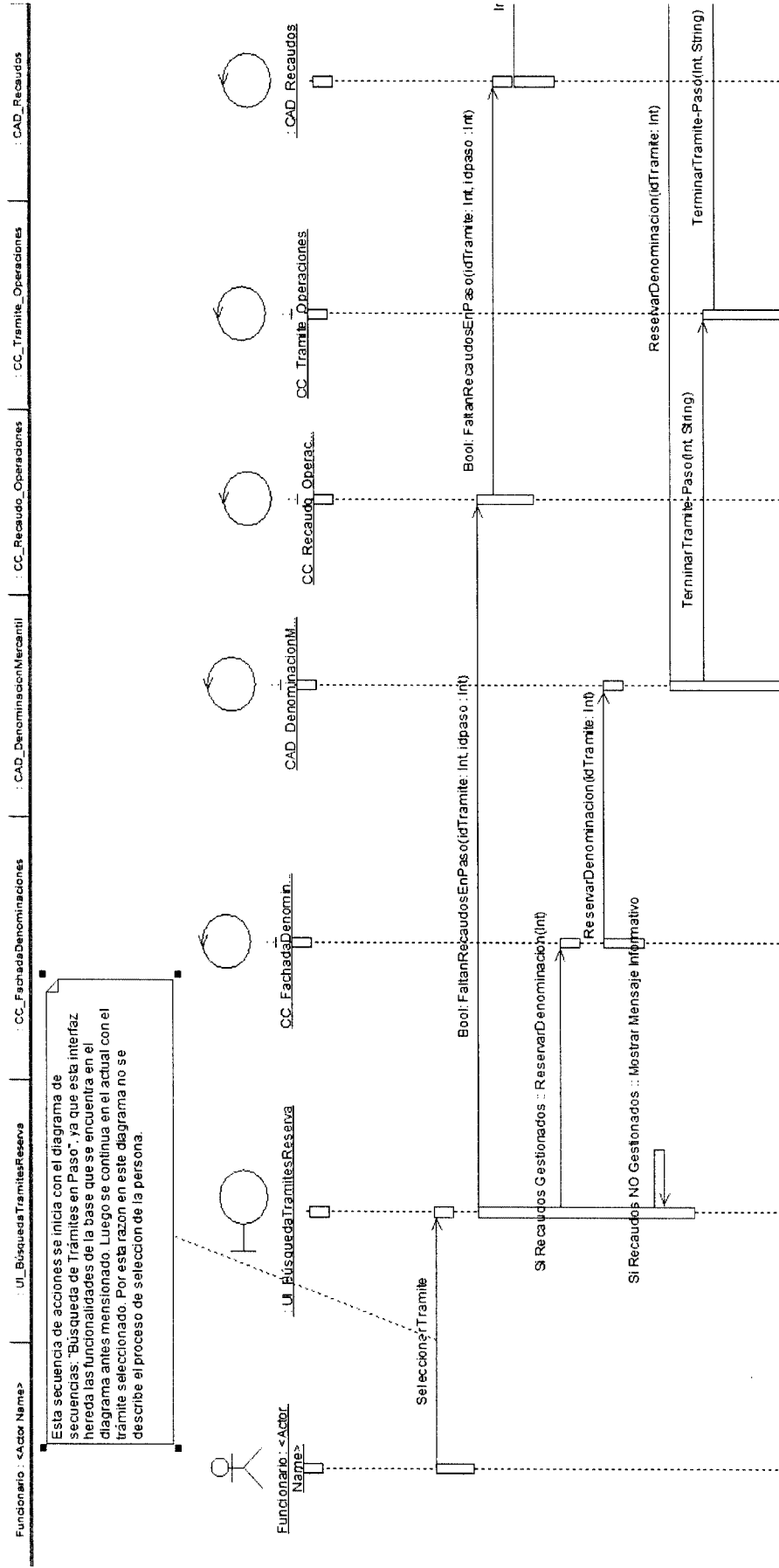
Resultado de Búsqueda. Estas operaciones se realizan después del usuario haber solicitado la denominación, aceptada ó denegada la misma por parte del funcionario y pagados los derechos de solicitud por parte del usuario. Se le entrega una planilla con informándole el resultado de la misma.



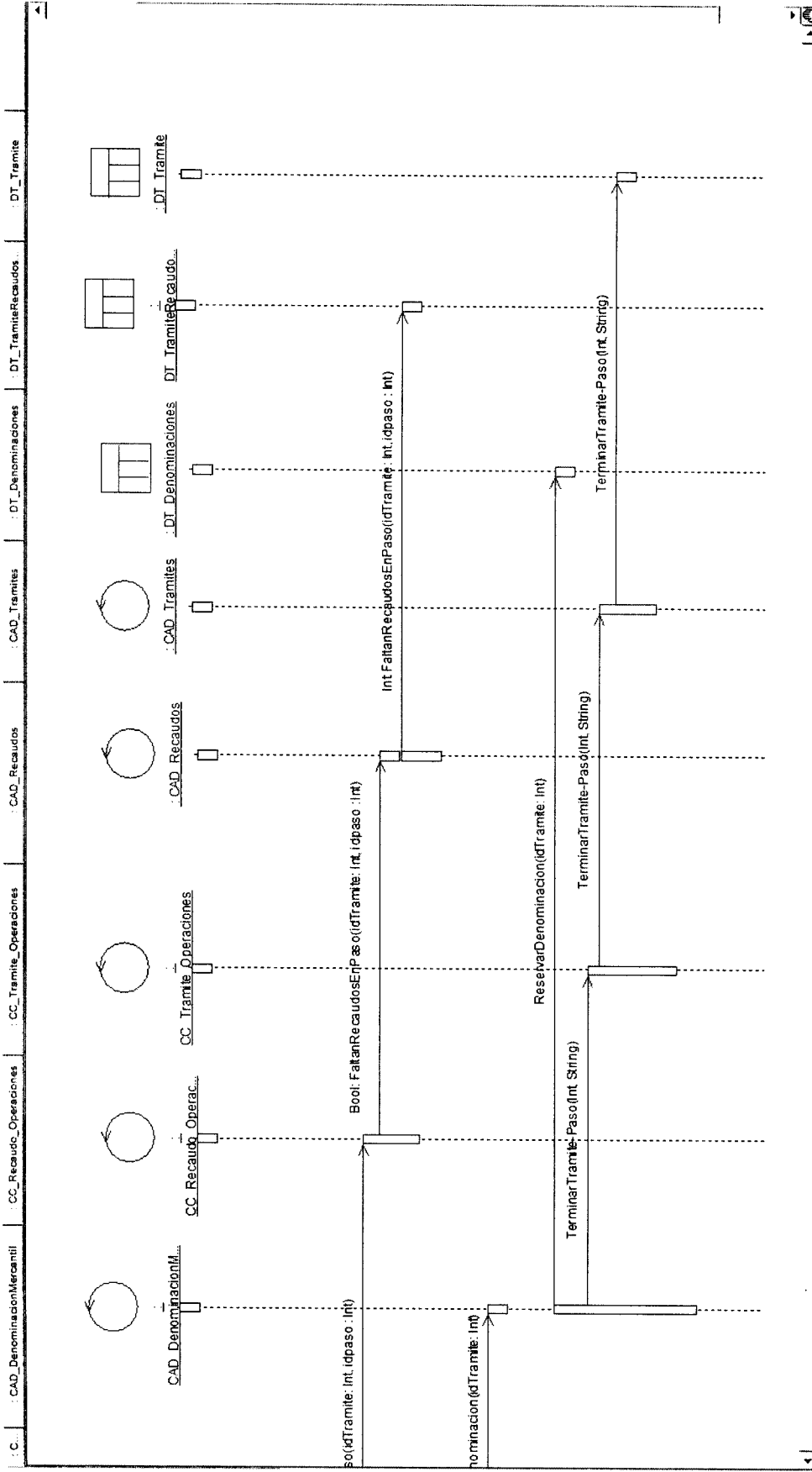
Diseño e Implementación



Reserva de Nombre. Es este el flujo que finaliza los procesos de Denominación Mercantil, dando como reservado el nombre solicitado y trasladando el trámite al paso de presentación que se realizará posteriormente.



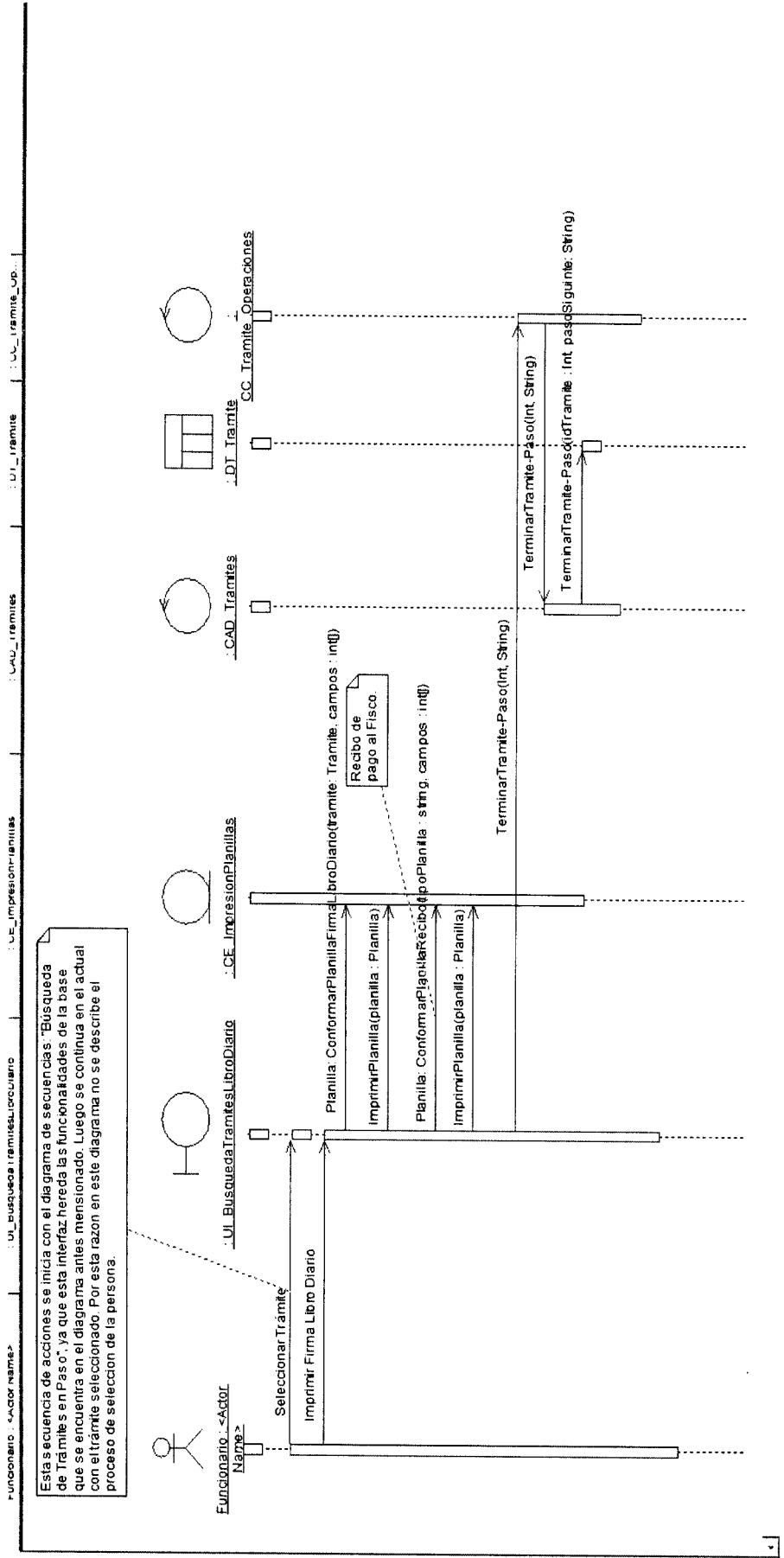
Diseño e Implementación



3.1.5. Presentación

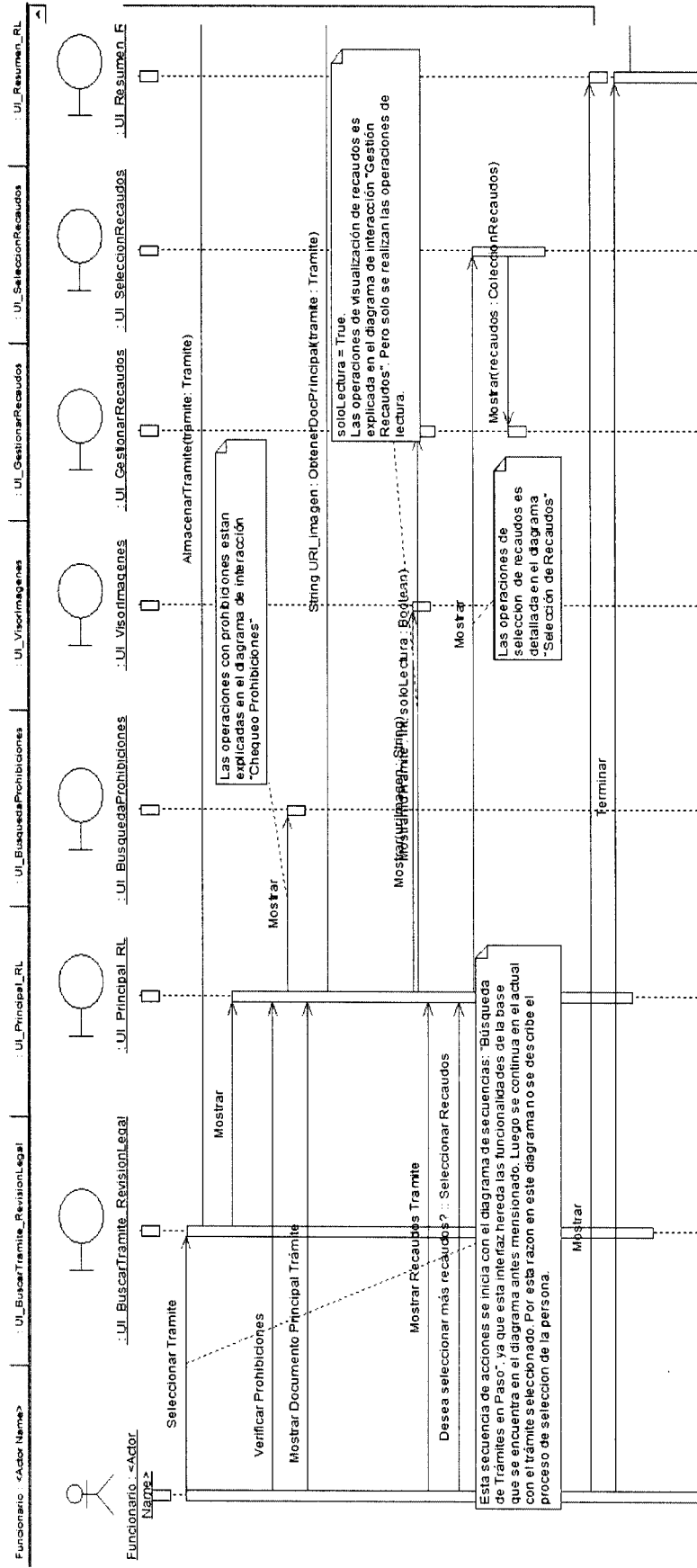
Presentación de documentos. Describe el flujo de presentación de los documentos por parte del usuario. Aquí se recogen los datos del presentante, otorgantes, conceptos por los cuales el cálculo elaborará el monto a pagar por tributarios, selección y gestión de los recaudos que se necesitan para llevar a cabo el trámite, entre otros. Véase el **Anexo # 6**.

Firma del Libro Diario de Presentación. Es aquí donde se recoge la firma del presentante y donde se registra finalmente en el sistema la presentación de los documentos. A continuación le sigue el proceso de revisión legal de documentos.

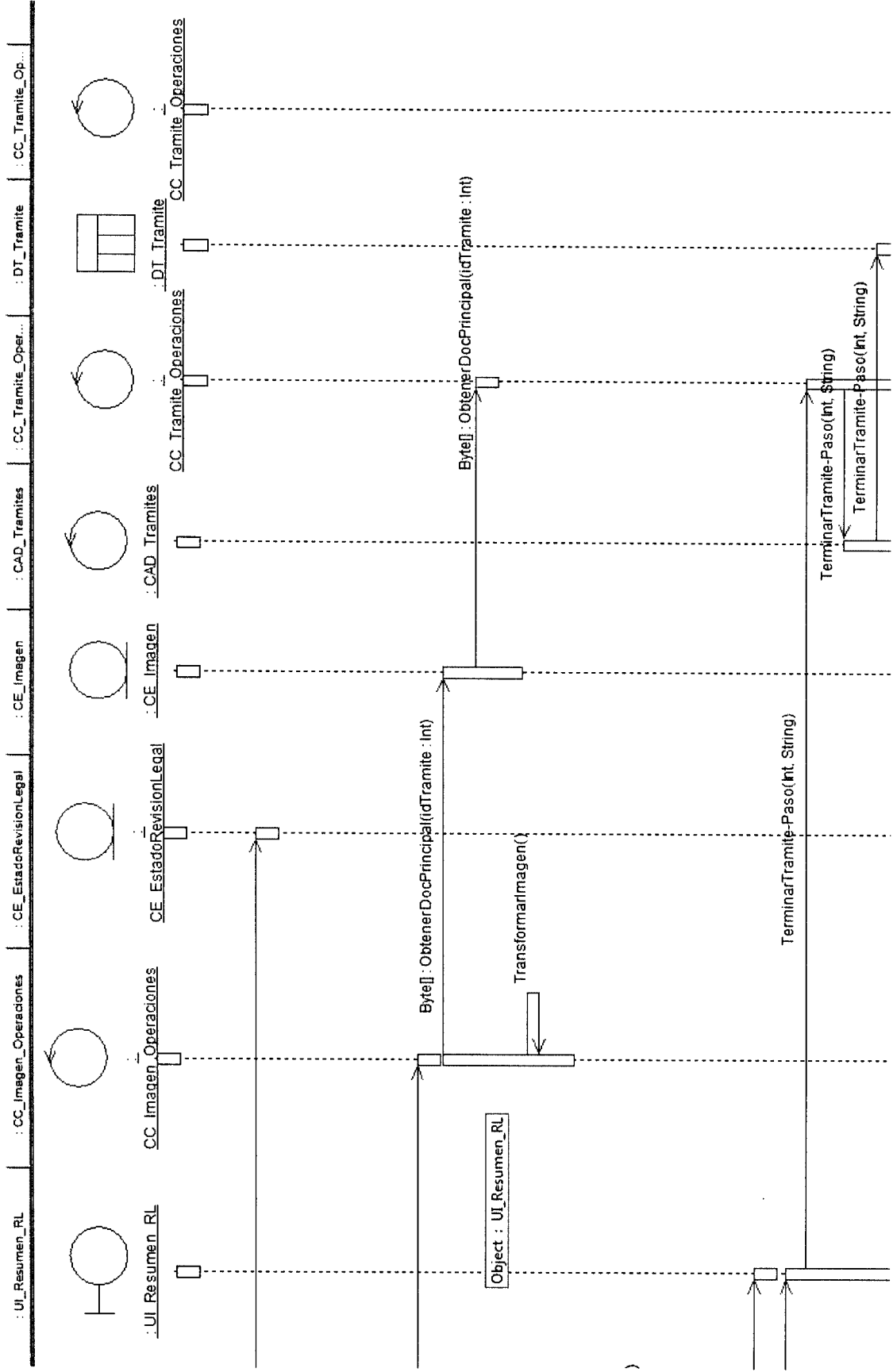


3.1.6. Revisión Legal

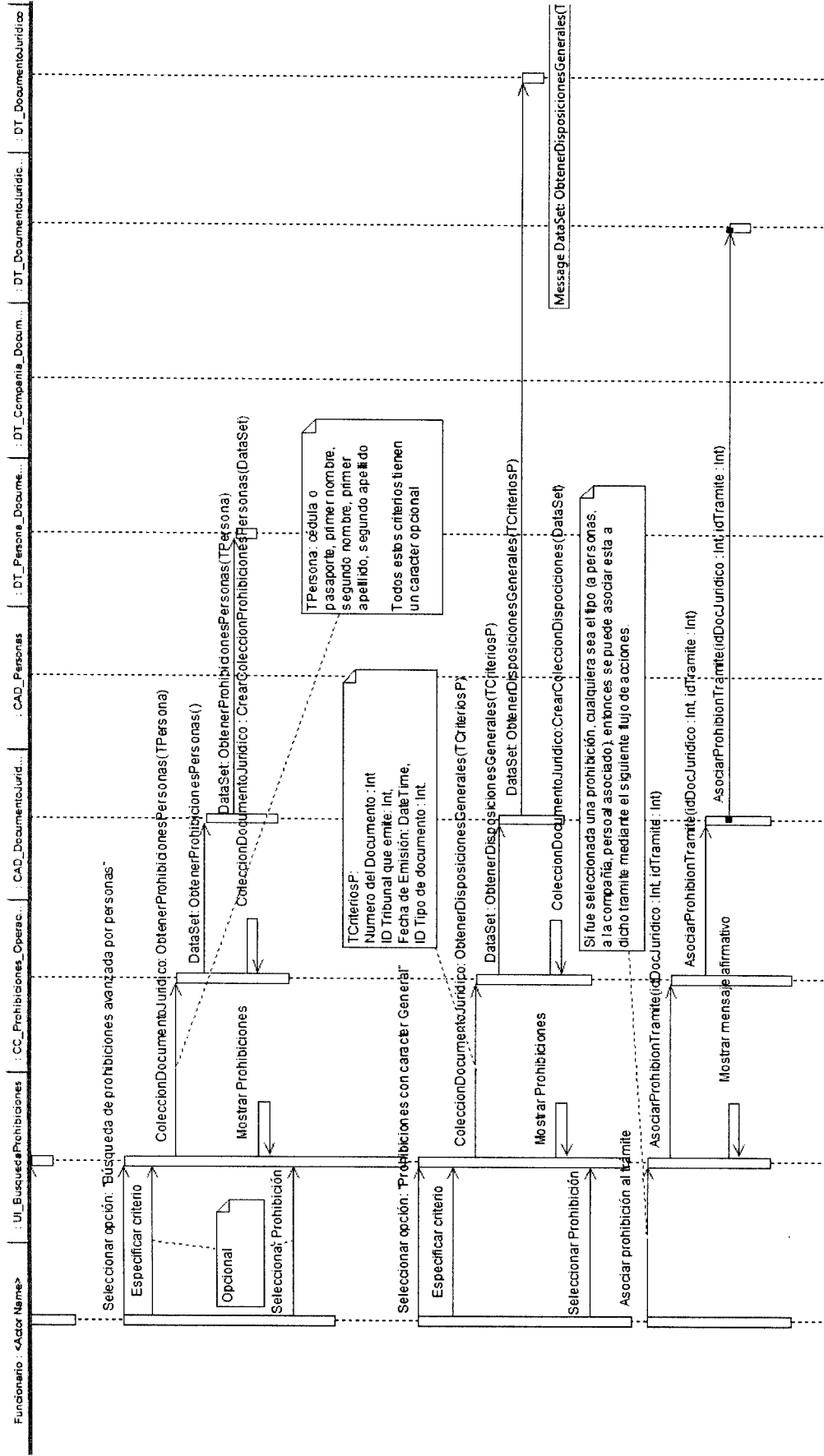
Revisión Legal de Documentos. Este diagrama describe las operaciones realizadas sobre el trámite, por parte del funcionario en el paso de Revisión Legal. Allí se consulta y revisa el documento principal del trámite, los recaudos asociados y la correcta gestión de los mismos, las prohibiciones en caso que alguna se relacione al trámite.



Diseño e Implementación

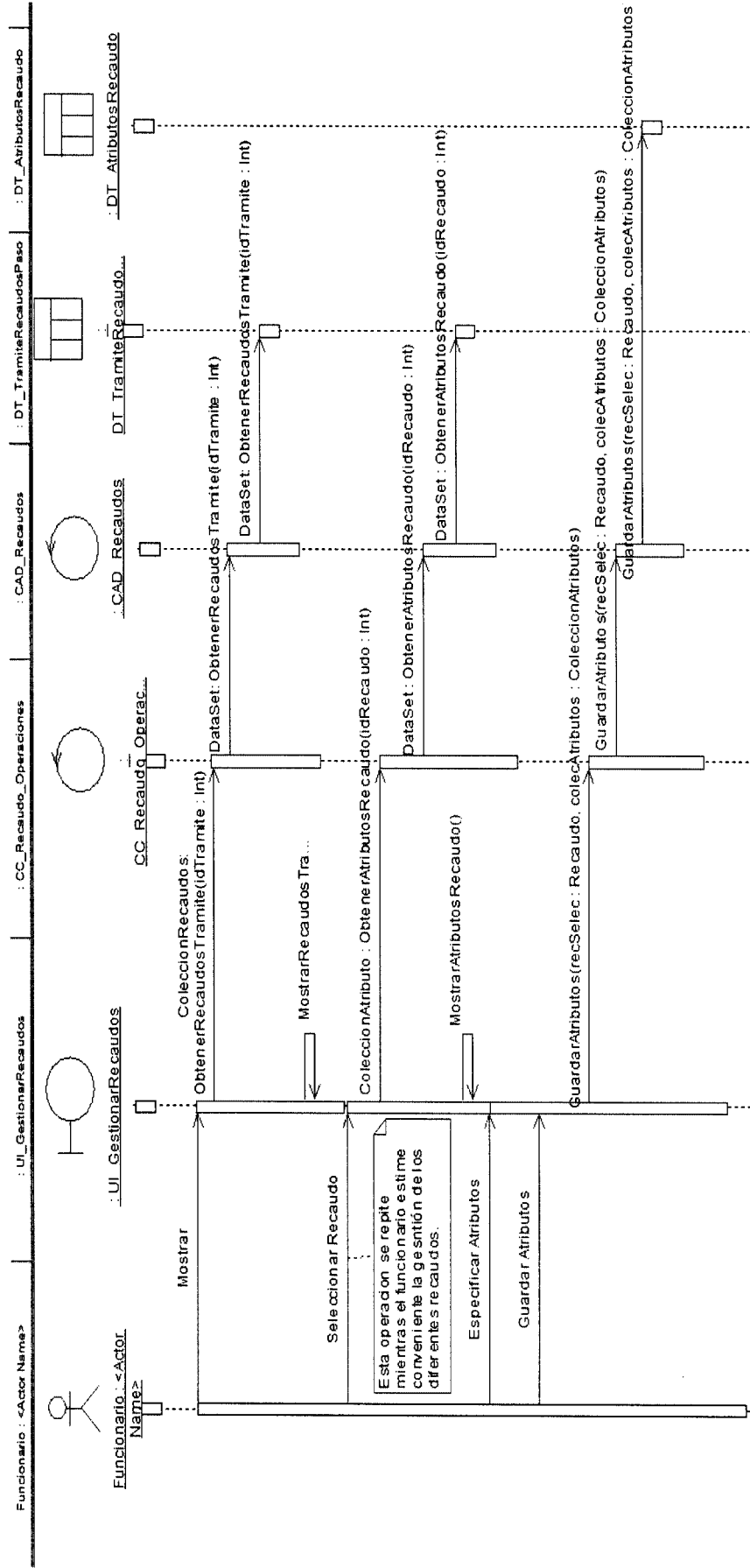


Diseño e Implementación



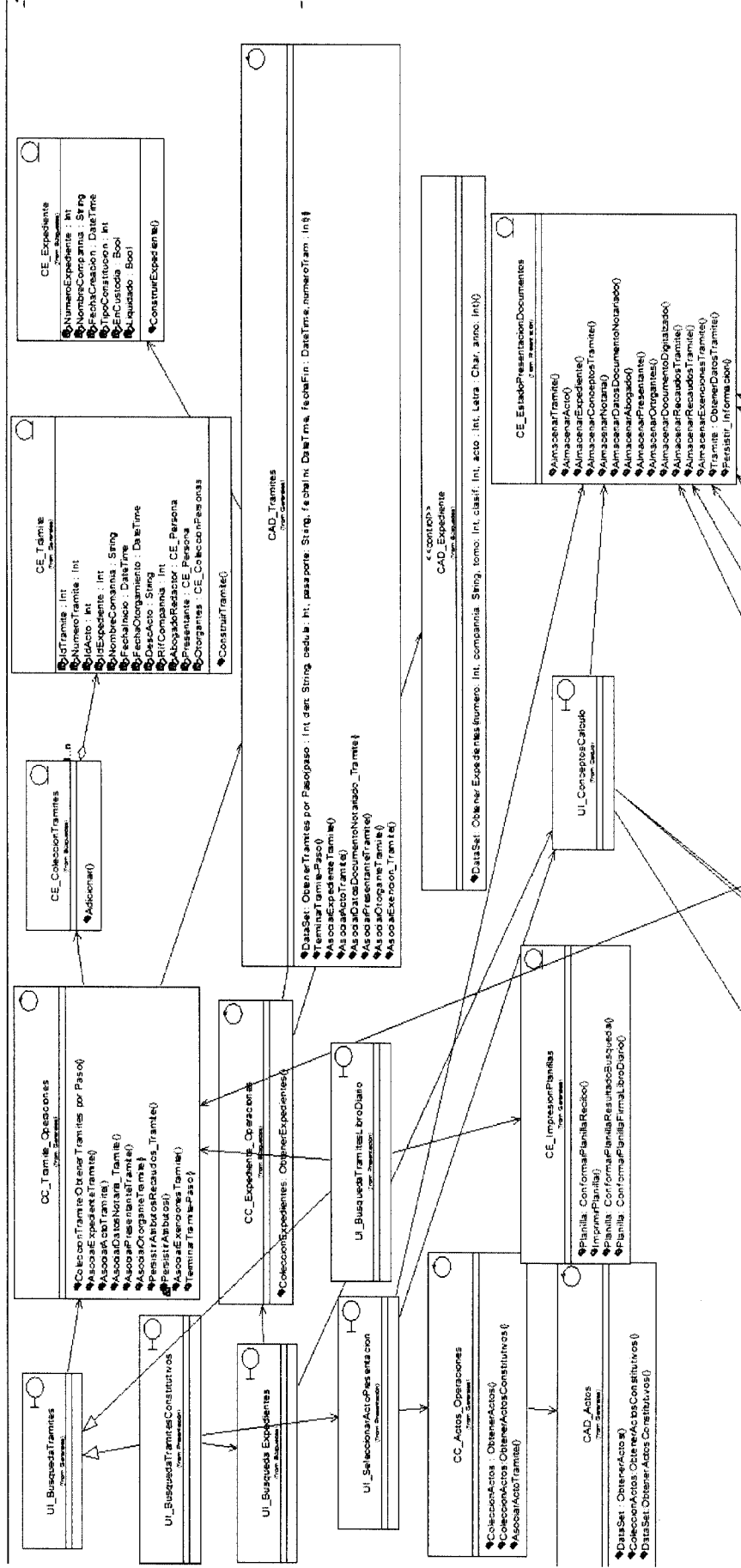
3.1.7. Gestión Documental

Este diagrama representa las operaciones que se llevan a cabo para la gestión de los recaudos asociados al trámite, dígame especificación de los atributos (datos referentes a cada recaudo presentado) como la digitalización de los mismos.

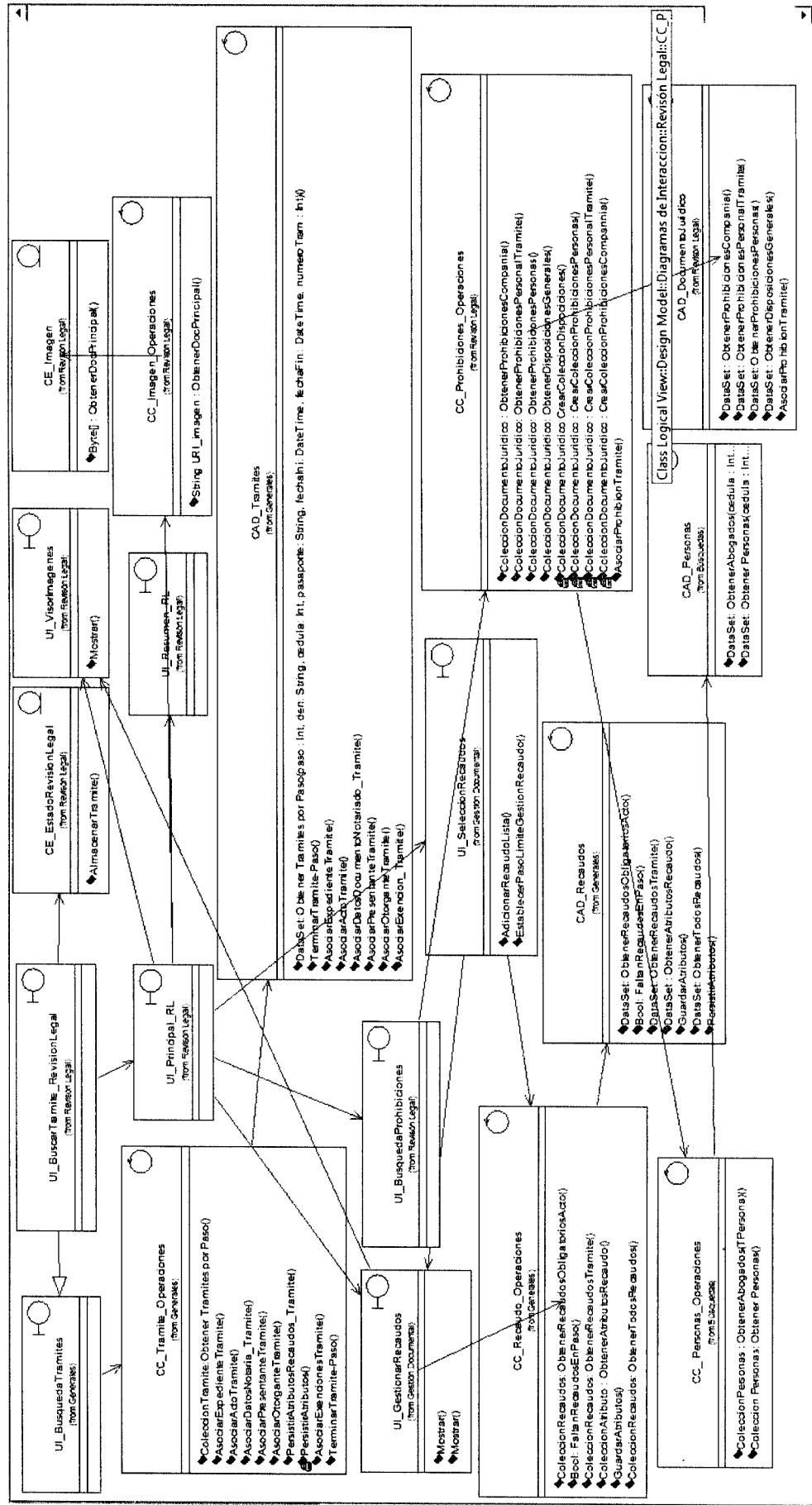


3.1.12. Presentación

Se incluye en este diagrama las clases y operaciones referentes a la Gestión Documental.



3.1.13. Revisión Legal



3.1.14. Modelo de Datos

Consultar este modelo en el **Anexo # 7**.

3.2. Modelo de Implementación.

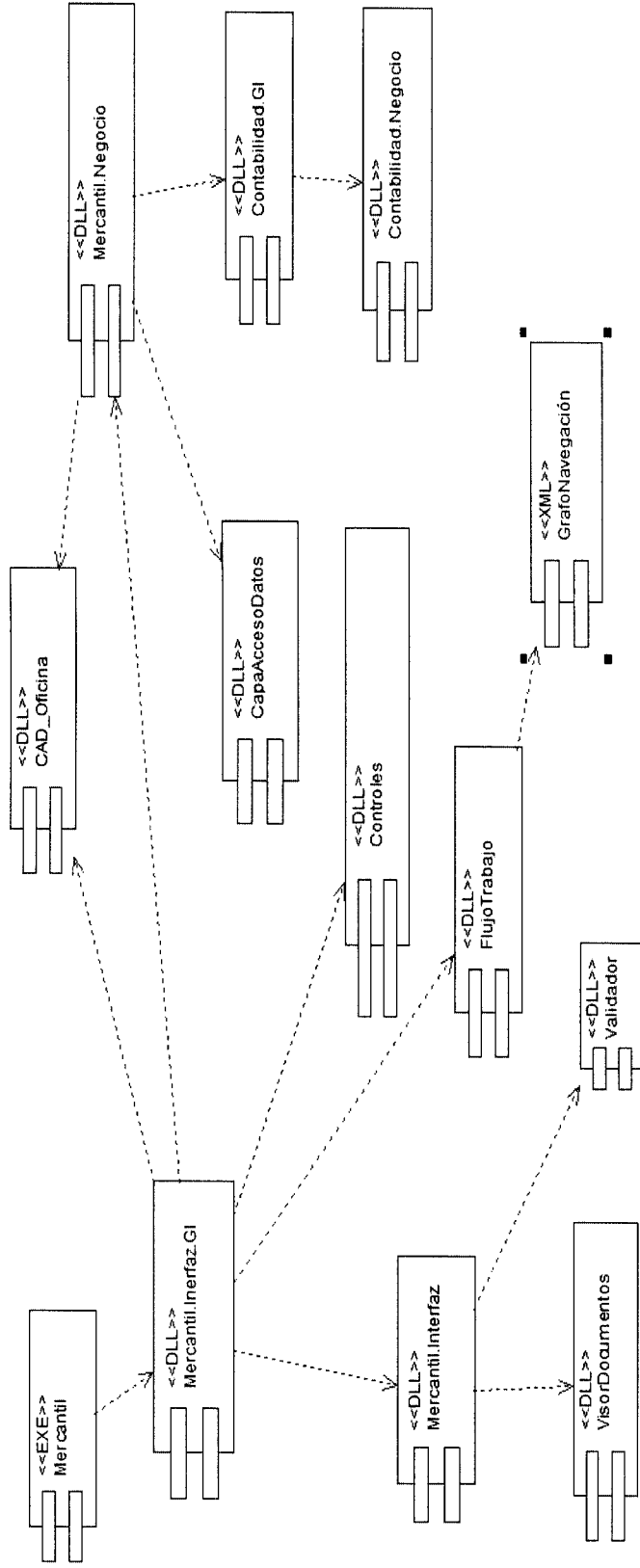
En este modelo se implementan el sistema diseñado en términos de componentes (archivos de código fuente, scripts, archivos de código binario, ejecutables, entre otros). Los propósitos de la implementación son los siguientes:

- Planificar las integraciones del sistema necesarias 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 fichero 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 y llevar a cabo las comprobaciones del sistema.

3.2.1. Diagrama de Componentes

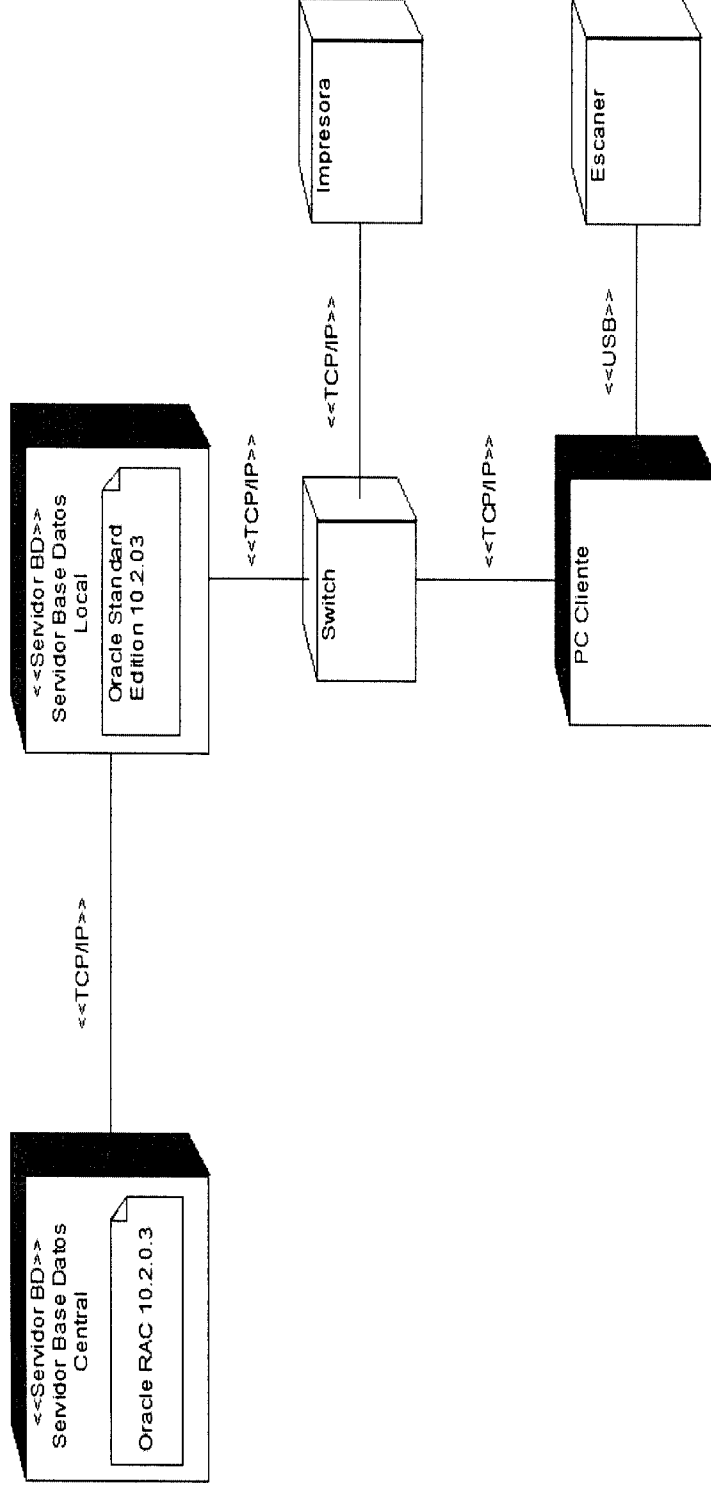
Se le llama componente al empaquetamiento físico de los elementos de un modelo, como son las clases en el Modelo de Diseño. Algunos estereotipos estándar de componentes son los archivos ejecutables, los archivos de código fuente, tablas de bases de datos, documentos en general, y otros.

Diseño e Implementación



3.2.2. Diagrama de Despliegue

Este modelo representa a través de nodos y relaciones entre ellos, el conjunto de dispositivos que interactuarían entre sí al desplegar el sistema implementado. Es decir por ejemplo, la comunicación entre computadoras, escáner, impresoras, entre otros.



Conclusiones

Con la implementación de este Módulo Mercantil, se arriban a las siguientes conclusiones:

1. Según el estado actual de los Registros Mercantiles, desde el punto de vista organizativo y la necesidad de un control centralizado, se demostró durante la etapa de pruebas piloto del módulo, que el sistema facilita la organización y control de todas las operaciones registrales de las oficinas.
2. Con la digitalización de los documentos y rápida visualización de estos para su consulta mediante el uso de este sistema, se logra la una mejor conservación de los mismos, así como cortos y eficaces tiempos de consulta al compararlo con el modo manual que actualmente es realizado por parte de los funcionarios en el archivo del Registro.
3. Este sistema se apoya en las Leyes Registrales para su correcto funcionamiento. Dígase con esto por ejemplo, que se contribuye al homogéneo proceso de cálculo de los gastos evitando de algún modo las equivocaciones que pudieron cometerse.
4. Los procesos de inscripción en los Registros cuentan en su parte inicial, con la selección búsqueda y reserva de las denominaciones seleccionadas por los usuarios. Actualmente el proceso de comparación de denominaciones es solo con las denominaciones registradas en el Registro donde fue solicitada, o vía telefónica se hacen también las consultas entre los diferentes Registros. Se logró y demostró entonces, que haciendo uso del sistema y su funcionalidad de Búsqueda Fonética de Coincidencias se garantiza:
 - a. Facilidad de trabajo para el funcionario, ya apoyándose en las búsqueda fonéticas cuenta con un margen de error al aceptar o denegar una denominación mucho mejor que el actual.
 - b. Todas las solicitudes de nombre de la república son accesibles y conocidas desde cada Registro.
 - c. Se evita de esta forma conflictos al eliminar la existencia de compañías con el mismo nombre, así como la claridad cuando se le informa al usuario el motivo por el cual se denegó su solicitud si es dado el caso.
5. El cálculo de gastos es avalado por leyes, por este motivo debe llevarse a cabo según los puntos de las mismas. Muchos Registros cuentan con diferentes sistemas o módulos que ayudan en la realización de las operaciones registrales, así como otros que todo lo hacen de forma manual. Este módulo de software realizado, le permite a los miembros del Servicio

Conclusiones

Autónomo definir según las leyes, los métodos para realizar el cálculo de los gastos en los Registros, contribuyendo de esta forma a un único método de cálculo para todas las oficinas de la nación.

6. Se logró realizar un organizado proceso de chequeo de prohibiciones por parte de los funcionarios, ya que el sistema le permite al funcionario hacer búsquedas centralizadas de prohibiciones asociadas a personas y compañías mediante el sistema, lo cual es rápido y eficaz si se tiene en cuenta la lejanía de muchos registros en el país. El actual método para realizar esta operación es ineficiente, dado que en ocasiones se tardan días o semanas en hacerles llegar los documentos a los Registros. Mientras que haciendo uso del módulo implementado, después de digitalizados los documentos que representan dichas prohibiciones por parte de los funcionarios de Servicio Autónomo, estos se descargan desde cada oficina de manera automática a través del sistema.

Recomendaciones

- Mejorar el componente de flujo de trabajo, incluyéndole una interfaz para mejorar el manejo de los flujos y la secuencia de ejecución de las acciones.
- Desarrollar un Framework que no sea rígido en cuanto al modo de presentar la información y que permita integrarse con los Bloques de Aplicación (Application Blocks) o mejorar el existente para que tenga estas funcionalidades.
- Documentar el Framework para que sea posible utilizarlo por otros desarrolladores.
- Desarrollar los diagramas de diseño e implementación de todos los procesos que se ejecutan en los Registros Mercantiles.

Bibliografía

Active Software Professional Alliance. Introducing Microsoft Enterprise Library Application Blocks. [En línea] http://aspalliance.com/1075_Introducing_Microsoft_Enterprise_Library_Application_Blocks.

Asamblea Nacional de la República Bolivariana de Venezuela. 22 de Diciembre de 2006. Ley de Registro Público y del Notariado de Venezuela. *Gaceta Oficial de la República Bolivariana de Venezuela*. 5833, 22 de Diciembre de 2006.

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

Congreso de la República Bolivariana de Venezuela. 1955. Código de Comercio. *Gaceta Oficial de la República Bolivariana de Venezuela*. 475, 1955.

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

El Proceso Unificado de Desarrollo de Software. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.**

Extreme Programming and Rational Unified Process – Contrasts or Synonyms? **Runeson y Greberg, Peter.**

Fonética y tecnologías del habla. **Joaquim Llisterri, Lourdes Aguilar, Juan M. Garrido, María Jesús Machuca, Rafael Marín, Carme de la Mota y Antonio Ríos.**

Gamma, Erich, y otros. 1994. *Design Patterns. Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wasley Longman, Inc., 1994.

Identificación de nombres personales por medio de sistemas de codificación fonética. **Llisterri, Joaquim, y otros.**

Kotec, Brian. MVC design pattern brings about better organization and code reuse. *Builder.Com*. [En línea] <http://builder.com.com/5100-6386-1049862.html>.

Métodologías Ágiles en el Desarrollo de Software. **Carnós, José H., Letelier, Patricio y Penadés, Carmen.**

2006. Model View Controller. *Web Application Component Toolkit*. [En línea] 9 de Diciembre de 2006. http://www.phpwact.org/pattern/model_view_controller.

Bibliografía

Real Academia Española. Diccionario de la Lengua Española. *Real Academia Española*. [En línea] [Citado el: 09 de 03 de 2007.] <http://www.rae.es/>.

Reynoso, Carlos y Kiccillof, Nicolás. 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. s.l. : Universidad de Buenos Aires, 2004.

TechTarget. Whatis.com. *The leading IT Encyclopedia and learning center*. [En línea] TechTarget. http://whatis.techtarget.com/definition/0,289893,sid9_gci1103696,00.html.

The Computer Language Company Inc. Encyclopedia PCMag. *PC Magazine*. [En línea] The Computer Language Company Inc. http://www.pcmag.com/encyclopedia_term/0,2542,t=application+framework&i=37907,00.asp.

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

Glosario de Términos

Administración Pública: “Organización ordenada a la gestión de los servicios y a la ejecución de las leyes en una esfera política determinada, con independencia del poder legislativo y el poder judicial. Conjunto de organismos encargados de cumplir esta función.” (Real Academia Española)

Registro: “En las diversas dependencias de la Administración Pública, departamento especial donde se entrega, anota y registra la documentación referente a ellas.” (Real Academia Española)

Dirección General de Registros y Notarías: encarga de la gestión y control de los procesos que se efectúan en cada dependencia, dígase Registros Civiles, Registros Públicos, Notarías y Registros Mercantiles.

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 presentado posteriormente, salvo las excepciones establecidas en esta 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

Asiento registral: Inscripción o anotación en un registro público. (Real Academia Española)

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

Fe pública: 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 Española)

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 Española).

Actos Constitutivos: traen como resultado de su inscripción la constitución de una Persona Jurídica apta para operar en territorio venezolano.

Actos No Constitutivos: su objetivo es actualizar las actividades tanto internas (Los propios de la vida interna de la sociedad, o sea los que se refieren al comportamiento de los socios entre sí y con la sociedad como persona jurídica distinta de ellos mismos individualmente considerados) y externa (Los que surgen en la vida externa de la sociedad, o sea entre la sociedad y los terceros con quienes ella contrata)

Otros Actos: son aquellos que no están directamente vinculados a una persona jurídica y que por lo tanto no se archivan (más adelante se explica el proceso de archivar documentos) junto al conjunto de los documentos relacionados con alguna persona jurídica.

Presentante: persona que presenta el documento en el Registro.

Otorgantes: una o más personas que firmaran el documento en Otorgamiento.

Recaudos: otros documentos que se acompañan con el Documento y que sirven para validarlo, tales como identificación del presentante y los otorgantes entre otros.

Smalltalk: Es un 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.

Dataset: Estructura de datos del Microsoft .Net framework que encapsula un conjunto de datos y soporta un modelo relacional de una o más tablas. (Bipin)

Proxy: Patrón de diseño cuya intención es proveer un sustituto para otro objeto para controlar el acceso a este.

Unidad Tributaria: un valor que se define anualmente en el República Bolivariana de Venezuela que se utiliza para las fórmulas de los cálculo de los impuestos o aranceles.

Fonética: Rama de la lingüística que estudia la producción, naturaleza física y percepción de los sonidos de una lengua.

Algoritmo: Es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema.

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

CLR: Entorno común de ejecución para los diferentes lenguajes del .Net Framework.

MSIL: Es un código de bytes (unidad más pequeña de almacenamiento usada en informática) que la Tecnología .NET de Microsoft utiliza para lograr independencia de la plataforma y seguridad en ejecución.

JIT: Es una técnica para mejorar el rendimiento de sistemas de programación en las tecnologías .NET.

UML: Lenguaje Unificado de Modelado, es el 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.

Anexo 1.

El sistema *Daitch-Mokotoff Soundex* desarrollado en 1985 por Randy Daitch y Gary Mokotoff (publicado un año después en *Avotaynu*, el diario de la genealogía judía, en un artículo titulado “The Jewish Soundex: a revised format”)¹⁸ y surge para solucionar el problema de la identificación de apellidos judíos, germánicos o eslavos. El sistema Daitch-Mokotoff codifica todos los sonidos en cifras, formando un código de 6 dígitos. Las letras o sonidos tienen diferentes valores si están al principio de la palabra, en el centro, variando si anteceden a una vocal o no.

Si no se alcanza a tener las 6 cifras, se completa con ceros, hasta llegar a los 6 dígitos, por ejemplo:

Blejsman 784660

Kestenbojm 543676

Drukker 395900

Fried 793000

Sharon 496000

Otra adaptación del método Soundex, pero en este caso, a los nombres franceses la constituye el denominado *Henry Code*. Este sistema también clasifica los nombres en códigos de tres letras, pero produce muchos fallos porque a menudo modifica la estructura fonética de las cadenas analizadas, además de generar falsas correspondencias entre nombres completamente diferentes, o no establecer una relación entre nombres similares (Bouchard & Pouyez, 1980).

No obstante, el auténtico problema de los sistemas anteriores es que no son capaces de establecer algún tipo de ordenación entre las cadenas similares. Este problema se resuelve con una variante de Soundex, denominada *Phonix* (Gadd, 1988, 1990), cuyo algoritmo es más complejo que sus predecesores. El método de codificación Phonix se basa en la sustitución de todos los caracteres

¹⁸ Para una información adicional sobre el sistema Daitch-Mokotoff Soundex, véase MOKOTOFF, G., AMDUR, S. Where once we walked. *Avotaynu*, p. 567-569, 2002.

menos el primero por valores numéricos, con una leve variación, como se muestra en la tabla 3, y en la eliminación de todas las apariciones del valor '0'. La novedad que introduce *Phonix* es que realiza previamente unas 163 transformaciones de grupos de letras que normalizan las cadenas (por ejemplo, el carácter 'X' se transforma en 'ECS', además si la primera letra es una vocal o la consonante 'Y' la transforma en 'V'). Sin embargo, la aportación más importante de este sistema de codificación es que computa los sonidos finales, y como consecuencia de esto es capaz de establecer tres rangos de similitud constituidos por palabras que concuerdan: en los sonidos finales, en los prefijos de los sonidos finales, o con sonidos finales distintos.

Tabla # 3 - Códigos fonéticos de Phonix

Código	Caracteres
0	a e h i o u w y
1	b p
2	c g j k q
3	d t
4	l
5	m n
6	r
7	f v
8	s x z

Un algoritmo de codificación fonética parecido a los anteriores lo constituye el sistema *Metaphone* (Philips, 1990). Se trata de un sistema de codificación especialmente diseñado para el inglés americano. El algoritmo de *Metaphone* elimina las vocales, aunque éstas permanecen si son la primera letra de una palabra, reteniendo solamente las consonante, que se reducen a 16 consonantes sin incluir los dígitos (aunque hay excepciones como '0' para representar el sonido 'TH'): B X S K J T F H L M N P O W Y. Además, se elimina la repetición de los caracteres consecutivos. Los códigos *Metaphone* estarían constituidos por cadenas que representarían aproximadamente cómo un nombre sonaría cuando se pronuncia usando las reglas de pronunciación de la lengua inglesa. El resultado de la aplicación del algoritmo *Metaphone* se muestra en la tabla 4.

Tabla # 4 - Resultado de la aplicación de Metaphone

	Apellidos	Variantes	Códigos
Coincidencias	Appelt	Apelt	APLT, APLT
	Hobbs	Hubbs	HBS, HBS
No Coincidencias	Appelt	Appell	APLT, APL
	Hobbs	Hobds	HBS, HBTS

Anexo 2.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión**, El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- **Ingeniería de Negocios**: Entendiendo las necesidades del negocio.
- **Requerimientos**: Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño**: Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación**: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas**: Asegurándose que el comportamiento requerido es el correcto y que todo el solicitado está presente.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto

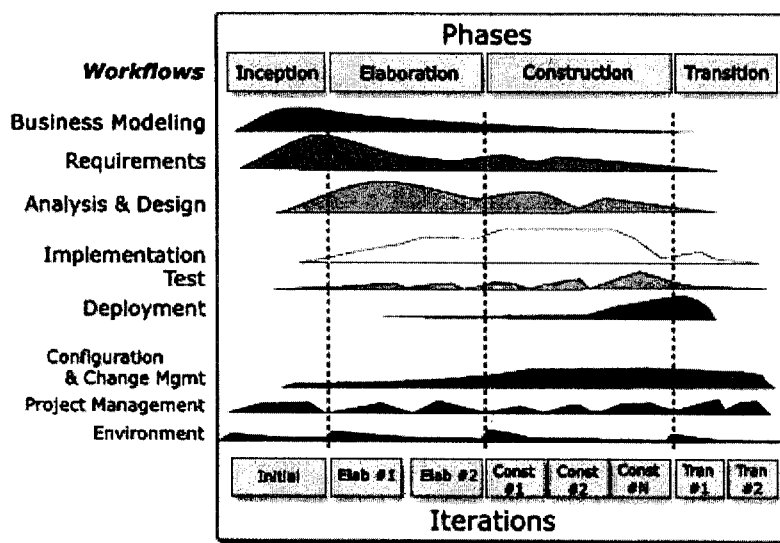


Figura 1: Fases e Iteraciones de la Metodología RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración. Los elementos del RUP son:

- **Actividades**, Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores**, Vienen hacer las personas o entes involucrados en cada proceso.
- **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Anexo 3.

Extreme Programing (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

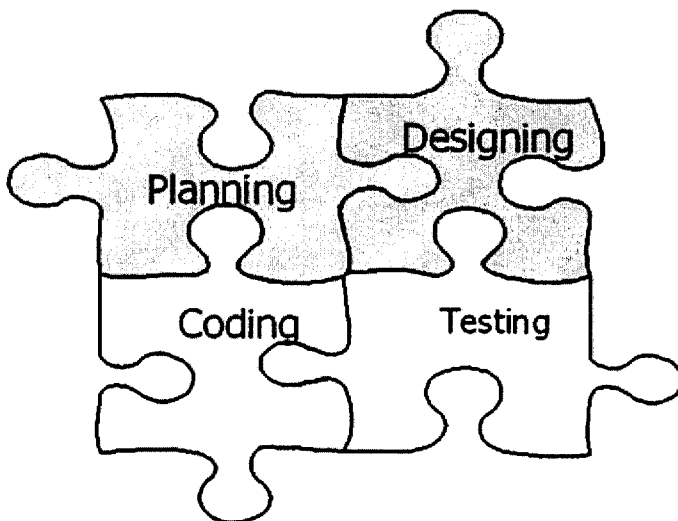


Figura 2: Metodología Extreme Programing

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Re fabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua
- El manejo del cambio se convierte en parte sustantiva del proceso
- El costo del cambio no depende de la fase o etapa
- No introduce funcionalidades antes que sean necesarias
- El cliente o el usuario se convierte en miembro del equipo

Derechos del Cliente

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Obtener un sistema funcionando cada 3 o 4 meses

Derechos del Desarrollador

- Decidir cómo se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores

Anexos

- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Anexo 4.

Este diagrama se encuentra en un fichero digital adjunto a la tesis.

Anexo 5.

Este diagrama se encuentra en un fichero digital adjunto a la tesis.

Anexo 6.

Este diagrama se encuentra en un fichero digital adjunto a la tesis.

Anexo 7.

Este diagrama se encuentra en un fichero digital adjunto a la tesis.