UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"Desarrollo del Sistema Integral Legislativo en la Suprema Corte de Justicia de la Nación"

I N F O R M E

QUE PARA OBTENER EL GRADO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

RODRIGO ARNOLDO SEVILLA THIERRY

DIRECTOR DE TESIS: M.C. ALEJANDRO VELÁZQUEZ MENA

México, D.F. **Junio 2012.**

A mis padres,

con mi gratitud infinita por su entrega y amor incondicional.

A mi hermano,

por su presencia entrañable en mi vida.

Al amor de mi vida,

por su apoyo y por el gran amor que me ha demostrado.

A mis maestros,

con mi admiración por sus grandes y valiosas enseñanzas.

A mi director de tesis M.C. Alejandro Velázquez Mena,

con mi reconocimiento por sus invaluables aportaciones

y acertados comentarios que enriquecieron este trabajo.

Al M.I. Jorge Valeriano Assem,

por la confianza depositada en mi

y la oportunidad de trabajar en este proyecto.

A la vida,

por ser cada día un mejor ser.

Índice

Introducción

El presente trabajo se enfoca al desarrollo del ‘Sistema Integral Legislativo’, en cuya participación precede por quien escribe el presente documento. Se menciona información general sobre el proyecto ‘Sistema de Informática Jurídica’, el cual es parte del convenio, pero no se realizó participación alguna de quien precede este documento.

Necesidades

La Dirección General del Centro de Documentación y Análisis, Archivos y Compilación de Leyes de la Suprema Corte de Justicia de la Nación, a través del área de Compilación de Leyes mantiene una constante labor de integrar y poseer un marco legislativo completo en beneficio de las funciones de los integrantes del Poder Judicial de la Federación y del público en general.

La creciente demanda de los servicios que brinda, la modernización en los medios de acopio y el acelerado e inesperado crecimiento de la base de datos que actualmente almacena toda su información en materia legislativa, implica que la Dirección de Compilación de Leyes requiera que los sistemas de información con los que trabaja al día de hoy, sean mejorados y adecuados a las necesidades actuales y a futuro tanto en capacidad, eficiencia y oportunidad con la que se ingrese y recupere la información.

Por lo que es necesario que dichos sistemas de información a su vez sean modernizados, contando con herramientas para la sistematización de la información en materia legislativa. Los sistemas de información ayudarán a soportar de una manera más adecuada sus procesos de trabajo y su homologación, así como también ayudarán en la optimización de los recursos humanos y materiales.

Se pretende realizar la completa reestructuración de los sistemas de información de la Dirección de Compilación de Leyes desarrollando un nuevo Sistema Integral Legislativo (SIL), el cual estará acoplado a las necesidades del área y conservando el *acervo* tanto documental como digital y los datos contenidos en las bases de datos con que actualmente se cuenta.

También se va a renovar, integrar y alinear a los nuevos procesos de compilación, los sistemas de recopilación y sistematización de *ordenamientos* de la institución, creando el Sistema Integral Legislativo de la Suprema Corte de Justicia de la Nación, que servirá como una herramienta de vanguardia para la compilación electrónica del acervo legislativo de este Alto Tribunal, y su posterior difusión a todo el Poder Judicial y público en general.

Se requiere del modelado de los procesos de trabajo de la Dirección de Compilación de Leyes con la intención de identificar con mayor claridad los requerimientos de funcionalidades que habrán de implementarse en el Sistema Integral Legislativo. Así como describir los escenarios bajo los cuales estará trabajando el nuevo sistema por medio de la definición de especificaciones funcionales.

Creación de documentación con las especificaciones técnicas bajo las cuales habrá de desarrollarse el sistema y que servirán como artefactos entregables para solicitar su codificación a la Universidad Nacional Autónoma de México.

Alcances

1. Identificar con exactitud las actividades que conforman cada uno de los procesos de trabajo de la Dirección de Compilación de Leyes.

2. Definir de una manera más clara y precisa los requerimientos que deberán ser expresados para el análisis y diseño del sistema.

3. Analizar los procesos de trabajo a los cuales estará dando soporte el sistema.

4. Diseñar la estructura técnica y tecnológica bajo la cual será construido el sistema.

Logros esperados

Para realizar la finalidad del proyecto es necesario realizar una descripción de las actividades que conforman cada uno de los procesos de trabajos identificados en la Dirección de Compilación de Leyes. Dicha descripción se hace con el objetivo de tener un amplio entendimiento de la forma de trabajo de la Dirección de Compilación de Leyes y como consecuencia, lograr la eficiente y eficaz automatización de dichos procesos en beneficio de los usuarios del Sistema Integral Legislativo, lo cual a su vez, se verá reflejado en:

- El adecuado control de las obras en materia jurídica que son donadas a la Suprema Corte de Justicia de la Nación y que a través de la Subdirección de Compilación Documental son integradas para ponerlas a disposición tanto del personal del Poder Judicial de la Federación como del público en general para su consulta.

- La optimización de los tiempos de respuesta al momento de ingresar y de recuperar la información para la integración de los ordenamientos jurídicos y para su cotejo.

- Agilización en la publicación de información ya sea en la Intranet, en el Internet y en la edición de los discos compactos que la Suprema Corte de Justicia de la Nación pone para su venta al público en general.

Figura I.1 Procesos de trabajo de la Dirección de Compilación de Leyes.

Desarrollo del proyecto

Para el desarrollo de la metodología del proyecto, se propusieron los siguientes pasos a seguir:

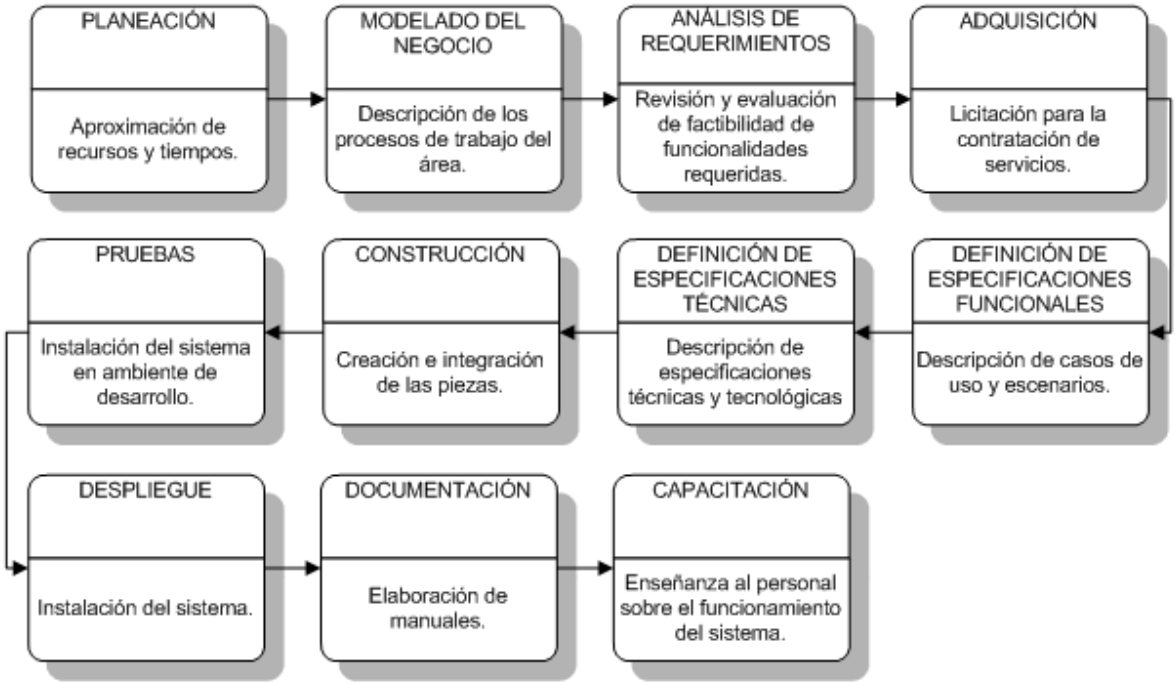


Figura I.2 Pasos para el desarrollo de la metodología.

Objetivos

- Mejorar, desarrollar y optimizar la estructura de un sistema informático ya empleado en la Suprema Corte de Justicia de la Nación, para el uso cotidiano de los empleados de la misma.

- Generar un sistema que sea más amigable con el usuario, así como documentación complementaria para el usuario, cliente y otros desarrolladores.

- Aplicación de la Ingeniería de software haciendo uso de los conocimientos y herramientas para la programación orientada a objetos en el lenguaje de programación C# y manejo de base de datos SQL, entre otras tecnologías necesarias para el desarrollo íntegro del sistema.

Capítulo 1. Antecedentes

1.1 Relación de proyectos UNAM / SCJN

La Universidad Nacional Autónoma de México en conjunto con la Suprema Corte de Justicia de la Nación, han llevado a cabo diferentes proyectos en diferentes áreas de la Ingeniería, algunos de los cuales son:

1.1.1 Convenio UNAM / 21355-132-5-II-08

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para llevar a cabo a través de la División de Ingeniería Eléctrica, el dictamen para definir la mejor opción de *ERP (Enterprise Resource Planning)* a implementar en la SCJN.

Vigencia y duración: del 16 de abril al 30 de agosto, 2008.

1.1.2 Convenio UNAM / 22644-1421-9-X-08

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para el desarrollo del Plan de Recuperación ante Desastres de la SCJN, el cual servirá para coadyuvar y asegurar el menor impacto de desastre, en caso de una interrupción de los servicios de Tecnologías de Información en sus centros de datos, en el cual se propuso la metodología y los elementos de información que faciliten el establecimiento de los procedimientos necesarios para la toma de decisiones y definición de las líneas de acción, para permitir el reanudo de las operaciones informáticas en el menor tiempo posible, ya sea para los casos de un desastre natural o por motivos humanos.

Vigencia y duración: del 1° de octubre al 31 de diciembre, 2008.

1.1.3 Convenio UNAM / 22240-1017-31-VII-08

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para el desarrollo de software del sistema 'Sistema de Informática Jurídica, primera fase'.

Vigencia y duración: del 1° de octubre al 31 de diciembre, 2008.

1.1.4 Convenio UNAM / 23507-392-19-III-09

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para el desarrollo de software para el sistema de 'Sistema de Informática Jurídica, segunda fase' y para el sistema 'Sistema Integral Legislativo, primera fase'.

Vigencia y duración: del 1° de junio al 31 de diciembre, 2009.

1.1.5 Convenio UNAM / 26150-860-17-V-10

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para la revisión del proyecto denominado 'Actualización del cableado estructurado para voz y datos del edificio sede de la Suprema Corte de Justicia de la Nación', en el cual la UNAM se encargó de asegurar que el diseño ejecutivo con el que cuenta la SCJN, cumplirá con los objetivos planteados en dicho proyecto; además de emitir las recomendaciones necesarias en caso de requerirse.

Vigencia y duración: del 15 de junio al 30 de agosto, 2010.

1.1.6 Convenio UNAM / 26131-841-13-V-10

Descripción: La Facultad de Ingeniería de la UNAM prestó sus servicios a la SCJN para el desarrollo, mantenimiento, actualización, administración y todas aquellas funciones directamente relacionadas con los proyectos 'Sistema de Informática Jurídica' y 'Sistema Integral Legislativo'.

Vigencia y duración: del 1° de julio del 2010 al 31 de mayo del 2011. Se cuenta con un periodo de garantía de software para el proyecto 'Sistema Integral Legislativo', comprendido del 1° de junio al 31 de agosto del 2011. [1]

1.2 Antecedentes del proyecto SIL

El principal objetivo de la Suprema Corte de Justicia de la Nación a través de este proyecto es reunir, organizar, conservar, controlar y poner a disposición para consulta física y electrónica el marco jurídico nacional.

La Subdirección de Compilación Documental es la encargada de organizar y revisar la actualización de acervos así como lo relativo a su préstamo y consulta a través de 4 Jefaturas de Departamento (Federal, Estatal, Procesos Legislativos y atención al público); paralelamente coordina al personal que se encuentra en las diversas *Casas de la Cultura Jurídica* del interior de la República a fin de que se realicen trabajos de compilación documental para su consulta en la entidad, por

consiguiente a razón de esta manera, descentralizar el servicio de atención al público en el edificio sede de este Alto Tribunal.

Los trabajos de compilación documental los realiza el personal de la Dirección General de Casas de la Cultura Jurídica y Estudios Históricos, realizando la captura de cronologías de *reformas* de las leyes más importantes y vigentes y el inventario de publicaciones y *tomos*. Dicho personal cuenta con acervo documental como lo son los *cuadernillos*, tomos y ejemplares de leyes, así como acervo electrónico como lo es la *red jurídica* nacional, Internet e inventarios de los Diarios Oficiales de la Federación desde el año de 2003 hasta la fecha en hojas de cálculo de Excel para ponerlos a consulta de funcionarios del Poder Judicial Federal y del público en general.

Las sedes metropolitanas localizadas en el Distrito Federal ubicadas en los edificios de Las Flores, de San Lázaro, y de Revolución realizan su labor de atención al público y de compilación documental de una manera muy similar a como lo hacen en las Casas de la Cultura Jurídica pero la materia se limita a los ámbitos Federal y del Distrito Federal. En dichas sedes, no se cuenta con los mismos recursos informáticos que en la Suprema Corte de Justicia de la Nación, motivo por el cual, sus controles de usuarios y servicios e inventarios son llevados en Excel o en su defecto, en Word.

1.2.1 Proceso de recopilación y organización

El proceso de recopilación y organización se puede resumir a grandes rasgos de la siguiente manera:

- El Departamento de Compilación Documental Estatal se encarga de ingresar los *extractos de las reformas* a los ordenamientos publicados en los Periódicos Oficiales de cada uno de los Estados de la República Mexicana.

- El Departamento de Compilación Documental Federal se encarga de ingresar los extractos de las reformas en material Federal, Distrito Federal e Internacional.

- Una vez hecha la captura de los extractos a los ordenamientos, se genera el cuadernillo (si no es que ya existía previamente) al cual se le imprime la portada y también se actualiza su índice.

- La Dirección de Compilación de Leyes recibe de la Cámara de Diputados, del Senado de la República, de la Asamblea Legislativa y de los Congresos Locales los documentos de *procesos legislativos* (y sus correspondientes textos) los cuales dieron origen a la reforma y que habrán de ser capturados y escaneados por el Departamento de Procesos Legislativos.

- El Departamento de servicio al público y acervo se encarga de resguardar las publicaciones oficiales en papel periódico, cuadernillos en papel periódico o fotocopias bond, legislación editada por empresas privadas, discos compactos y tomos de legislación histórica; así como de realizar el préstamo del acervo, localizar información específica que se encuentra en el mismo, investigar información que no se localiza en las publicaciones oficiales, llevar el control de inventarios en cualquiera de sus modalidades (cuadernillo, tomo, editoriales privadas, discos compactos, colecciones históricas y especiales), material que se encuentra en préstamo (boletas), registro de usuarios y servicios (actualización de bases), así como el desglose de la estadística correspondiente.

Debido a que es necesario localizar información de carácter jurídico que no necesariamente se publica en los periódicos oficiales y que es proporcionada por algunas dependencias de la Administración Pública Federal y Local, es necesario que una vez obtenida la información, esta pueda ser ingresada al acervo bajo una clasificación.

1.2.2 Resultados obtenidos

Los resultados obtenidos del proceso de recopilación y organización son:

- Ingreso a base de datos del registro inicial para obtener ordenamientos actualizados y confiables.
- Integración de cuadernillos y su actualización permanente.
- Obtención de publicaciones oficiales.
- Ingreso a base de datos de los textos de procesos legislativos.
- Préstamo de acervo y servicio de consulta de información legislativa en toda la República Mexicana.
- Control de inventarios, así como la conservación y resguardo del acervo documental.
- Síntesis Legislativa del *Diario Oficial de la Federación*.

1.2.3 Proceso de integración y sistematización

La Subdirección de Sistematización de Textos Legislativos recibe por parte de la Subdirección de Compilación Documental, Servicio al Público y Acervo un listado de ordenamientos nuevos o de fechas de publicación de reformas recientemente ingresados al banco de datos con la intención de que sean analizados, turnados y cotejados por los diversos Departamentos (Federal, Distrito Federal, Tratados Internacionales y Legislación Extranjera o al Departamento encargado de las 31 Entidades Federativas).

Dicho análisis consiste en la aplicación del *decreto* de reformas y la evolución que ha tenido la ley, a efecto de no aplicar inadecuadamente un párrafo o en su caso, aplicar notas aclaratorias. Con la finalidad de ampliar la información que habrá de ser analizada, en algunas ocasiones, se realiza una solicitud directa a los Congresos Locales o se consigue la información por medio de Internet o por medio de la *digitalización* de documentos.

Una vez que han sido catalogados los ordenamientos, se comienza el proceso de captura e ingreso a la base de datos en las modalidades de textos originales, decretos (puede ser algunas modificaciones menores al texto de la reforma previa y para ello se genera la *historia legislativa*), *fe de erratas*. Para estar en condiciones de ingresar los textos en la base de datos, es necesario que la Subdirección de Compilación Documental, Servicio al Público y Acervo haya dado de alta previamente los registros de las fechas de publicación.

El texto del ordenamiento está comprendido por tablas con tarifas, gráficas o fotografías las cuales son generadas de manera separada en donde se recopila esta información. Una vez que se recopila la información se asocia a su correspondiente artículo con la finalidad de que los usuarios finales puedan consultar la información de una forma clara y lo más apegada posible a como fue publicada en los diversos medios oficiales.

Cada departamento es el encargado de capturar el texto completo de los ordenamientos jurídicos que les correspondan así como de realizar el cotejo correspondiente.

Adicionalmente, la Subdirección de Sistematización de Textos Legislativos realiza la actualización de los instrumentos internacionales (tratados) lo cual consiste en una asociación de las tesis jurisprudenciales y aisladas emitidas por la Suprema Corte de Justicia de la Nación y por los Tribunales Colegiados de Circuito almacenadas en el *IUS* tomando como punto de partida los tratados ingresados en la base de datos, así como el índice temático con el que se cuenta realizando un análisis, selección y asociación en los apartados correspondientes de dicho índice temático.

Los resultados obtenidos del proceso de integración y sistematización son:

- Ingreso a base de datos de todos los registros que conforman el texto completo de los ordenamientos jurídicos.

- Ordenamientos actualizados, cotejados y confiables de todos los *ámbitos*.

- Asociaciones entre *tesis aisladas* y jurisprudenciales con ordenamientos en materia internacional.

1.2.4 Clasificación y selección de los ordenamientos

La Asesoría de Automatización Legislativa realiza una revisión selectiva de los ordenamientos que la Subdirección de Sistematización de Textos Legislativos reporta como trabajados y que van a ser publicados en las páginas Web o incluidos en la edición de algún disco compacto.

En caso de que el contenido del ordenamiento no esté correcto o se encuentre incompleto, es remitido de regreso a la Subdirección de Sistematización de Textos Legislativos para que sea corregido en la base de datos. Una vez concluida la corrección, dicho ordenamiento es actualizado en la Red Jurídica Nacional y en Internet.

Por otra parte si el ordenamiento tiene completo el *articulado* (es decir, que se hayan capturado todos los encabezados y artículos), se genera el documento en formato Word en donde se tiene completo el texto del ordenamiento. Adicionalmente, se copia al documento Word el texto de los artículos que tienen información como texto, tablas o gráficas.

Una vez que se ha revisado exhaustivamente el ordenamiento, se inicia la migración de los ordenamientos a las bases de datos correspondientes para su publicación en las páginas de la Intranet para lo cual, el área de Automatización y Digitalización Legislativa se encarga de establecer las fechas de corte de actualización para las bases de datos de los ámbitos Internacional, Federal y del Distrito Federal. La Subdirección de Sistematización de Textos Legislativos reporta por medio de un oficio interno aquellos ordenamientos que han sido trabajados.

Para el ámbito estatal, las fechas de corte de actualización se establecen tomando en consideración:

- La relación de fechas de la “última actualización compilada” de cada uno de los estados, la cual es enviada vía correo electrónico por el Departamento de Compilación Estatal.

- El informe sobre el “concentrado de actualizaciones en materia estatal” enviado semanalmente por la Subdirección de Compilación Documental, Servicio al Público y Acervo.

- El informe de la Subdirección de Sistematización de Textos Legislativos.

- Al concluir la revisión, corrección y generación de textos de Word ya sea de forma automática o manual en el caso de existir archivos anexos y migración a las bases de datos, se corroboran las fechas de los reportes y se procede a cambiar y actualizar las fechas de cada una de las entidades federativas.

En coordinación con la Dirección General de Informática, y una vez que se ha hecho la migración de los ordenamientos a su correspondiente base de datos y que se ha establecido la fecha de su actualización, se genera la actualización de las bases de datos que se tienen para consulta de los usuarios por medio de la página de Internet de la Suprema Corte de Justicia de la Nación (www.scjn.gob.mx) y poder así, brindar el servicio de consulta de los ordenamientos jurídicos a través de dicha página.

Se solicita a la Dirección General de la Coordinación de Compilación y Sistematización de Tesis su indexación, para su revisión en línea, tanto en su contenido como en la operatividad y se realizan las adecuaciones al manual de ayuda.

Los resultados obtenidos del proceso de clasificación o selección son:

- Publicación del marco jurídico nacional e internacional a través de la Intranet e Internet para consulta tanto de funcionarios del Poder Judicial Federal así como de personal en general.

- Edición de discos compactos para su venta al público en general. [2]

1.3 Ley Federal de transparencia

La Ley Federal de transparencia es de orden público. Tiene como finalidad proveer lo necesario para garantizar el acceso de toda persona a la información en posesión de los Poderes de la Unión, los órganos constitucionales autónomos o con autonomía legal, y cualquier otra entidad federal. [3]

A través de la Ley Federal de transparencia el Programa Anual de Ejecución de Adquisiciones, Arrendamientos y Prestación de Servicios y Programa Anual de Ejecución de Obra Pública y Servicios Relacionados, da conocimiento sobre dichos sistemas:

Tabla 1.1 Programa Anual de Ejecución de Adquisiciones, actualizado al año 2011. [4]

N°	CAPÍTULO DE GASTO	PARTIDA PRESU-PUESTA-RIA	CONCEPTO	UNIDAD DE MEDIDA	VALOR TOTAL ESTIMADO CON IVA M.N. (MILES DE PESOS)	PERIODO ESTIMADO DE CONTRATACIÓN		VIGENCIA DE LOS SERVICIOS	DESTINO DE LOS BIENES Y/O SERVICIOS
						1er SEMESTRE	2do SEMESTRE		
38	3000	33301	SISTEMA DE INFORMÁTICA JURÍDICA: servicios para el desarrollo de software con diferentes perfiles de especialistas, en el esquema de consumo bajo demanda.	SERVICIO	5,000.00		JUL	SEP-DIC	D.F.
42	3000	33301	SISTEMA INTEGRAL LEGISLATIVO: servicios para el desarrollo de software con diferentes perfiles de especialistas, en el esquema de consumo bajo demanda.	SERVICIO	2,000.00	JUN		AGO-DIC	D.F.

Capítulo 2. Proyectos involucrados

En este capítulo se presenta de forma general un resumen para las mejoras del Sistema de Informática Jurídica (SIJ) y el desarrollo del Sistema Integral Legislativo (SIL).

2.1 Objetivos generales de los proyectos

Desarrollar componentes, nuevos módulos, mejoras y mantenimiento del Sistema de Informática Jurídica, esto con el fin de estabilizar la gestión, seguimiento y difusión electrónica de expedientes judiciales de la Suprema Corte de Justicia de la Nación.

Desarrollar el Sistema Integral Legislativo incluyendo etapas de concepción, elaboración, construcción y transición del sistema, permitiendo el desarrollo de los módulos: Ordenamiento, Consulta General y Consulta Avanzada.

2.2 Objetivo específicos de los proyectos

- Desarrollar un plan de trabajo con base en los requerimientos establecidos en el convenio 2009 y 2010.

- Establecer el procedimiento para la administración de la configuración de software, desde su concepción hasta su liberación final.

- Desarrollar un proceso de pruebas para validar las diversas funcionalidades de la aplicación.

- Llevar a cabo las revisiones de aseguramiento de la calidad de procesos y productos (CMMI - Process and Product Quality Assurance, PPQA).

- Llevar un control de cambios para cada solicitud de cambio.

2.3 Módulos relacionados por proyecto

2.3.1 Módulos SIJ

El desarrollo de mejoras, componentes y módulos establecidos para el Sistema de Informática Jurídica (SIJ) son los siguientes:

a) Mejoras y Mantenimiento al SIJ.
b) Funcionalidad para Tesis y Jurisprudencia.
c) Componente de Seguimiento físico.
d) Componente de Alertas por Correo.
e) Componente de Autocompletado.
f) Reportes de control (Informes).
g) Versión gráfica de la línea de tiempo de expedientes.
h) Módulo de Correspondencia.
i) Módulo de Archivo.
j) Módulo de Tablero Web.
k) Módulo de Configuración.
l) Módulo de Auditoría.
m) Módulo de Ventanilla Jurídica.
n) Módulo de Generación de Plantillas.
o) Módulo de Planeación de lo Jurídico.

2.3.2 Módulos SIL

El desarrollo del Sistema Integral Legislativo (SIL) contempla los siguientes módulos:

a) Módulo de ordenamiento.
b) Módulo de consulta general.
c) Módulo de consulta avanzada.

2.4 Limitaciones y exclusiones

Para el desarrollo del Sistema Integral Legislativo y mejoras al Sistema de Informática Jurídica, se requirió del convenio UNAM-SCJN 2009 y UNAM-SCJN 2010.

Dentro de las actividades de la UNAM no se incluyen:

- Diseño de la base de datos.
- Mantenimiento de bases de datos.

- Migración de base de datos.
- Captura, carga de información y depuración de datos.
- Instalación del sistema en equipos de cómputo del usuario final.
- Desarrollo de aplicaciones no establecidas en los convenios.
- Capacitación de base de datos, lenguajes de programación, sistemas operativos, administradores de correos y otras tecnologías.
- Viáticos y pasajes para realizar trabajos fuera del área metropolitana.

2.5 Arquitectura de los sistemas

La arquitectura propuesta para la realización de estos sistemas está basada en el concepto SOA (Arquitectura Orientada a Servicios), que define la utilización de servicios para dar soporte a los requisitos del negocio, proporcionando una metodología y un marco de trabajo para documentar las capacidades de negocio, dando soporte a actividades de integración y consolidación de la organización. En la Figura 2.1 se muestran las capas de integración SOA.

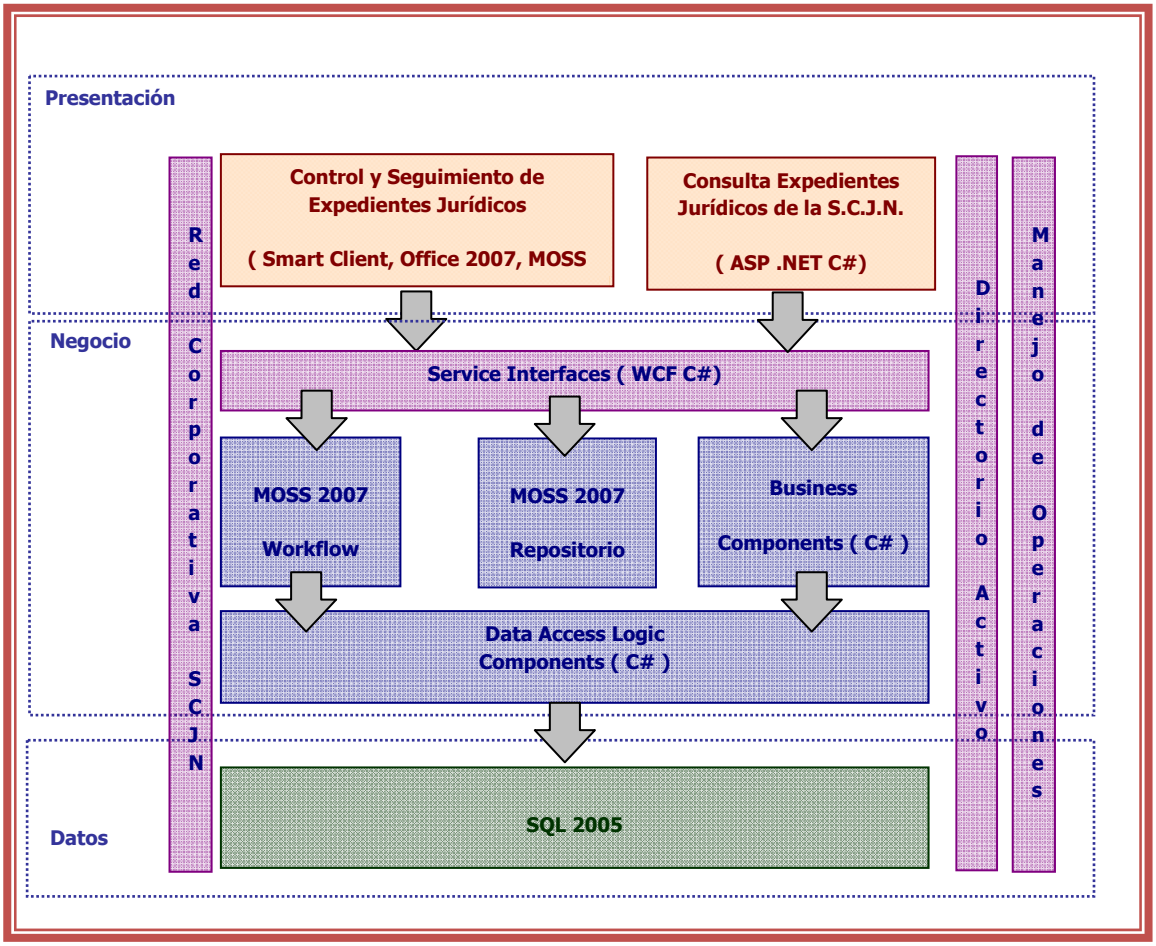


Figura 2.1 Arquitectura orientada a servicios (SOA).

Con una Arquitectura Orientada a Servicios (SOA), los usuarios no tienen que iniciar sesión en varios sistemas, buscar los datos relevantes e integrar los resultados manualmente. Los datos de las actividades de los procesos de negocios se entregan como un servicio integrado, en una sola aplicación y con un sólo inicio de sesión.

El diagrama UML es de la siguiente forma:

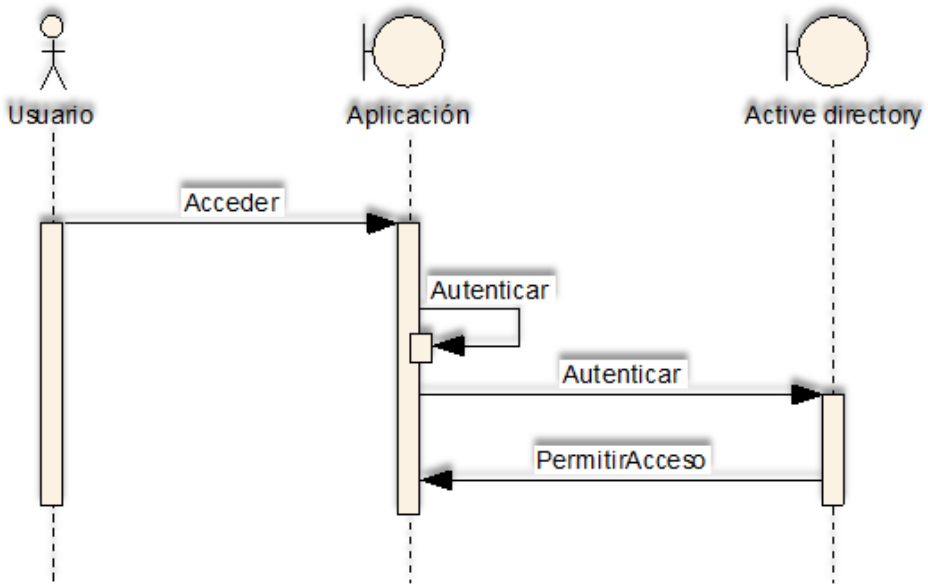


Figura 2.2 Diagrama UML de la conexión al servicio.

2.6 Perfiles y roles

Los roles relevantes de los involucrados son:

2.6.1 Líder de proyecto

El Líder de Proyecto (LP) es quien coordinará a los integrantes del equipo de trabajo y realizará funciones de comunicación entre la UNAM y el cliente, interactuando entre el Gerente de Proyectos y cada uno de los líderes técnicos de la SCJN. Las funciones del LP son las siguientes:

- Tomar decisiones en cualquier momento.
- Acudir a las instalaciones del cliente para realizar las actividades de seguimiento hasta la formalización del término de los servicios contratados.
- Canalizar los servicios de atención a fallas y soporte técnico.
- Desarrollar el plan de trabajo del proyecto.
- Contribuir al buen desarrollo de las actividades del proyecto.

- Elaborar informes ejecutivos para reportar en las reuniones de seguimiento los avances y monitoreo del plan del proyecto con la finalidad de documentar los acuerdos derivados al respecto.
- Identificar, calificar y documentar riesgos y presentarlos al gerente de proyecto del cliente.
- Establecer y revisar con el gerente de proyecto del cliente los *hitos* por cada fase e iteración del proyecto.
- Coordinar la validación de los requerimientos con cada líder de proceso de Información.
- Documentar con los líderes por proceso de información de cada proceso, especificando los objetivos, datos y procedimientos involucrados (además de las minutas, cuestionarios, y documentación soporte.).
- Establecer, documentar, validar y controlar el alcance de cada requerimiento a través del proceso de administración de requerimientos y control de cambios.
- Garantizar la integridad de los artefactos mediante el establecimiento de un proceso formal de administración de la configuración.

2.6.2 Arquitecto de software

El arquitecto de software realizará las siguientes funciones:

- Analizar y diseñar la propuesta de solución.
- Diseñar la estructura y diccionario de datos de la base de datos.
- Coordinar a los ingenieros de software.
- Obtener el entendimiento de los requerimientos del producto.
- Responsable de la arquitectura del sistema, presentada principalmente en los modelos de diseño, implementación y despliegue, es decir, de la existencia y la calidad de las decisiones más significativas para la construcción del sistema.
- Revisar los productos de implementación generados por los ingenieros de software, para asegurar la integridad de la arquitectura.
- Planear la secuencia de la construcción del sistema que sea necesaria en cada iteración y asegurar la integración de cada construcción.
- Recibir los reportes de defectos y dar el seguimiento correspondiente a los mismos hasta su resolución.
- Mantener informado al líder de proyecto del estado de la calidad de los productos mediante estadísticas reportadas a partir de los resultados de las pruebas.
- Planear, diseñar y ejecutar los distintos tipos de pruebas (funcionales, técnicos de sistema, de volumen, de estrés, etc.) basadas en un proceso formal incluyendo el diseño de pruebas funcionales basadas en los casos de uso y escenarios definidos para el requerimiento para garantizar los objetivos de calidad de cada requerimiento a desarrollar.

2.6.3 Ingeniero de software

Desarrollar y mantener el código fuente de uno o varios componentes, garantizando que cada componente implementa la funcionalidad correcta apegándose a los estándares establecidos, su rol consiste en:

- Realizar las pruebas del código generado.
- Resolver solicitudes de Soporte Técnico y de Fallas.

2.6.4 Aseguramiento de calidad (QA)

Su rol consiste en:

- Realizar auditorías para validar la adherencia a procesos y productos asociados.
- Reportar y dar seguimiento a las desviaciones encontradas.

2.6.5 Analista

Su rol consiste en:

- Elaborar los manuales de usuario del sistema.
- Generar el material de capacitación.
- Administrar y documentar las actividades.

2.7 Organigrama

El equipo de trabajo va a estar compuesto de la siguiente manera:

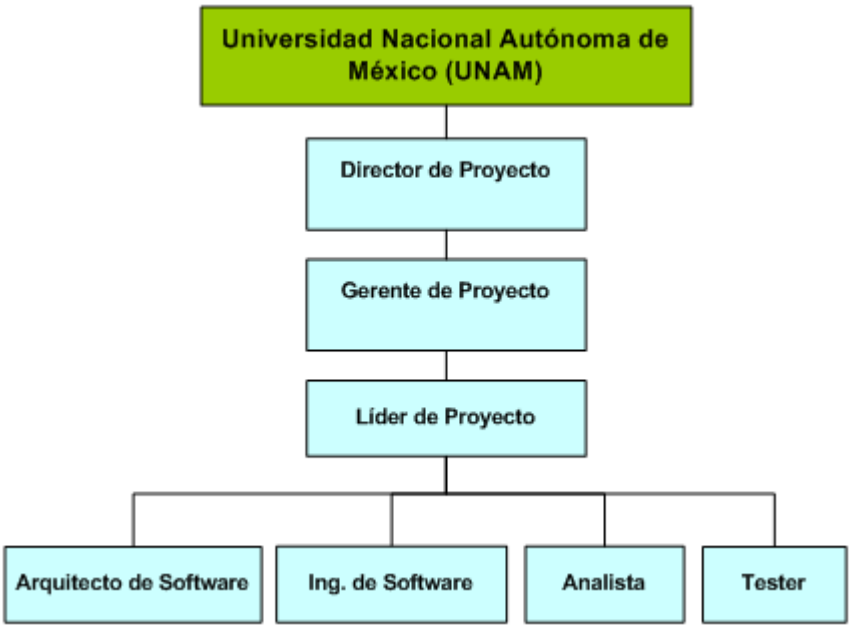


Figura 2.3 Organigrama del equipo de trabajo, UNAM.

[2]

Capítulo 3. Sistema Integral Legislativo

3.1 Arquitectura

El desarrollo del sistema se basó en una arquitectura orientada a servicios, sin embargo debido a mejoras y a petición del cliente, la arquitectura pasó a ser una arquitectura cliente / servidor de tres capas, en la cual se tiene:

3.1.1 Arquitectura orientada a servicios

La Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture) supone una estrategia general de organización de los elementos de IT, de forma que sistemas distribuidos y aplicaciones complejas se pueden transformar en una red de recursos integrados, simplificados y flexibles. Permite alinear los recursos de IT de forma más directa con los objetivos de negocio, proporcionando una inteligencia de negocio más precisa y accesible con la cual se podrán adoptar mejores decisiones para la optimización de procesos internos y flujos de información para mejorar la productividad individual. [5]

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes, de manera que se pueda acceder a las funcionalidades, las cuales se establecen como servicios. La forma más habitual de implementación es mediante Servicios Web, los cuales son una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. [6]

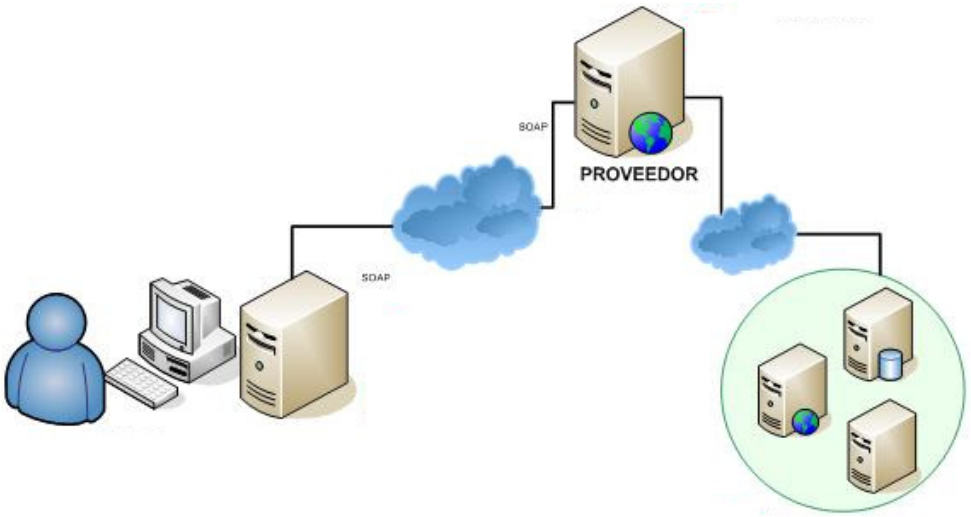


Figura 3.1 Arquitectura SOA.

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos de IT a través de la red. Los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente.

Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Los servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración. Se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP (Simple Object Access Protocol) para el intercambio de datos y el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades de un servicio Web. [7]

Existen tres tipos de servicios:

- *Servicios controladores*: Son los encargados de recibir las peticiones de los clientes y realizar las llamadas necesarias a otros servicios (en la secuencia adecuada) para devolver una respuesta.

- *Servicios de negocio*: Son los servicios que representan una tarea de negocio, y que forman parte de un proceso de negocio.

- *Servicios de utilidad*: Son aquellos servicios que se caracterizan por representar una tarea altamente reutilizable.

Una aplicación SOA se puede dividir en tres capas:

- La capa de recepción de peticiones (servicios controladores).

- La capa de tareas (servicios de negocio).

- La capa de lógica reutilizables (servicios de utilidad). [8]

Se requiere de cuatro elementos esenciales para la construcción de una Arquitectura Orientada a Servicios:

- *Operación*: Es la unidad de trabajo o procesamiento en una arquitectura SOA.

- *Servicio*: Es un contenedor de lógica, el cual estará compuesto por un conjunto de operaciones, las cuales las ofrecerá a sus usuarios.

- *Mensaje*: Para poder ejecutar una determinada operación, es necesario un conjunto de datos de entrada, una vez ejecutada la operación, esta devolverá un resultado. Los mensajes son los encargados de encapsular esos datos de entrada y de salida.

- *Proceso de negocio*: Son un conjunto de operaciones ejecutadas en una determinada secuencia (intercambiando mensajes entre ellas) con el objetivo de realizar una determinada tarea. [9]

Los beneficios de utilizar SOA son los siguientes:

- *Mejorar la toma de decisiones*: Al integrar el acceso a los servicios e información de negocio dentro de un conjunto de aplicaciones dinámicas compuestas, los directivos disponen de más información y de mejor calidad (más exacta y actualizada).

- *Mejorar la productividad de los empleados*: Un acceso óptimo a los sistemas y a la información, así como la posibilidad de mejorar los procesos que permiten a las empresas aumentar la productividad individual de los empleados.

- *Potenciar las relaciones con clientes y proveedores*: Las ventajas de SOA trascienden las fronteras de la organización. Los procesos de fusión y compra de empresas se hacen más rentables al ser más sencilla la integración de sistemas y aplicaciones diferentes.

- *Aplicaciones más productivas y flexibles*: La estrategia de orientación a servicios permite a IT conseguir una mayor productividad de los recursos de IT existentes y obtener mayor valor de sin necesidad de aplicar soluciones de integración desarrolladas ex profeso para este fin.

- *Desarrollo de aplicaciones más rápido y económico*: El diseño de servicios basado en estándares facilita la creación de un repositorio de servicios reutilizables que se pueden combinar en servicios de mayor nivel y aplicaciones compuestas en respuesta a nuevas necesidades de la empresa.

- *Aplicaciones más seguras y manejables*: Las soluciones orientadas a servicios proporcionan una infraestructura común y documentación para desarrollar servicios seguros, predecibles y gestionables. [10]

3.2.2 Arquitectura cliente / servidor

Este tipo de arquitectura es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

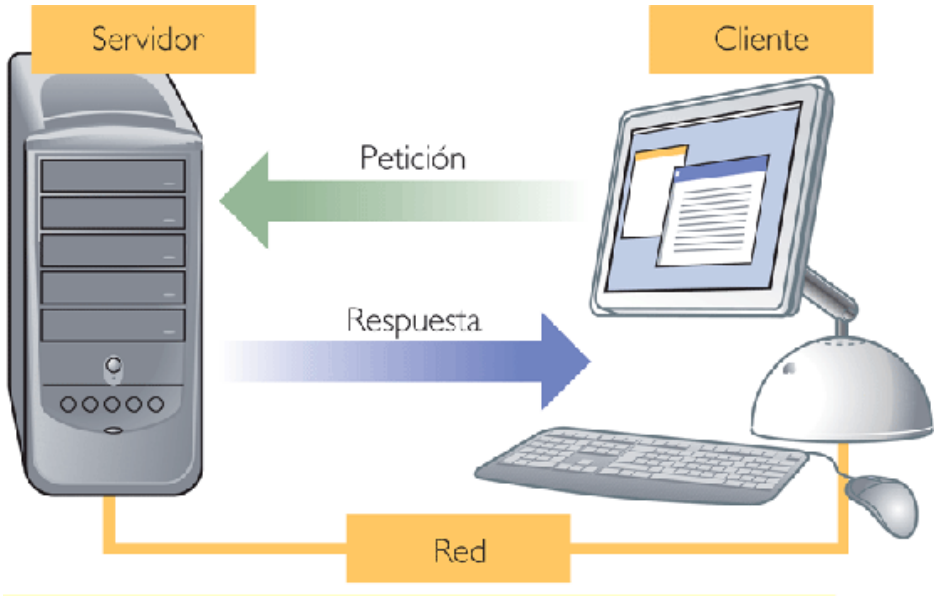


Figura 3.2 Arquitectura Cliente/Servidor

Un sistema cliente / servidor es un Sistema de Información distribuido basado en las siguientes características:

- *Servicio*: unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- *Recursos compartidos*: Los clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- *Protocolos asimétricos*: Los clientes inician peticiones. Los servidores esperan su establecimiento pasivamente.
- *Transparencia de localización física de los servidores y clientes*: El cliente no tiene conocimiento de dónde se encuentra situado el recurso que desea utilizar.
- *Independencia*: Indistinto de la plataforma HW y SW que se emplee.
- *Sistemas acoplados*: Interacción basada en envío de mensajes.

- *Encapsulamiento de servicios*: Los detalles de la implementación de un servicio son transparentes al cliente.

- *Escalabilidad*: Horizontal (añadir clientes) y Vertical (ampliar potencia de los servidores).

- *Integridad*: Datos confiables. [11]

Cliente: Es todo proceso que reclama servicios de otro en el cual permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término Front-End. El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos.

Las funciones que lleva a cabo el proceso cliente son:

- Administrar la interfaz de usuario.

- Interactuar con el usuario.

- Procesar la lógica de la aplicación y hacer validaciones locales.

- Generar requerimientos de bases de datos.

- Recibir resultados del servidor.

- Dar formato a los resultados.

Servidor: Es todo proceso que proporciona un servicio a otros, en el cual esta encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Se lo conoce con el término Back-End.

Las funciones que lleva a cabo el proceso servidor son:

- Aceptar los requerimientos de bases de datos que hacen los clientes.

- Procesar requerimientos de bases de datos.

- Formatear datos para trasmitirlos a los clientes.

- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos. [12]

3.2.3 Arquitectura de tres capas

El sistema se compone por una arquitectura de tres capas, las cuales son:

- *Capa de Presentación*: Es la capa encargada de implementar una interfaz con el usuario. Interacciona con el usuario, presenta los datos y recibe las entadas.

- *Capa de Negocio*: Es la capa encargada del procesamiento de la lógica del negocio. La capa de presentación se relaciona con la capa de negocio únicamente para ejecutar las acciones requeridas por el usuario.

- *Capa de Acceso a Datos*: Es la capa encargada de la gestión y almacenamiento de los datos en el servidor. [13]

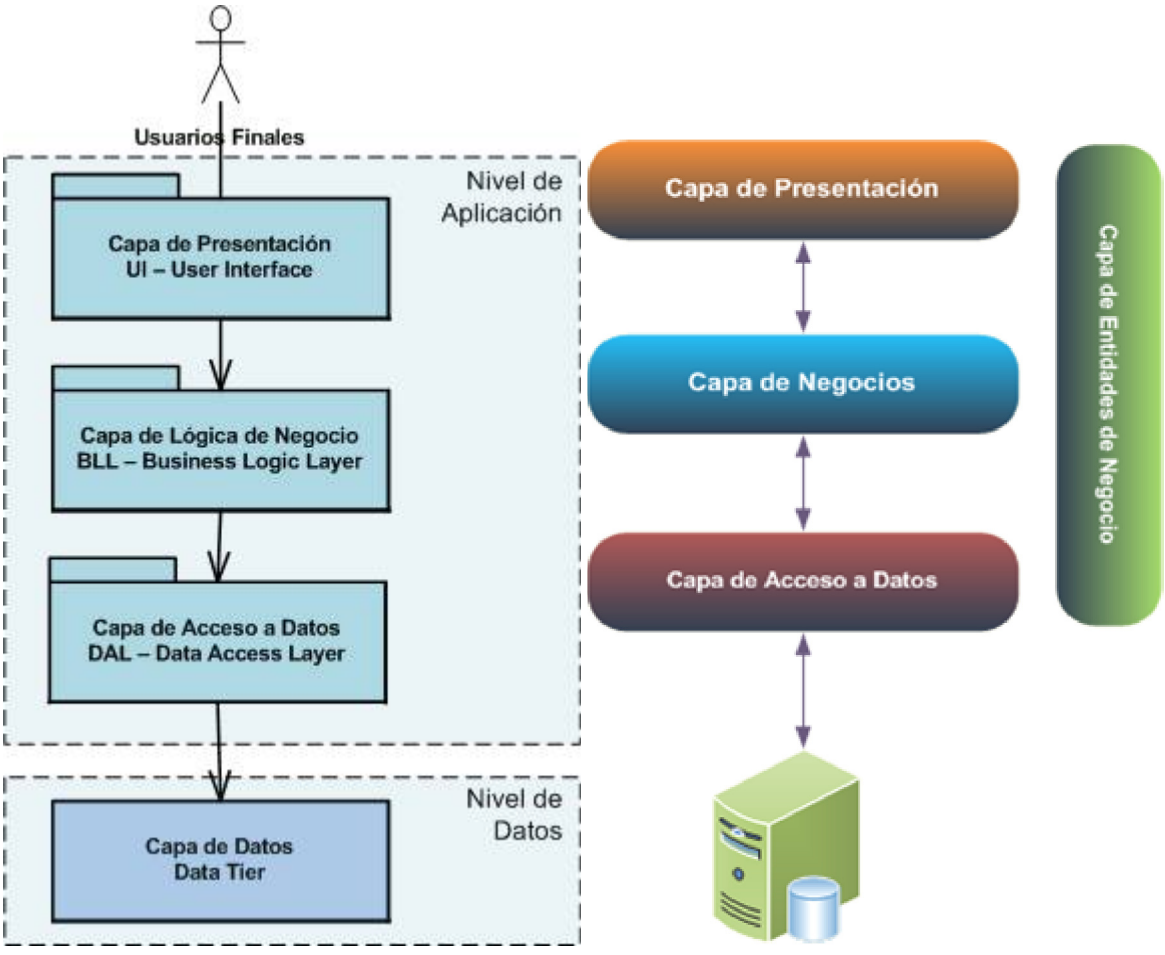


Figura 3.3 Arquitectura de tres capas.

3.2.4 Diagrama de la arquitectura

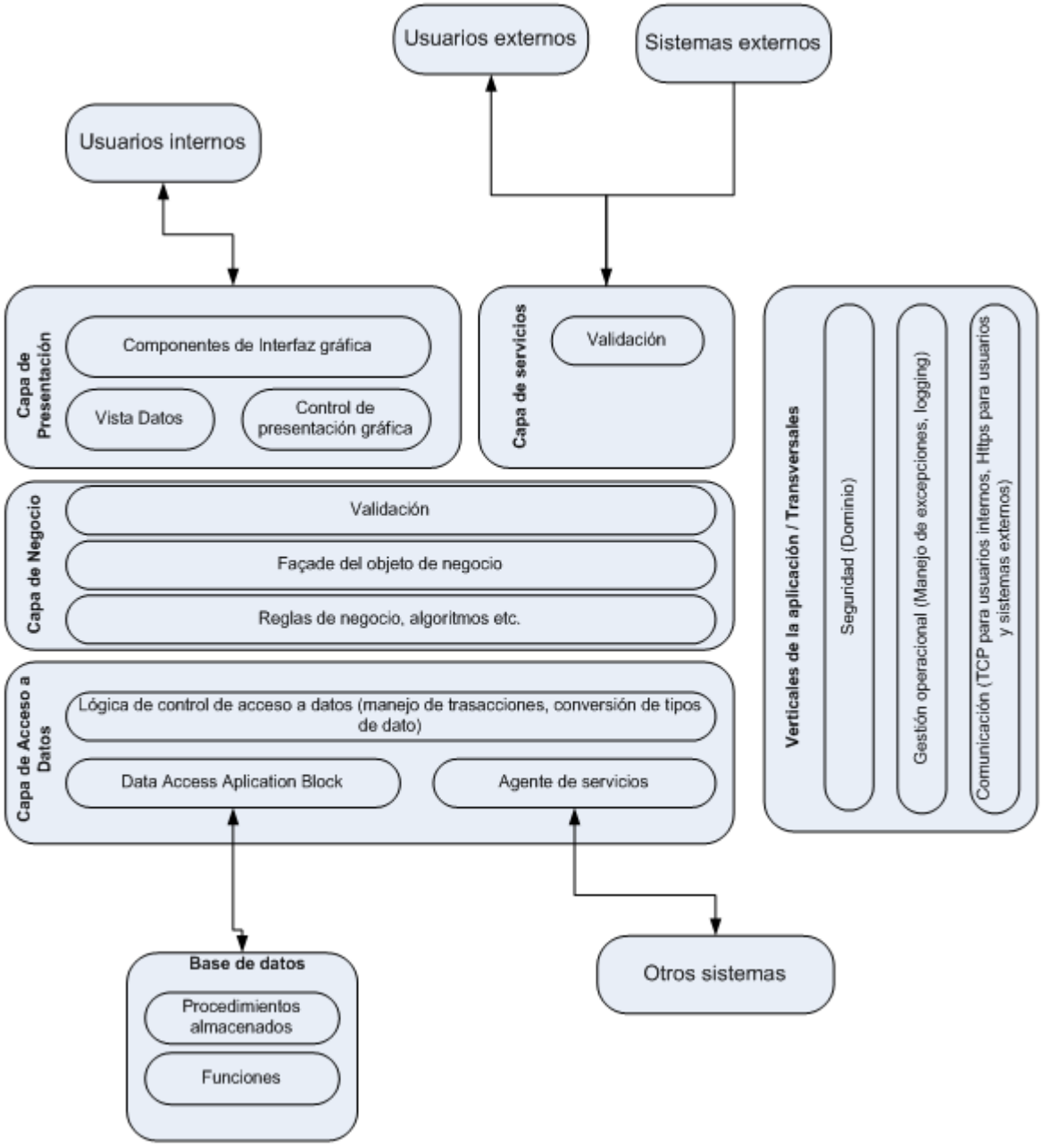


Figura 3.4 Diagrama de la arquitectura. [14]

3.2 Modelo de software

El modelo empleado para el desarrollo de software se basó en el desarrollo iterativo y creciente (o incremental).

3.2.1 Desarrollo iterativo e incremental

En el cual el proyecto se planifica en diversos bloques temporales llamados iteraciones.

Las iteraciones se pueden entender como pequeños proyectos en los que todas las iteraciones repiten un proceso de trabajo similar para proporcionar un resultado completo sobre producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. [15]

Cada requisito se debe completar en una única iteración en el cual el equipo debe realizar todas las tareas necesarias para completarlo (incluyendo pruebas y documentación) y realizar el entregable para el cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos. [16]

En cada iteración el equipo evoluciona el producto (entrega incremental) a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos requisitos o mejorando los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la priorización de los requisitos en función del valor que aportan al cliente. [17]

Las ventajas de la utilización de este modelo son las siguientes:

- Resolución de problemas de alto riesgo en tiempos tempranos del proyecto.
- Visión de avance en el desarrollo desde las etapas iniciales del desarrollo.
- Obtención de la retroalimentación del usuario lo antes posible, para orientar el desarrollo al cumplimiento de sus necesidades y realizar todas las adaptaciones identificadas para cumplir con los objetivos planteados.
- Menor tasa de fallo del proyecto, mejor productividad del equipo, y menor cantidad de defectos.
- Permite manejar la complejidad del proyecto, apuntando a la resolución de los problemas por partes, y no caer en la inanición del análisis exhaustivo del producto.

- El aprendizaje y experiencia del equipo iteración tras iteración, mejora exponencialmente el trabajo, aumenta la productividad y permite optimizar el proceso en el corto plazo.

- El trabajo iterativo deja una experiencia en el equipo que permite ir ajustando y mejorando las planificaciones, logrando menores desvíos en la duración total del proyecto. [18]

3.3 Metodología de software

La metodología empleada en el sistema se basó en la Programación Extrema (Extreme Programming, XP).

3.3.1 Programación Extrema

La programación extrema es una metodología de desarrollo ligera (o ágil) basadas en el objetivo de aumentar la productividad de desarrollo. Se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo, generando una metodología única y compacta.

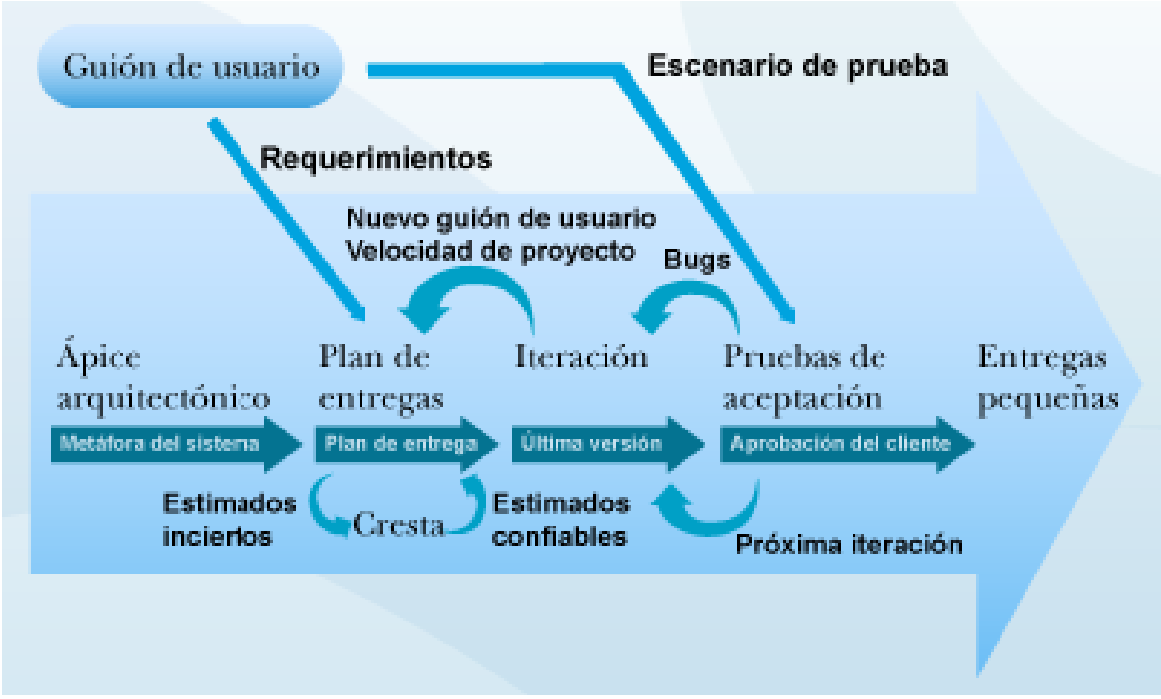


Figura 3.5 Programación Extrema.

Valores de la programación extrema:

- *Comunicación*: XP ayuda mediante sus prácticas a la comunicación entre los integrantes del grupo de trabajo: jefes de proyecto, clientes y desarrolladores

- *Sencillez*: Los programas deben ser los más sencillos posibles y tener la funcionalidad necesaria que se indican en los requisitos. No hay la necesidad de añadir algo que se requiera de forma inmediata.

- *Retroalimentación*: Las pruebas que se le realizan al software nos mantienen informados del grado de fiabilidad del sistema. Los clientes y los usuarios del sistema tienen una retroalimentación real.

- *Valentía*: Asumir retos, ser valientes ante los problemas y afrontarlos, esto se refleja al intentar mejorar algo que ya funciona. [19]

3.3.2 Objetivos de la programación extrema

- *Satisfacción del cliente*: El cliente debe recibir los que necesita en tiempo y forma, por lo que es necesario responder a las peticiones del cliente lo antes posible, incluso cuando los cambios sean al final de ciclo de la programación.

- *Potenciar al máximo el trabajo en grupo*: Todos los integrantes del grupo de trabajo, incluido el líder de proyecto, los desarrolladores y los clientes, son parte del equipo y están involucrados en el desarrollo del software para lograr el objetivo del proyecto.

3.3.3 Características de las metodologías ligeras

Las características principales de las metodologías ligeras son las siguientes:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.

- El software que es funcional es más importante que la documentación exhaustiva.

- La colaboración con el cliente es prioritaria que las negociaciones de contratos.

- Generación de propuestas, alternativas y soluciones para generar un cambio en lugar de seguir un plan cerrado o específico. [20]

3.4 Base de datos

La base de datos que se utiliza para el sistema cumple los requisitos para la primera, segunda y tercera forma normal.

3.4.1 Primera Forma Normal (1FN)

Es la más elemental de todas las formas normales, la cual establece que una tabla está en la 1FN si el valor que contiene un atributo de un registro, un campo, es único y elemental. En cada uno de los atributos sólo se puede incluir un dato, aunque sea compuesto, pero no se pueden incluir una lista de datos. [21]

- Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
- Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
- Cada columna debe tener un nombre único.
- Dos filas o renglones de una misma tabla no deben ser idénticas. [22]

3.4.2 Segunda Forma Normal (2FN)

Un atributo o conjunto de atributos tiene dependencia funcional de otro u otros si a cada uno de los primeros le corresponde sólo uno de los segundos.

Una tabla está en 2FN cuando está en 1FN y todo atributo que no pertenece a la clave primaria tiene una dependencia funcional de la clave completa y no de parte de ella. Si la clave principal está formada por un solo atributo y ya está en 1FN, ya estará en 2FN. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, está en segunda forma normal. [23]

3.4.3 Tercera Forma Normal (3FN)

Se dice que hay dependencia funcional transitiva entre dos atributos cuando un atributo que no pertenece a la clave primaria permite conocer el valor de otro atributo. [24]

Una tabla está en 3FN si está en 2FN y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forma parte de la clave primaria; no hay dependencias funcionales transitivas. [25]

3.5 Composición del sistema

La aplicación SIL se compone de 17 proyectos, de los cuales 16 proyectos se utilizan como bibliotecas para el proyecto principal (SIL). El proyecto principal contiene una referencia de las bibliotecas necesarias para que el sistema pueda ejecutarse y realizar las funciones deseadas, así también de las bibliotecas .Net y del sistema.

La estructura del sistema se conforma de la siguiente manera con sus bibliotecas:

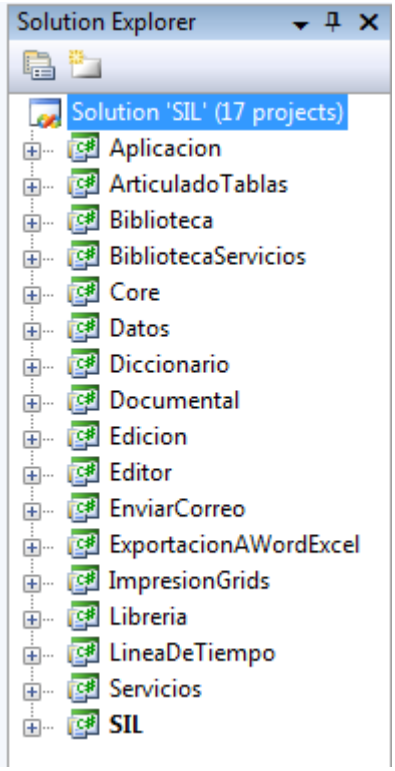


Figura 3.6 Aplicación 'SIL' (17 proyectos).

Cada biblioteca contiene sus propias clases con las que realizará alguna tarea en específico. El objetivo de la utilización de bibliotecas es con el motivo de evitar la duplicidad de código. La biblioteca 'BibliotecaServicios', la cual es la encargada de realizar la conexión a la base de datos es la más utilizada para la explicación anterior.

Existe una aplicación ('EnviaAlertas') totalmente independiente de la aplicación SIL, que tiene como referencia a la biblioteca 'BibliotecaServicios' y 'Servicios', la cual se generó como un proyecto de consola que utiliza clases de las bibliotecas anteriormente mencionadas, para realizar la conexión con la base de datos.

La aplicación se instaló en un servidor empleado en la SCJN, el cual está destinado a realizar tareas programadas, por lo que ejecuta la aplicación diariamente con la finalidad de enviar una alerta (correo electrónico) al realizar un cambio de vigencia de una reforma, únicamente se le envía la alerta a los usuarios afines según sus actividades laborales.

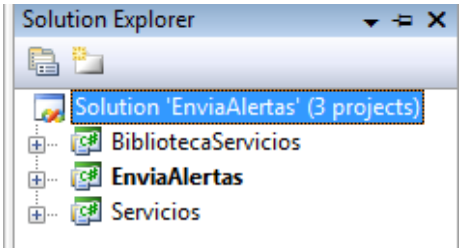


Figura 3.7 Solución 'EnviaAlertas' (3 proyectos).

A continuación se explicará cada una de las bibliotecas que utiliza el sistema para el funcionamiento correcto:

3.5.1 Aplicacion

Se encarga de almacenar en memoria la información respecto a la sesión actual, es decir guarda la información del usuario con el que se accede al sistema como el número de usuario, nombre, etc., mismos datos que van a ser utilizados posteriormente para acceso al sistema y validaciones de operaciones que puede realizar el usuario en dicha sesión.

También almacena en memoria la información necesaria para realizar la conexión a la base de datos, como el nombre del servidor o IP, usuario, contraseña, base de datos, etc. ('ConexionActiva'), misma que obtiene del archivo de configuración '*app.config*'.

Un ejemplo del archivo de configuración para la conexión activa se muestra en la siguiente figura:

```
<add name="ConexionActiva" connectionString="server=XXXX; User=XXXX; Password=XXXX; database=XXXX; Connect Timeout=0" />
```

Figura 3.8 Cadena de conexión, archivo app.config.

Define los directorios importantes para el sistema, donde se encuentran los archivos referentes al diccionario, las plantillas, archivos *XML* para realizar el '*Binding*', así como los directorios donde se encuentra la aplicación y sus archivos ejecutables.

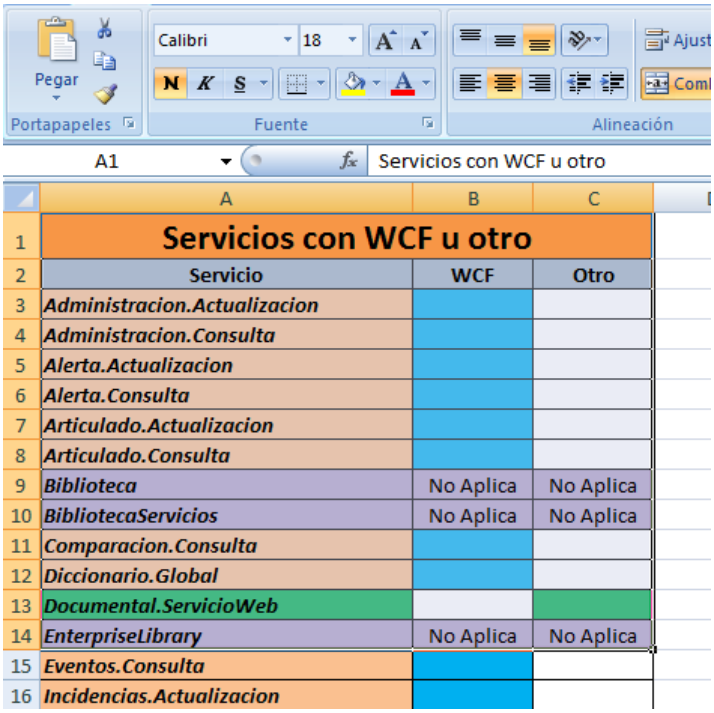
3.5.2 ArticuladoTablas

Se generó a partir de la necesidad de incluir tablas tipo Word o Excel, dentro del Articulado. La biblioteca contiene elementos *XAML* que son utilizados por *Windows Presentation Foundation (WPF)*.

La biblioteca 'ArticuladoTablas', se utiliza para guardar tablas en la base de datos, las cuales han sido copias de un archivo Word o Excel e insertadas en los componentes utilizados de WPF.

Para ejemplificar esta parte se describirá en breve el proceso de tablas en articulado de la siguiente forma:

- Se tiene una tabla en Word o Excel que se quiere copiar al articulado, única es necesario seleccionar la tabla y utilizar la función de copiar del portapapeles.



	A	B	C
1	**Servicios con WCF u otro**		
2	**Servicio**	**WCF**	**Otro**
3	*Administracion.Actualizacion*		
4	*Administracion.Consulta*		
5	*Alerta.Actualizacion*		
6	*Alerta.Consulta*		
7	*Articulado.Actualizacion*		
8	*Articulado.Consulta*		
9	*Biblioteca*	No Aplica	No Aplica
10	*BibliotecaServicios*	No Aplica	No Aplica
11	*Comparacion.Consulta*		
12	*Diccionario.Global*		
13	*Documental.ServicioWeb*		
14	*EnterpriseLibrary*	No Aplica	No Aplica
15	*Eventos.Consulta*		
16	*Incidencias.Actualizacion*		

Figura 3.9 Copiar a portapapeles.

- En el articulado se selecciona el tipo de artículo a 'Tipo Tabla' y se pega el contenido del portapapeles al componente de texto.

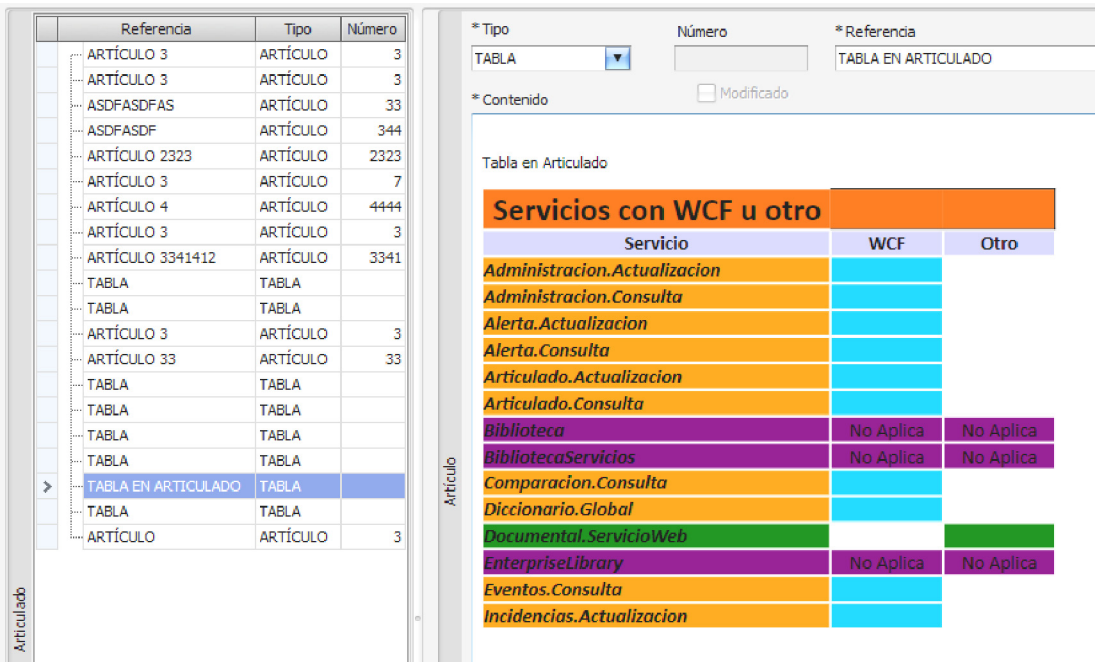


Figura 3.10 Pegar la tabla en el articulado.

- Posteriormente al guardar la tabla en la base de datos, se traduce el contenido del componente a etiquetas XAML, para así generar una cadena de caracteres y poder almacenar la tabla en un formato válido para SQL (nvarchar). Un ejemplo de la cadena de caracteres que se almacena en la base de datos es la siguiente:

```
<Section xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xml:space="preserve"
TextAlignment="Left" LineHeight="Auto" IsHyphenationEnabled="False" xml:lang="en-us"
FlowDirection="LeftToRight" NumberSubstitution.CultureSource="Text"
NumberSubstitution.Substitution="AsCulture" FontFamily="Tahoma" FontStyle="Normal" FontWeight="Normal"
FontStretch="Normal" FontSize="11" Foreground="#FF000000" Typography.StandardLigatures="True"
Typography.ContextualLigatures="True" Typography.DiscretionaryLigatures="False"
Typography.HistoricalLigatures="False" Typography.AnnotationAlternates="0"
Typography.ContextualAlternates="True" Typography.HistoricalForms="False" Typography.Kerning="True"
Typography.CapitalSpacing="False" Typography.CaseSensitiveForms="False" Typography.StylisticSet1="False"
Typography.StylisticSet2="False" Typography.StylisticSet3="False" Typography.StylisticSet4="False"
Typography.StylisticSet5="False" Typography.StylisticSet6="False" Typography.StylisticSet7="False"
Typography.StylisticSet8="False" Typography.StylisticSet9="False" Typography.StylisticSet10="False"
Typography.StylisticSet11="False" Typography.StylisticSet12="False" Typography.StylisticSet13="False"
Typography.StylisticSet14="False" Typography.StylisticSet15="False" Typography.StylisticSet16="False"
Typography.StylisticSet17="False" Typography.StylisticSet18="False" Typography.StylisticSet19="False"
Typography.StylisticSet20="False" Typography.Fraction="Normal" Typography.SlashedZero="False"
Typography.MathematicalGreek="False" Typography.EastAsianExpertForms="False"
Typography.Variants="Normal" Typography.Capitals="Normal" Typography.NumeralStyle="Normal"
Typography.NumeralAlignment="Normal" Typography.EastAsianWidths="Normal"
Typography.EastAsianLanguage="Normal" Typography.StandardSwashes="0" Typography.ContextualSwashes="0"
Typography.StylisticAlternates="0"><Section IsHyphenationEnabled="False"><Section
IsHyphenationEnabled="False"><Paragraph FontFamily="TimesNewRomanPS-BoldMT" FontWeight="Bold"
FontSize="17.333333333333332" Margin="0,0,0,0"><Run>UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO</Run></Paragraph></Section></Section></Section>
```

Figura 3.11 Código XAML del ejemplo de una tabla almacenada.

- Por otra parte, para recuperar la tabla original al realizar la consulta a la base de datos, se traduce el código XAML y se genera nuevamente la tabla almacenada.

Las tablas almacenadas a través del componente de WPF pueden ser editadas, copiadas o borradas. Sin embargo está limitado la edición de las tablas, ya que no permite agregar columnas adicionales, únicamente se limita a edición de texto visible, por otro lado se puede borrar la tabla original y ser reemplazada por otra.

3.5.3 Biblioteca

Fue diseñada para generar menús contextuales al presionar 'Click Derecho' del ratón. Debido a que algunos de los componentes de Visual Studio y en su mayoría los de *DevExpress* ya cuentan con su propio menú contextual, no fue necesario implementar la biblioteca en el sistema, sin embargo se conserva como referencia en caso de ser requerida para la realización de pruebas.

3.5.4 BibliotecaServicios

La importancia de esta biblioteca es que en ella se contiene las interfaces DAO, las clases SQLHelper, así como clases de envoltorios, entidades y clases necesarias para el manejo de excepciones y fallas de negocio, por lo cual está referenciada en la mayoría de los proyectos que conforman la aplicación.

Esta biblioteca se caracteriza por contar con una arquitectura por capas, debido a las interfaces DAO y DAL, mismas que sirven de intermediario entre el usuario y la conexión con la base de datos, de tal forma que el sistema no realiza una conexión directa con la base de datos.

La conexión con la base de datos se va a realizar por medio de esta biblioteca, así mismo las clases SQLHelper, son las encargadas del envío y recepción de las consultas a la base de datos. Se tienen dos clases principalmente:

- SQLHelperBase.cs, es aquella que obtiene la conexión activa a la base de datos.

- SQLHelper.cs, es aquella que realiza las consultas a la base de datos, preparando previamente los parámetros que van a ser utilizados por el *Stored Procedure* al cual hace referencia el sistema según sea el caso para realizar una tarea específica.

La biblioteca se diseñó con ayuda de las herramientas de Microsoft, la cual se hizo referencia a la biblioteca Microsoft.ApplicationBlocks.Data, para generar las clases y métodos necesarios para la conexión a la base de datos.

Se utiliza en conjunto de los Web Services (*Windows Communication Foundation, WCF*). La biblioteca se puede representar a grandes rasgos de la siguiente forma:

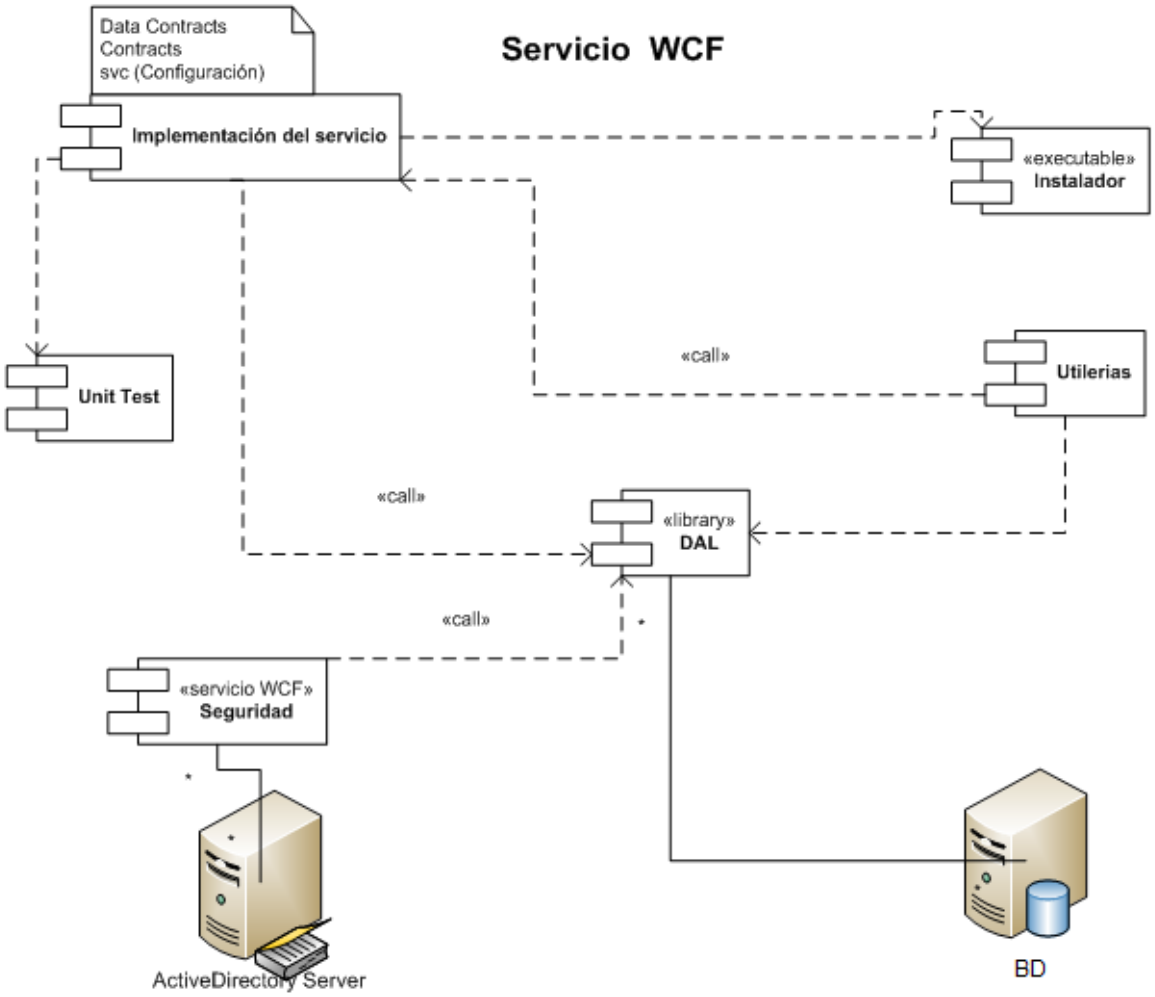


Figura 3.12 Conexión a la base de datos por medio de 'BibliotecaServicios'.

3.5.5 Core

Es la que obtiene la información por medio de los archivos XML. Los archivos XML contienen etiquetas para ser interpretadas y asignadas al *BindingSource*, mismo que se va a utilizar para generar las propiedades de los controles y componentes de Windows Forms, el cual proporciona una capa de direccionamiento indirecto al enlazar los controles de un formulario a los datos.

El componente BindingSource sirve como un origen de datos, de tal manera que interactúa con la información de los archivos XML, realizando un enlace de los controles del formulario al componente BindingSource. Las propiedades de los controles se encuentran definidas en las etiquetas de los archivos XML, únicamente se tiene que realizar llamadas al componente BindingSource para su implementación.

3.5.6 Datos

Su propósito principal fue considerada como la biblioteca que realizara la conexión con la base de datos, por otra parte se sustituyó por la biblioteca 'BibliotecaServicios', la cual realiza conexiones a la base de datos de manera más eficiente, actualmente ya no se utiliza la biblioteca 'Datos' pero se conserva para pruebas.

3.5.7 Diccionario

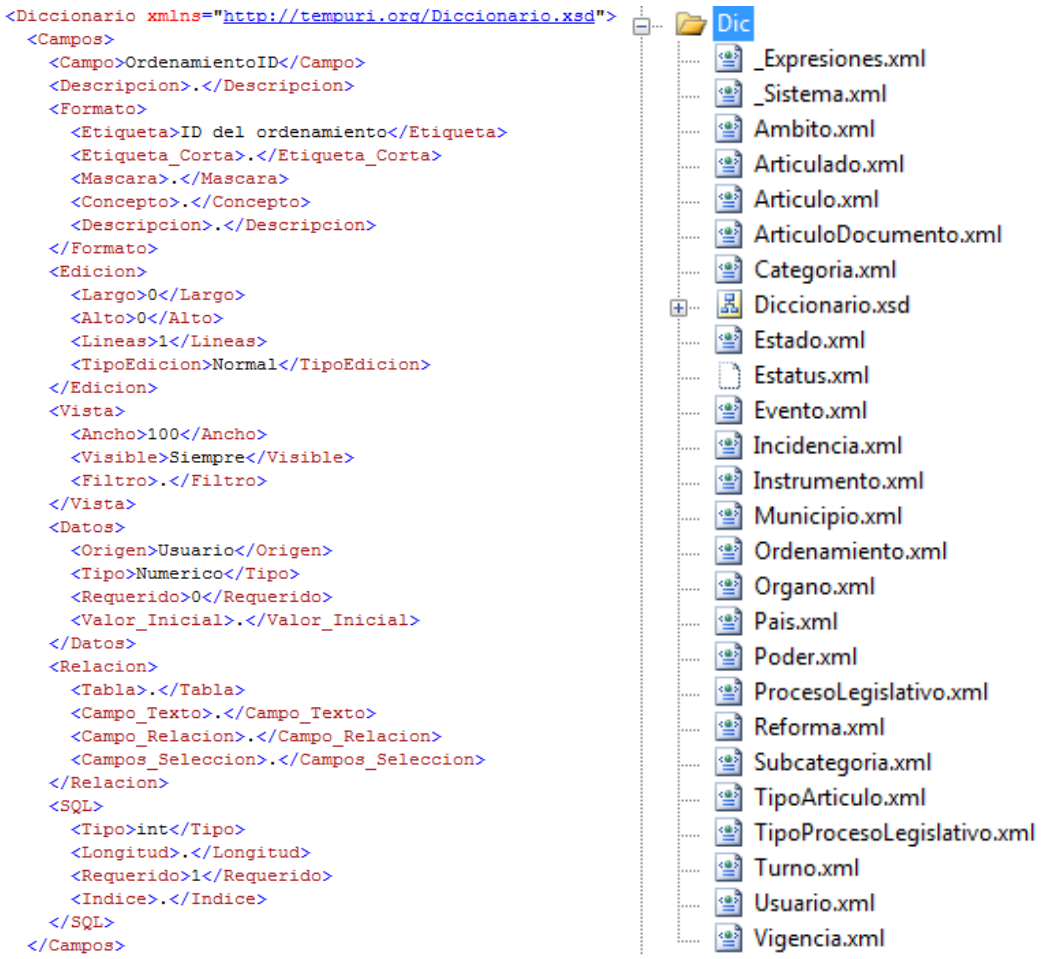
La biblioteca 'Diccionario', es la biblioteca que interactúa directamente con los archivos XML, definidos con el propósito de diseño en las formas realizadas manualmente sin necesidad de la ayuda de Visual Studio. Se definen las propiedades de cada uno de los controles que componen la interfaz principal del sistema, entre las propiedades que se tienen, son el tamaño del componente, habilitado o deshabilitado, tipo de componente, así como la localización en la interfaz principal.

La importancia de esta biblioteca en cuanto al diseño, es que se crea un objeto nuevo por cada componente que se desea implementar en la interfaz principal, cada objeto va a tener sus propias características, mismas que se deben especificar en los archivos XML.

Para agregar un nuevo componente, inicialmente se tienen que recurrir a las carpetas de 'Dic' y 'Dic_Instancias', que se encuentran dentro del proyecto 'SIL'. En la carpeta 'Dic', se definen las propiedades y características del componente a utilizar, como antes se mencionó, posteriormente en la carpeta 'Dic_Instancias' se tiene que agregar el nuevo componente a la interfaz principal, para ello es

necesario conocer la ubicación y definir las etiquetas, para que el sistema pueda interpretar dichos archivos y realizar la implementación.

Un ejemplo del código XML utilizado en 'Dic', es el siguiente:



```
<Diccionario xmlns="http://tempuri.org/Diccionario.xsd">
  <Campos>
    <Campo>OrdenamientoID</Campo>
    <Descripcion>.</Descripcion>
    <Formato>
      <Etiqueta>ID del ordenamiento</Etiqueta>
      <Etiqueta_Corta>.</Etiqueta_Corta>
      <Mascara>.</Mascara>
      <Concepto>.</Concepto>
      <Descripcion>.</Descripcion>
    </Formato>
    <Edicion>
      <Largo>0</Largo>
      <Alto>0</Alto>
      <Lineas>1</Lineas>
      <TipoEdicion>Normal</TipoEdicion>
    </Edicion>
    <Vista>
      <Ancho>100</Ancho>
      <Visible>Siempre</Visible>
      <Filtro>.</Filtro>
    </Vista>
    <Datos>
      <Origen>Usuario</Origen>
      <Tipo>Numerico</Tipo>
      <Requerido>0</Requerido>
      <Valor_Inicial>.</Valor_Inicial>
    </Datos>
    <Relacion>
      <Tabla>.</Tabla>
      <Campo_Texto>.</Campo_Texto>
      <Campo_Relacion>.</Campo_Relacion>
      <Campos_Seleccion>.</Campos_Seleccion>
    </Relacion>
    <SQL>
      <Tipo>int</Tipo>
      <Longitud>.</Longitud>
      <Requerido>1</Requerido>
      <Indice>.</Indice>
    </SQL>
  </Campos>
```


Figura 3.13 Elementos XML, Dic.

De la misma forma un ejemplo para ‘Dic_Instancias’:

```
<Consulta Campo="OrdenamientoID" Tabla="Ordenamiento" Modo="Edicion" Sector="Ordenamiento" Grupo="" Esquema=""> </Consulta>
```

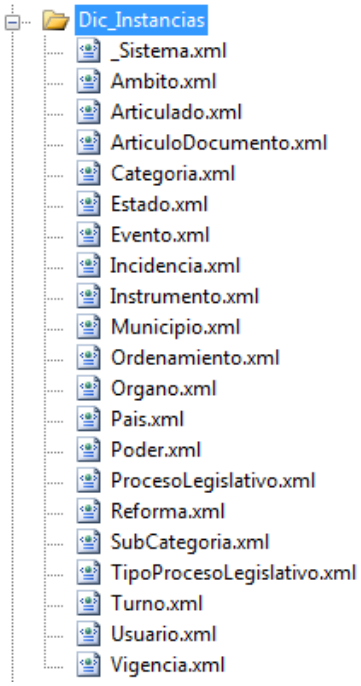


Figura 3.14 Elementos XML, Dic_Instancias.

3.5.8 Documental

Contiene las clases necesarias para generación de documentos en Word de carácter oficial, los cuales contienen un listado de las reformas de un ordenamiento y el listado de los artículos contenidos en el articulado de una reforma. Los documentos se generan en formato ‘.docx’ para ser manipulados con Microsoft Office Word.

La biblioteca que se utilizan para la generación de dichos documentos es DocumentFormat.OpenXml, en la cual utiliza la clase DocumentFormat.OpenXml.Wordprocessing, para generar el nuevo documento con el mismo formato que la plantilla utilizada. Se apoya de otras clases de OpenXml para generar la documentación.

Esta biblioteca está encargada de realizar las consultas pertinentes a la base de datos, dependiendo del listado de reformas o articículos, y posteriormente plasmar la información al documento Word.

3.5.9 Edicion

Tiene por objetivo definir los controles de los elementos utilizados en el sistema, define las acciones que deben realizar algunos elementos, como en el caso de que un campo de texto quede vacío, se muestra un error mostrando que el campo está vacío. En términos generales se definen las acciones de los controles de algunos componentes del sistema.

Así mismo la biblioteca es quien realiza las llamadas al BindingSource, encargándose de ligar la información con los componentes, adicionalmente realiza validaciones para asegurar que los componentes van a tener un funcionamiento adecuado. Se apoya de la biblioteca 'Diccionario' para instanciar objetos que van a tener información referente a los componentes y así poder definir sus atributos según el componente utilizado. Se definen propiedades adicionales de los controles, como es el caso del diseño visual (skin), entre otras propiedades.

El mayor aporte que brinda esta biblioteca, son los controles de ortografía, definidos por las clases SpellCheckerDictionaryBase, propios de los componentes DevExpress, definidos en DevExpress.XtraSpellChecker. Esta biblioteca permite que los componentes asociados puedan contar con una revisión de ortografía. Para obtener el diccionario de palabras se hace uso del diccionario de OpenOffice y aparte el diccionario propuesto por los administradores del sistema SIL, conocido como el 'Diccionario Global'.

Las dos clases referentes a la revisión de ortografía son:

- *LocalDic.cs*: La cual va a servir para definir los objetos y asociar los diccionarios: Local, Global y OpenOffice. Esta clase se encarga de cargar los datos en memoria.

- *Corrector.cs*: Es la clase que permite asociar un componente con los diversos diccionarios con los que se cuenta. Siempre que un componente requiera de una revisión ortográfica se tiene que referir a esta clase.

Los errores ortográficos se subrayan automáticamente en rojo, así mismo el menú contextual mostrará las palabras propuestas que puedan ajustarse a la palabra erróneamente escrita.

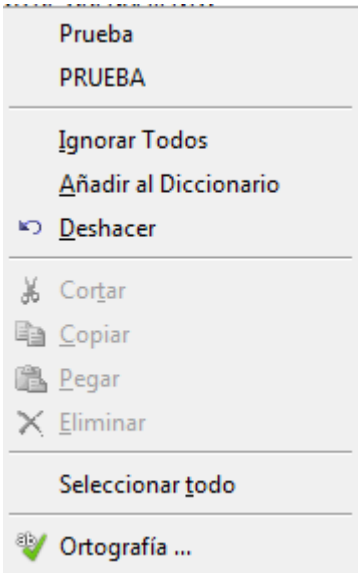


Figura 3.15 Menú contextual del corrector ortográfico.

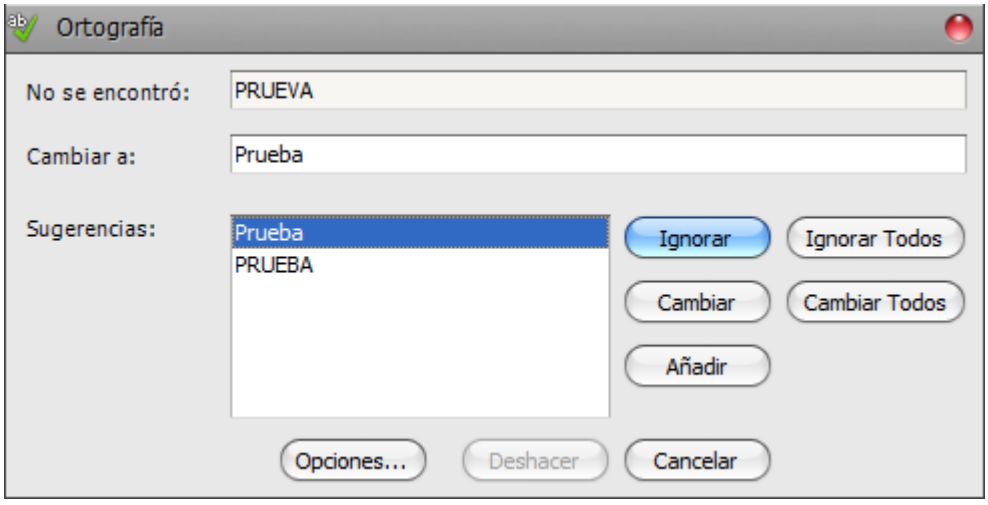


Figura 3.16 Corrector ortográfico.

En conjunto con las bibliotecas de revisión de ortografía, se implementó el 'Diccionario Global', el cual consiste en un diccionario independiente, el cual es administrado por ciertos usuarios encargados de agregar palabras permitidas en el sistema. Las palabras adicionales, pueden ser propuestas por los usuarios que utilizan el sistema o por el mismo administrador. Este conjunto de palabras propuestas están sujetas a validación del administrador, en el cual se tienen cuatro posibles escenarios:

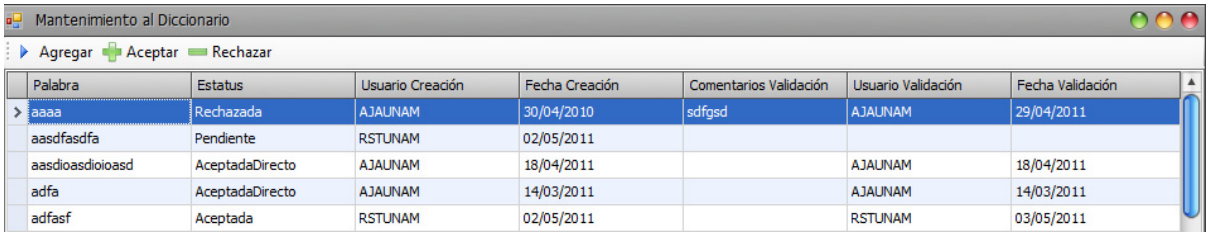
- *Aceptada*: Cuando un usuario propone una palabra para agregar al diccionario y el administrador la cataloga como una palabra válida.

- *AceptadaDirecto*: Cuando el administrador ingresa alguna palabra al diccionario, por defecto es catalogada como válida.

- *Rechazada*: Cuando un usuario propuso una palabra para el diccionario y el administrador la cataloga como una palabra no válida.

- *Pendiente*: Cuando un usuario propone una palabra y el administrador aún no la ha catalogado.

El mantenimiento al diccionario global se realiza desde la aplicación SIL, contando con un perfil de administrador.



Palabra	Estatus	Usuario Creación	Fecha Creación	Comentarios Validación	Usuario Validación	Fecha Validación
aaaa	Rechazada	AJAUNAM	30/04/2010	sdfgsd	AJAUNAM	29/04/2011
aasdfasdfa	Pendiente	RSTUNAM	02/05/2011			
aasdioasdioioasd	AceptadaDirecto	AJAUNAM	18/04/2011		AJAUNAM	18/04/2011
adfa	AceptadaDirecto	AJAUNAM	14/03/2011		AJAUNAM	14/03/2011
adfasf	Aceptada	RSTUNAM	02/05/2011		RSTUNAM	03/05/2011

Figura 3.17 Mantenimiento al diccionario

3.5.10 Editor

Contiene controles para los componentes que aceptan texto o números. Realiza validaciones según el tipo definido para evitar conflictos de pérdida de precisión. Entre las validaciones los controles definidos como numéricos en los componentes de texto, únicamente aceptan números, por otra parte algunos controles de tipo texto convierten todo el texto a mayúsculas y los controles para fechas, se les aplica una máscara para visualizarse completamente la fecha o en algún formato especial, esto depende de la definición establecida en los archivos XML que utiliza la biblioteca 'Diccionario'.

La biblioteca 'Editor' es en forma general similar a la biblioteca 'Edicion'. Actúan como bibliotecas de apoyo para no tener la necesidad de repetir código para cada control.

3.5.11 EnviaCorreo

Es la única biblioteca que interactúa con otros usuarios de forma indirecta. Se emplea para enviar correos electrónicos informativos sobre cambios en la fecha de publicación de una reforma. Estos correos electrónicos exclusivamente serán enviados a las personas afines y relacionadas con el ámbito y perfil.

Esta biblioteca se diseñó con el fin de ser utilizada por la solución aparte 'EnviaAlerta', misma que utiliza estas clases para el envío de correos electrónicos.

La diferencia en su implementación de esta biblioteca entre la solución 'EnviaAlerta' y la solución 'SIL', está definida que en la solución 'EnviaAlerta' se procede a enviar un correo electrónico a los usuarios afines con el ámbito y perfil, al realizarse un cambio en la vigencia de una reforma, por el otro lado en la solución 'SIL', se realiza el envío de correos cuando una reforma cambia su fecha de publicación.

Los correos electrónicos son enviados desde el servidor *SMTP* (Simple Mail Transfer Protocol) de la SCJN, donde el remitente es una cuenta de correo destinada al sistema, de la misma forma se envía una copia oculta a otra cuenta de correo, para corroborar la legitimidad y prevenir el *no repudio*.

3.5.12 ExportacionAWordExcel

Genera documentos Word o Excel con la información de los ordenamientos y reformas marcados para exportar, su principal función es generar documentación para mantenerla digital o impresa, según se requería.

A diferencia de la biblioteca 'Documental', en la cual se utilizaba OpenXML para la generación de documentos, en esta biblioteca se utilizan las bibliotecas de 'Microsoft.Office.Interop' para la generación de los documentos Word o Excel. Para la generación de estos documentos no se utilizó una plantilla, por lo cual la generación completa del documento se realizó a través de código.

La generación del documento es en forma de tabla, debido a que se utiliza la información contenida en los resultados de una consulta a la base de datos, posteriormente se le asigna formato para mostrar su contenido en el documento. Se selecciona previamente la información que se quiere mostrar por medio de un Windows Forms, donde muestra las diferentes columnas de información para mostrar.

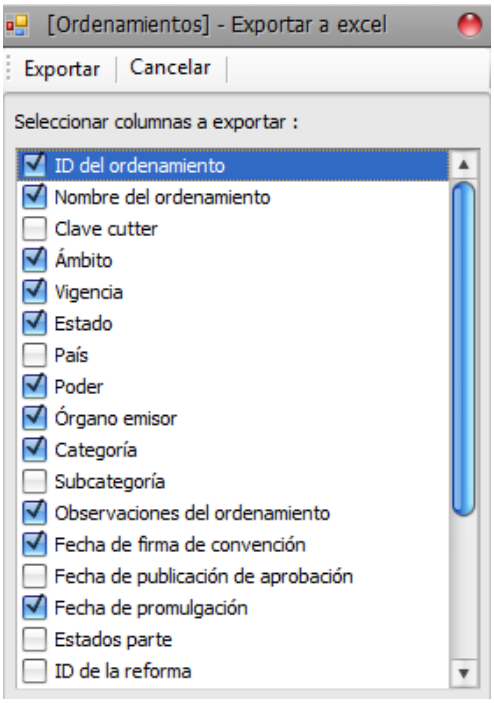


Figura 3.18 Exportación a Word o Excel.

3.5.13 ImpresionGrids

Su función principal es generar documentos Word para su impresión, fue diseñada de igual manera que la biblioteca 'ExportacionAWordExcel', la cual genera sus documentos con las bibliotecas de 'Microsoft.Office.Interop'. Los documentos generados se almacenan en el disco duro para su posterior uso.

A diferencia con la exportación a Word de la biblioteca 'ExportacionAWordExcel', es que el orden de la información puede ser acomodada en la tabla para la impresión. Se utiliza un Windows Forms similar, con botones que permiten acomodar el orden de las columnas a mostrarse, de igual manera se puede seleccionar las columnas deseadas para la impresión.

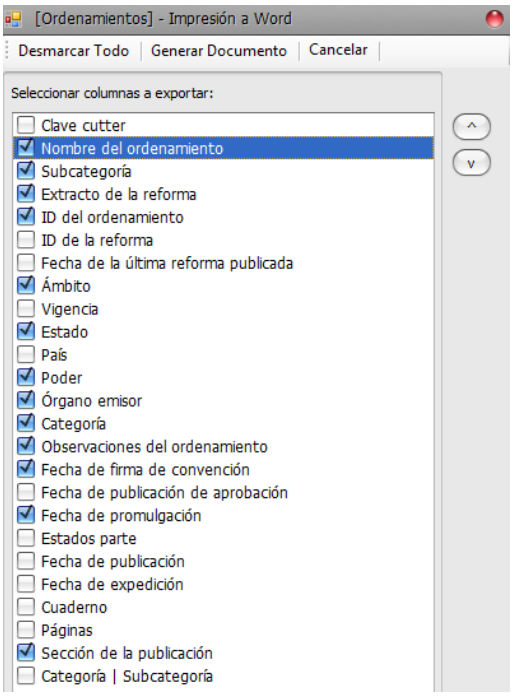


Figura 3.19 Impresión ordenamiento.

3.5.14 Libreria

Su funcionalidad es definir componentes. La interfaz principal utiliza estos componentes, cada componente contiene sus propios controles para realizar una funcionalidad del sistema, generalmente se utilizan para mostrar la información, como es el caso de los componentes de texto, también define componentes de paneles para contener otros elementos. Existen componentes que interactúan con el usuario final, como es el caso de los botones, hipervínculos, barras de herramientas, menús contextuales, etc.

La interfaz principal que utiliza esta biblioteca se define de la siguiente manera:

- *Detalle*: Muestra la información en el módulo de ordenamientos.
- *Vista*: Muestra la información en las tablas al realizar consultas.
- *Edición*: Muestra la información al realizar una edición.
- *Consultas*: Muestra la información del ordenamiento.

Un ejemplo de estos controles es el detalle de la interfaz principal, la cual está definida por estos componentes.

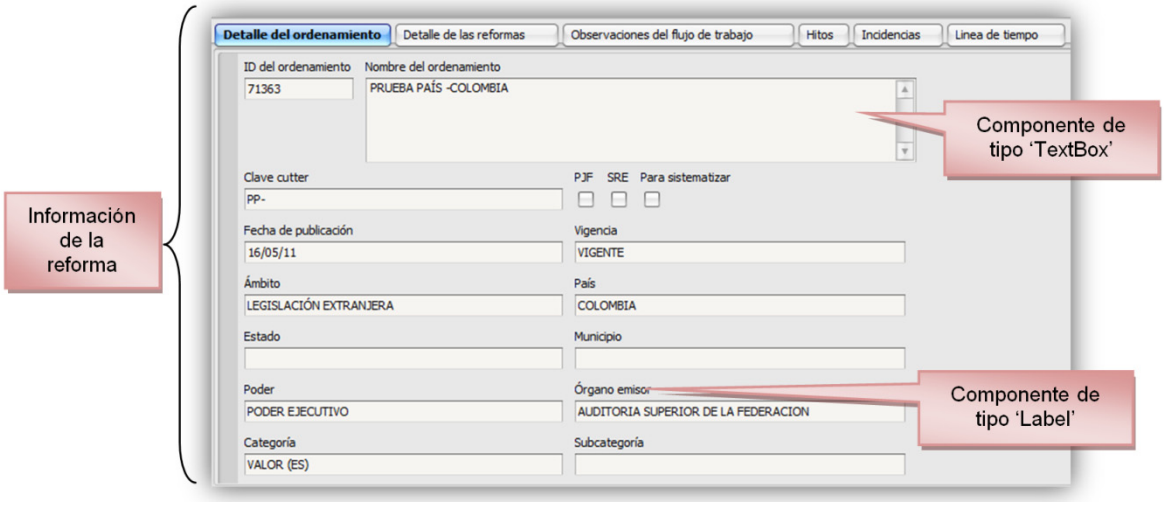


Figura 3.20 Interfaz principal, detalle.

Para las columnas mostradas en una tabla, componente GridView, se definen columnas especiales como los hipervínculos o botones, a los cuales se definen controles especiales para realizar alguna acción específica. También se definen controles para cada tipo de columna, como es el caso de las columnas que tienen valores numéricos, fechas, etc. El sustento de la información visual para el usuario, recae en esta biblioteca que es indispensable para el sistema.

Categoría	Extracto de...	Observaciones	PDF	DOC	Articulado
CONSTITUCION	CONSTITUCI...	VEASE: 1136		00130029.doc	Ver
FE DE ERRATAS	FE DE ERRA...				
DECRETO	DECRETO Q...			00130136.doc	Ver

Figura 3.21 Columnas de tipo texto e hipervínculo.

Adicionalmente la biblioteca contiene las imágenes e íconos que se utilizan en el sistema para darle una mayor estética visual con el usuario. Se encuentran definidos todas las imágenes utilizadas, así como imágenes e íconos que pudieran ser de utilidad para el diseño visual con el usuario.

3.5.15 LineaDeTiempo

Se encarga de generar una línea de tiempo para la reforma. Se muestran aquellos sucesos importantes de la reforma, desde su creación. La línea de tiempo se define en esta biblioteca la cual es referenciada para poder utilizarse, utiliza elementos de WPF para la generación de la misma.

Su principal objetivo es poder tener conocimiento y corroborar los momentos en la que dicha reforma ha ocurrido un punto crítico.

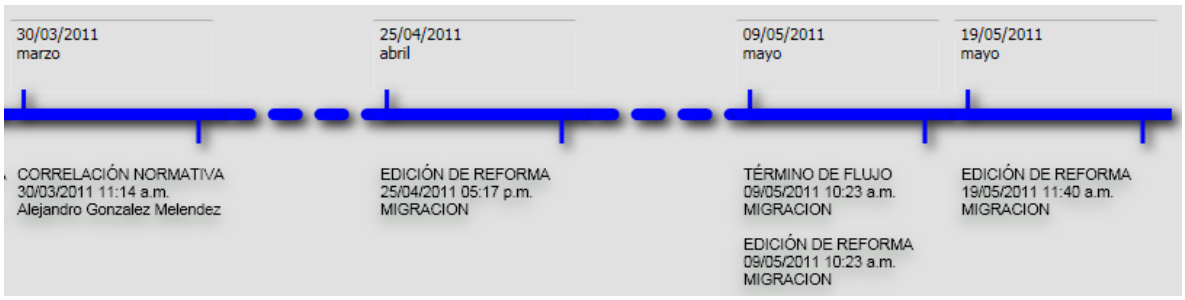


Figura 3.22 Línea de tiempo.

3.5.16 Servicios

Contiene clases para realizar conexiones con la base de datos, en la que se utiliza Ado.Net, para obtener la información. Se apoyaban en estas clases previamente a la utilización de los Web Services, mismos que fueron removidos.

3.5.17 SIL

Es el proyecto principal, contiene el método main, con el cual va a iniciar la aplicación. La aplicación inicia con la clase Inicio.cs, donde se invocan todos los componentes para generar la interfaz principal.

El proyecto contiene todas las clases adicionales a los proyectos biblioteca para poder utilizar el sistema. Los módulos se cargan a partir de estas clases, mismas que utilizan las bibliotecas incluidas en la solución para realizar una acción específica.
El proyecto se divide en tres módulos importantes:

- Ordenamientos.
- Consulta General.
- Consulta Avanzada.

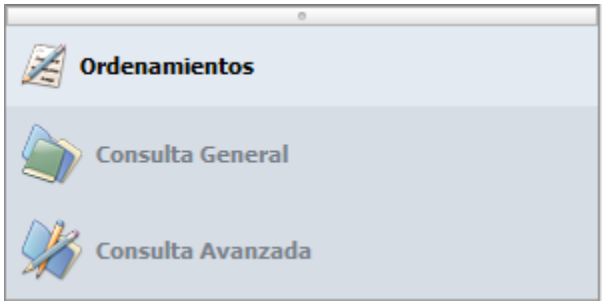


Figura 3.23 Módulos.

3.6 Módulos relacionados

Los tres módulos importantes que conforman al SIL se describen a continuación:

3.6.1 Módulo de Ordenamientos

Se muestra la información referente a la reforma, es el módulo principal en el cual se pueden observar las reformas que están en el flujo de trabajo. Así como las operaciones que se pueden realizar dependiendo del perfil que se esté utilizando.

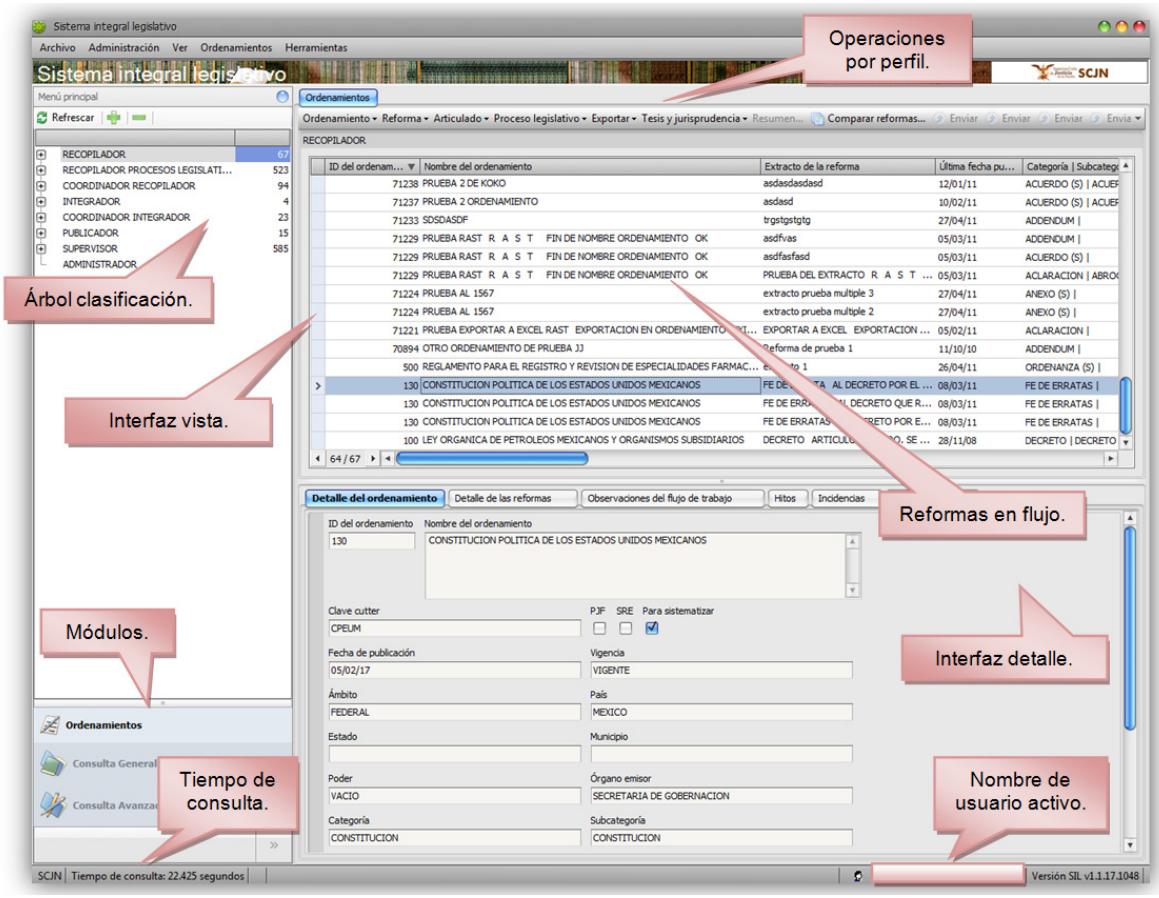


Figura 3.24 Módulo de ordenamientos.

- *Árbol de clasificación*: Dependiendo de perfil que el usuario tenga, es el número de nodos con los que va a contar dicho árbol. Para el ejemplo mostrado es un usuario con todos los perfiles. Este árbol es de utilidad para mostrar el número de reformas separadas por ámbito, así como al seleccionar un nodo del árbol muestra únicamente las reformas pertenecientes al ámbito.

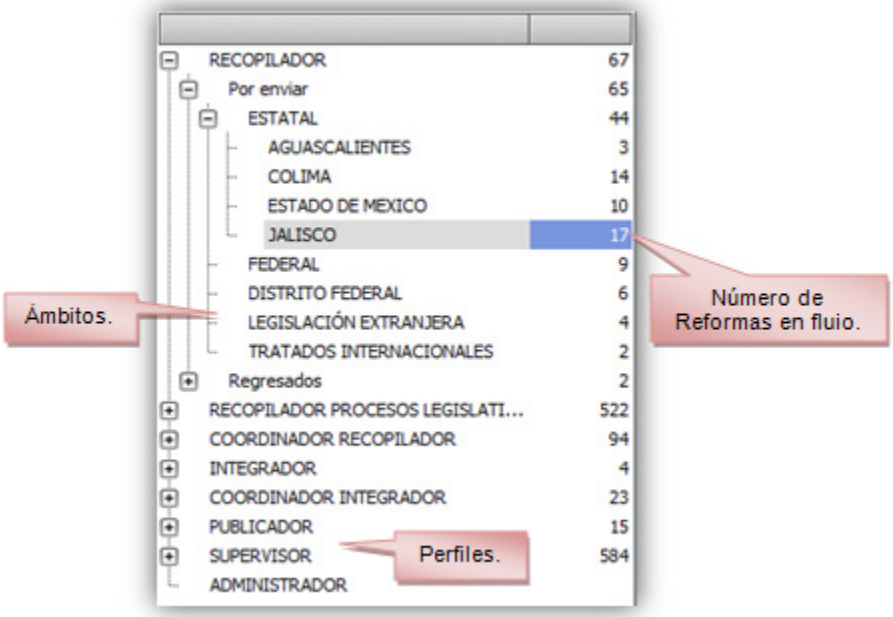


Figura 3.25 Árbol de clasificación.

- *Reformas en flujo*: Son aquellas reformas que se muestran dependiendo del nodo seleccionado en el árbol. Estas reformas le pertenecen al usuario activo y se puede realizar alguna operación dependiendo del perfil.

- *Nombre de usuario activo*: Es el nombre de red con el que el sistema autentica al usuario para utilizar el sistema. (Se censuró el nombre de usuario activo).

- *Operaciones por perfil*: Son aquellas acciones que el usuario activo puede realizar a alguna reforma. Cada perfil tiene definidas sus propias acciones, así como los coordinadores pueden realizar las mismas acciones que sus subordinados.

- *Interfaz vista*: Es la interfaz donde se muestran las reformas en flujo, se utilizaron componentes 'DevExpress' para su visualización en forma de tabla, en la cual se pueden seleccionar una o varias reformas.

- *Interfaz detalle*: Se muestra la información referente a la reforma seleccionada, los detalles de esta interfaz ya han sido mencionados en este documento.

- *Tiempo de consulta*: Es el tiempo en segundos que tarda el servidor SQL en realizar una consulta.

3.6.2 Módulo de Consulta General

La consulta general, es una búsqueda de reformas por el número identificador de su ordenamiento o se pueden realizar búsquedas por medio de alguna palabra o frase en específico.

Tiene por objetivo facilitar la búsqueda de información cuando no se tiene claro la información a buscar. A través de la consulta general es una forma de acceder a los datos almacenados en el acervo.

A través de esta consulta se puede realizar las siguientes operaciones:

- Editar ordenamientos.

- Editar reformas que se encuentran en el acervo.

- Editar procesos legislativos de reformas que se encuentran en el acervo.

- Editar el articulado de reformas que se encuentran en el acervo.

- Comparar reformas.

- Imprimir el resultado de la búsqueda.

- Exportar a Excel y a Word.

Para ejemplificar la consulta general se realizó una consulta por ID del Ordenamiento y una consulta por Nombre:

- *Consulta por ID:*

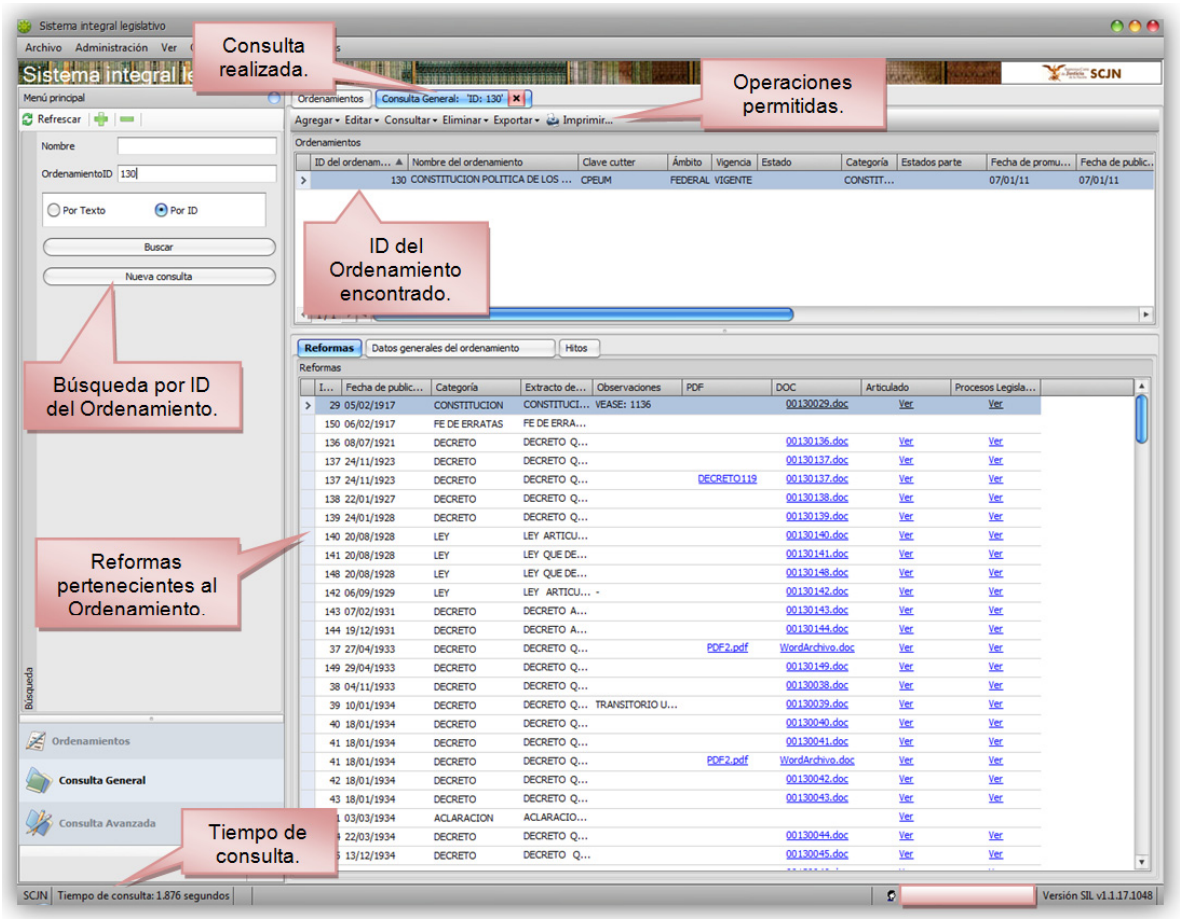


Figura 3.26 Módulo de consulta general, consulta por ID.

- *Búsqueda por ID*: Se introduce el ID referente al ordenamiento a buscar, en caso de existir muestra el ordenamiento con sus respectivas reformas, por el contrario muestra un mensaje de que no existe dicho ordenamiento.

- *Barra de consulta*: En caso de tener varias consultas en el sistema, la barra de consulta muestra el ID de la consulta a la cual le pertenece y así desplazarse a las diferentes búsquedas.

- *Consulta por nombre:*

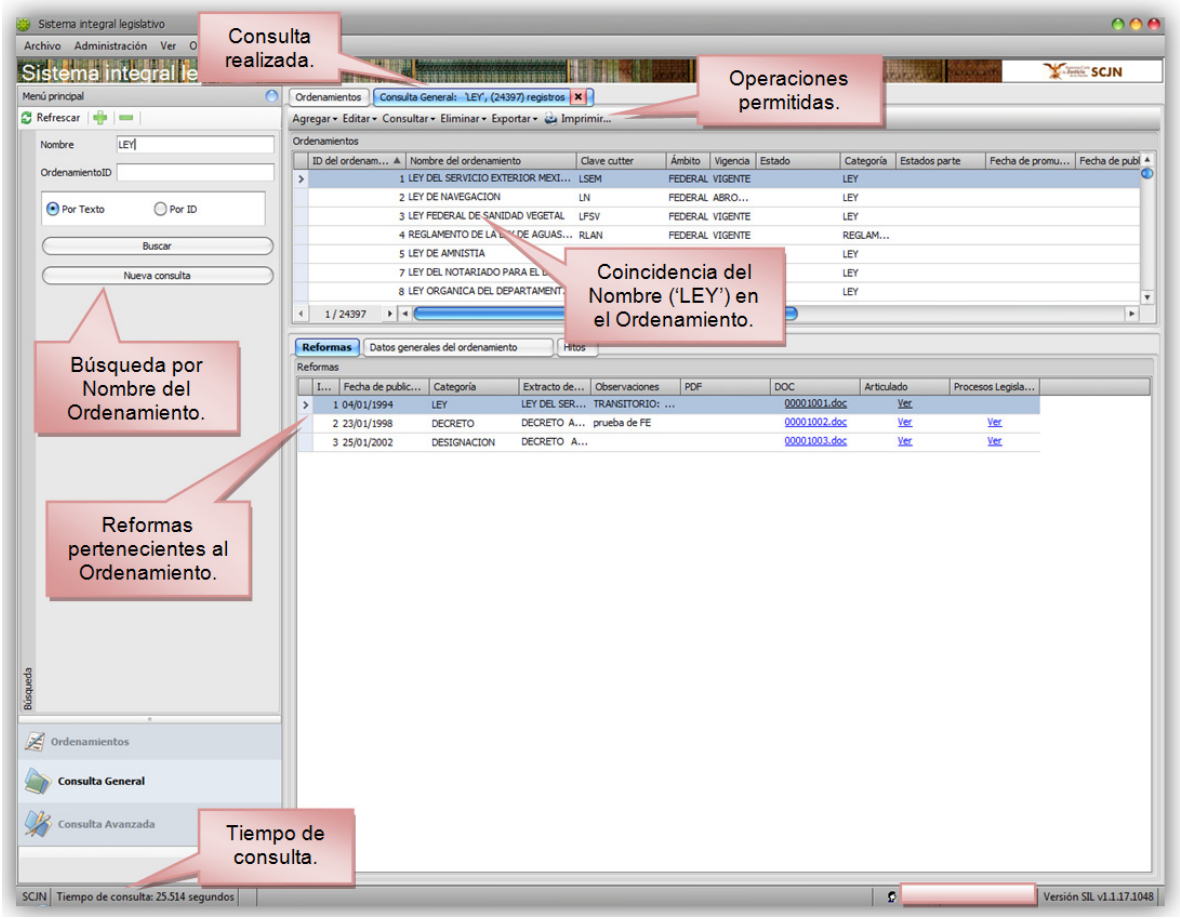


Figura 3.27 Módulo de consulta general, consulta por Nombre.

A diferencia de la búsqueda por ID, la búsqueda por nombre puede encontrar uno o más registros, así como el tiempo de consulta es mayor en la búsqueda por nombre, debido a que consulta todos los ordenamientos registrados en la base de datos. La búsqueda por nombre, se puede realizar a través de una palabra o frase.

3.6.3 Módulo de Consulta Avanzada

La consulta avanzada, es una búsqueda de reformas más específica, en la cual se puede generar uno o más resultados de la búsqueda, dependiendo de los parámetros que se establecen para la búsqueda.

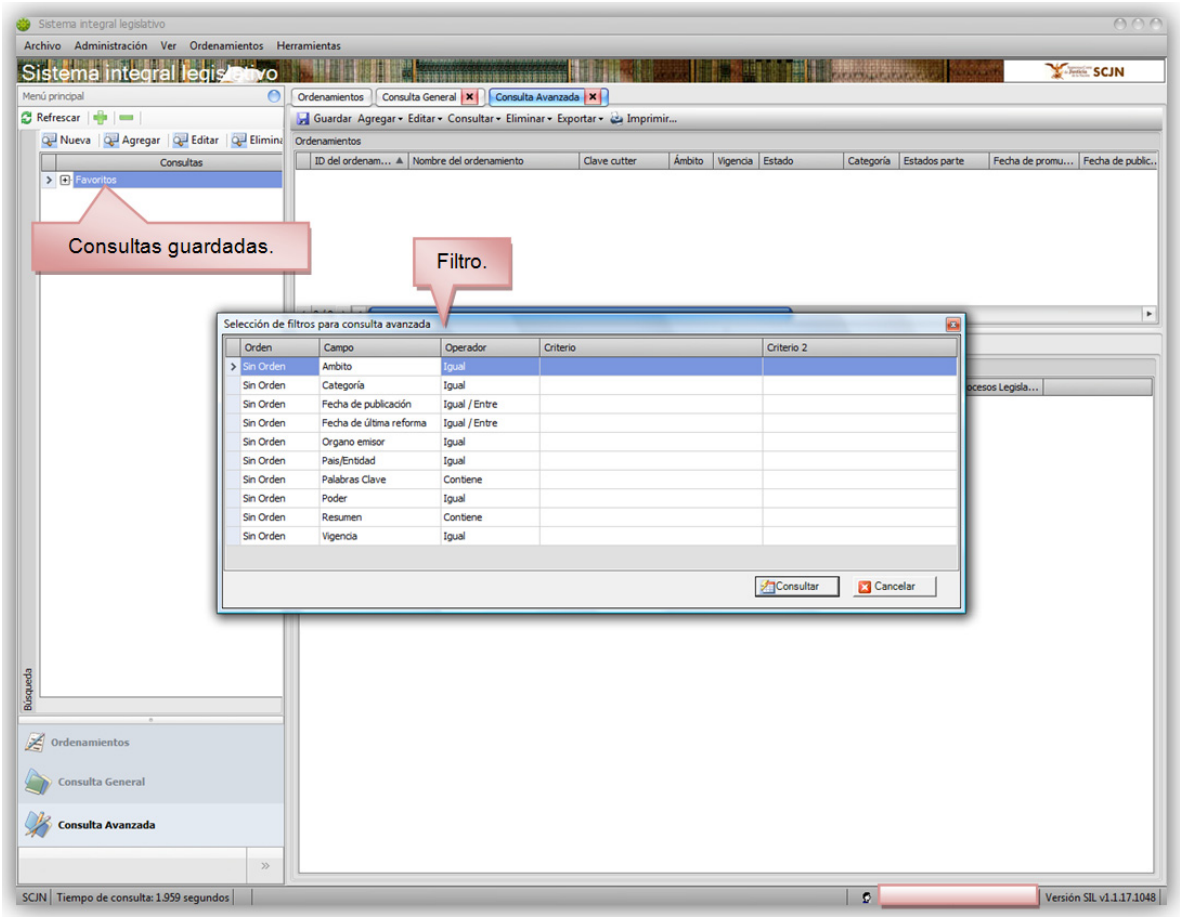


Figura 3.28 Módulo de consulta avanzada.

A diferencia de la consulta general, la consulta avanzada es más específica por el filtro de búsqueda. Así mismo la consulta avanzada puede tardar mucho más tiempo que la consulta general, debido a los parámetros del filtro, los cuales tiene que comparar en todos los ordenamientos.

Filtro: Para realizar una búsqueda avanzada deben utilizarse parámetros de búsqueda que le permitirán simplificar la forma en la que ingresa los criterios de búsqueda.

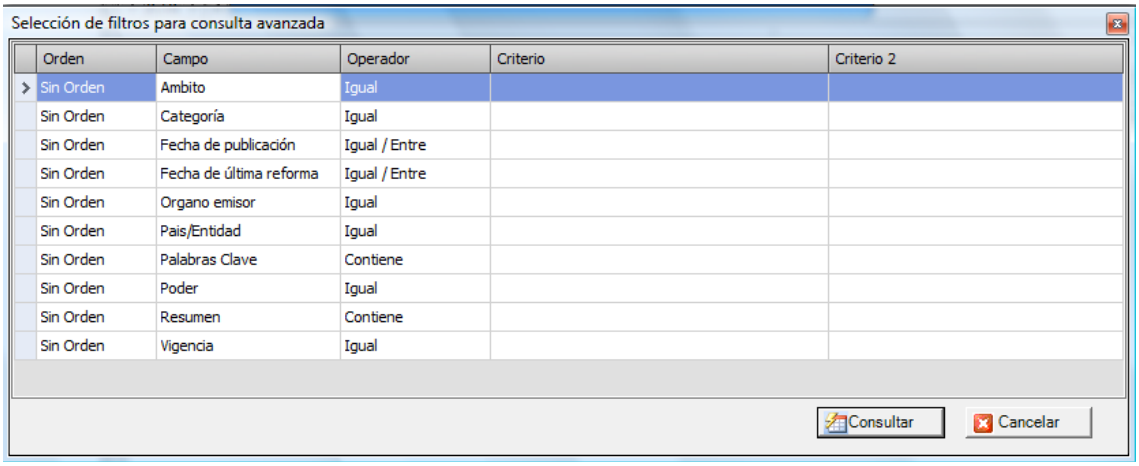


Figura 3.29 Filtro de consulta avanzada.

La descripción de los parámetros de búsqueda se encuentra definido en la siguiente tabla:

Tabla 3.1 Parámetros de búsqueda.

Columna	Valores permitidos	Descripción
Orden	*Ascendente*	Se utiliza para que el sistema ordene la información por el campo seleccionado.
	Descendente	
	Sin Orden	
Campo	*Palabras clave*	Es el listado de campos que el usuario podrá llenar de acuerdo a los datos que tenga para realizar la búsqueda. No es necesario que el usuario conozca todos y cada uno de los campos, podrá llenar aquellos datos que conozca.
	Categoría	
	Vigencia	
	Ámbito	
	País / Entidad	
	Municipio	

	Poder *Órgano emisor* *Fecha de publicación* *Fecha de última reforma* *Resumen*	
Operador	-	Clausulas sobre las cuales el sistema estará realizando la búsqueda dentro del campo seleccionado.
Criterio	-	Palabras o fechas que introduzca o seleccione el usuario para filtrar la información.

Consultas guardadas: Para agilizar las búsquedas frecuentes, se pueden almacenar en la base de datos, con ello no es necesario volver a realizar la consulta extensa. Las consultas guardadas tienen como propósito ahorrar tiempo y recursos del servidor de la base de datos.

3.7 Flujo de Reformas y perfiles

Cada perfil definido en el sistema participa en el flujo de las reformas , éste flujo comienza en el recopilador el cual captura los ordenamientos y reformas; envía a su coordinador de recopiladores; éste revisa la información reportada como trabajada y envía al coordinador de integradores quien a su vez turna a sus integradores con base en el ámbito y estado que cada integrador vaya a trabajar; éstos dan mantenimiento al articulado y envían al coordinador de integradores (para revisión) quien envía al publicador con la intención de informarle cuales han sido los registros que se trabajaron y que están disponibles para su publicación.

Durante la ejecución del flujo, puede ocurrir uno o varios regresos de registros del coordinador de integradores al coordinador de recopiladores ó de los coordinadores (recopilador e integrador) a sus subordinados (recopiladores e integradores) ó del publicador a los coordinadores (recopilador e integrador) para corregir algún dato incorrecto.

Los coordinadores (recopilador e integrador) pueden reasignar registros a sus elementos de trabajo.

El perfil de recopilador de procesos legislativos está presente en cualquier parte del flujo general.

El coordinador de recopiladores (cuando recibe un ordenamiento marcado como "No sistematizar") y el publicador pueden terminar el flujo de una reforma.

El flujo general se muestra en la siguiente figura:

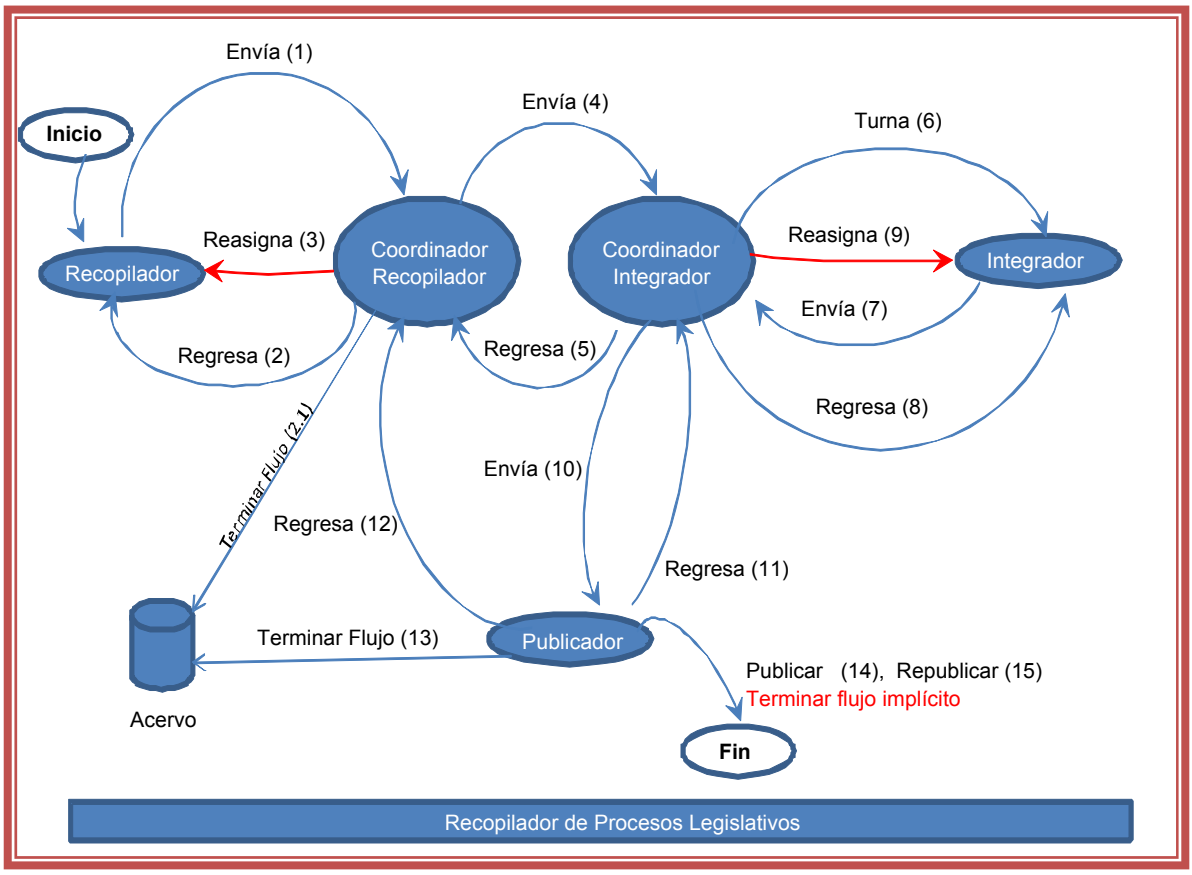


Figura 3.30 Flujo de la reforma.

3.8 Participación profesional

Mi participación profesional en el desarrollo del sistema se basó en casos de uso a resolver. Los cuales fueron asignados por el analista y el arquitecto, entre ellos se destacan los siguientes:

Tabla 3.2 Bitácora de actividades

Caso de uso	Descripción
1736	El formulario que realiza la función de dar de alta ordenamientos y reformas (‘FormAltaOrdenamiento.cs’), requería de validaciones para evitar que algunos campos obligatorios sean ‘nulos o vacíos’. Se requería de ajustes al formulario para mostrar el texto correctamente al incrementar el tamaño de letra. Las correcciones realizadas al formulario fueron de diseño y de control. No se realizó desarrollo de alguna funcionalidad nueva.
1845	Se realizaron correcciones en la programación para la generación de los documentos para impresión que contenían información de las reformas, tales correcciones correspondieron al orden de impresión de las columnas mostradas por cada uno de los campos que componen a la reforma.
1915	A todos los componentes ‘GridView’ únicamente muestran los primeros 50 caracteres del texto que contenga, debido a que algunos textos son muy extensos. El componente se utiliza como consulta previa.
1919	Se modificó la biblioteca de ‘ExportacionAWordExcel’, para que generara los documentos de manera similar a los que son para impresión. Se hace uso de plantillas para su generación.
M-1	Debido a la utilización de los servicios web (WCF), el tiempo de la consulta a la base de datos incrementó de manera notable, por lo que se pidió la remoción de los servicios, dejando las clases necesarias para realizar una consulta a la base de datos directamente sin tener que utilizar el servicio web. Debido a esta mejora se tuvieron que modificar todas las clases de la

	aplicación que requerían una consulta a la base de datos por medio de algún servicio web. Como beneficio de la remoción de los servicios web, el tiempo de consulta a la base de datos se redujo notablemente.
0248	Al realizar una consulta avanzada, el tiempo de consulta excedía los 5 minutos, por lo que se realizaron las correcciones pertinentes al procedimiento almacenado para que la consulta fuera precisa y eficiente. Se creó un nuevo procedimiento almacenado para generar una tabla temporal en lugar de realizarla en una función que itera 'n veces el número de registros', con el objetivo de disminuir la carga de trabajo y así reducir el tiempo de la consulta.
1957	Se generaron hipervínculos dentro de un componente 'GridView', en el cual al presionar el hipervínculo se muestra el documento generado del articulado completo de la reforma seleccionada. El documento se genera por medio de las bibliotecas DocumentFormat.OpenXML.
1902	De igual forma con el caso de uso 1957, se generó un hipervínculo que al ser presionado muestra el proceso legislativo de la reforma seleccionada.
1913	Se corrigió la biblioteca 'DiccionarioGlobal' para que actualice los diccionarios locales acorde la base de datos, donde contiene las palabras aceptadas o rechazadas. Así mismo agregar palabras a la base de datos desde el sistema, posteriormente el administrador validará las palabras y se les asignará un estatus.
1758	Se creó en el articulado un nuevo tipo, el cual corresponde a la inserción de tablas de Excel, Word, etc. Dichas tablas se insertan en el articulado y posteriormente se guardan en la base de datos. Utiliza la biblioteca 'ArticuladoTablas' para guardar la información en la base de datos.

1744	Con motivo del caso de uso 1758 de guardar las tablas insertadas en la base de datos. El documento de Word correspondiente al articulado completo de la reforma, requiere de la inserción de dichas tablas en el mismo documento. Se utilizaron las bibliotecas 'DocumentFormat.OpenXml' y 'Microsoft.Office.Interop.Word'.

Capítulo 4. Resultados

4.1 Logros del sistema

El sistema se construyó de acuerdo a la estructura y tecnología propuesta en el análisis del sistema.

Por medio del sistema se lleva un adecuado control de las obras en materia jurídica y a través de la Subdirección de Compilación Documental ésta información es integrada para su disposición del personal del Poder Judicial de la Federación como del público en general para su consulta.

Se optimizaron los tiempos de respuesta al momento de ingresar y de recuperar la información para la integración de los ordenamientos jurídicos y para su cotejo, así como para su disposición de consulta.

Se actualizó la información para la publicación en Intranet, Internet y en la edición de los discos compactos que la Suprema Corte de Justicia de la Nación pone para su venta al público en general.

4.2 Pendientes del sistema

Toda funcionalidad solicitada por la SCJN fue implementada en tiempo y forma, no existen pendientes de desarrollo acordado en dicho convenio. Sin embargo mientras los empleados utilizan el sistema se ha generado un listado con nuevas funcionalidades útiles para desarrollar.

Acorde al convenio, el periodo de garantía no cubrirá el desarrollo de nuevas funcionalidades por lo que es estrictamente para la corrección de fallas de las funcionalidades ya existente.

4.3 Pruebas realizadas

Las pruebas de software las realizaron los “*testers de software”*, en el que se revisó íntegramente el sistema de manera exhaustiva para corroborar que no existan errores de programación y de diseño, se comprobaron todas las funcionalidades solicitadas. Las pruebas se realizaron previas a la entrega de proyecto.

Posteriormente para el periodo de garantía los empleados de la SCJN que utilizarán el sistema son quienes realizarán las pruebas de manera indirecta mientras evalúan el sistema, dando como resultado posibles fallas.

4.4 Entrega de proyecto

La entrega del proyecto se realizó en la fecha solicitada, por lo que no existieron sanciones por un retraso debido a algún problema inesperado. Se realizó una junta con los directores de área para una demostración de la implementación de una nueva funcionalidad con carácter demostrativo. También se hizo una revisión de diversas funcionalidades importantes del sistema, finalmente se evaluó el sistema para su aprobación, misma que fue exitosa.

Se generaron las órdenes de trabajo referentes al proyecto para su aprobación y firma de entrega, así como la documentación sobre el sistema solicitada para dicha entrega.

4.5 Fases posteriores

Como parte de las fases posteriores referente al Sistema Integral Legislativo, se contó un periodo de garantía, en el cual se quedó encargado un programador para solucionar cualquier error de software que surgiera durante dicho periodo. El periodo comprendido es del 1° de Junio al 31 de Agosto del 2011.

Al finalizar el convenio con la UNAM (UNAM / 26131-841-13-V-10), se entregó en su totalidad el código del sistema y documentación generada. Posterior a esta entrega no existe ninguna fase relacionada con la UNAM.

La SCJN por medio del convenio 2011 con la Universidad Autónoma del Estado de México, quien dará seguimiento al desarrollo, mejoras y correcciones de ambos sistemas (SIL y SIJ).

Conclusiones

El Sistema Integral Legislativo, es una herramienta que servirá de gran utilidad para los trabajadores de la Suprema Corte de Justicia de la Nación en lo referente a la compilación de los ordenamientos, reformas, articulados y procesos legislativos, ya que a través del sistema tendrán toda la información disponible para consultar o actualizar en forma digital, así también cuenta con la posibilidad de generación de documentos para impresión, de la misma forma el sistema recopilará la información para ser modificada según lo escrito en el diario oficial, la gaceta o cualquier otro medio donde se publiquen los cambios que haya sufrido un ordenamiento a nivel reforma, artículo o proceso legislativo. Dicha información será procesada, revisada, corregida y resguardada por quien utiliza el sistema, con el fin de tener actualizada toda la información y estar disponible al público en general vía Internet, accediendo a la página web de la SCJN.

El sistema fue diseñado y creado a partir de los objetivos que se plantearon por la SCJN, así mismo fue adecuado a las necesidades de los usuarios para realizar un mejor sistema con todas sus funcionalidades solicitadas. La SCJN tiene por objetivo final del sistema el recopilar toda la información existente de todas las leyes existentes en México, así como Tratados Internaciones y Legislaciones Extranjeras que conciernen a México o que se encuentra directamente involucrado en dicha Ley, la importancia de esta recopilación es para ser publicada de forma electrónica vía Internet. El desarrollo y diseño del sistema está orientado a todas aquellas necesidades para hacer posible el cumplimiento del objetivo de la SCJN, en la cual se tiene involucrado una aplicación de escritorio, la cual será la entrada de la información y como salida de la misma, una página web donde se mostrará la información públicamente.

El sistema fue adaptado para tener una similitud con el sistema que antiguamente se utilizaba para la recopilación de la información, el diseño se implementó de manera similar para que a los usuarios que anteriormente utilizaban ese sistema no tuvieran muchas dificultades para asimilar el nuevo sistema. Sin embargo, debido a todas sus funcionalidades y que cada usuario puede tener uno o más perfiles asociados, el sistema presenta complicaciones debido al tráfico de red generado por todos los empleados de la SCJN, los balanceadores de carga deben solventar este problema de tráfico en la red para poder tener de manera eficiente en cuanto a tiempos de respuesta el SIL. Como anteriormente se mencionó, el sistema es en aspectos generales muy robusto, por lo que yo considero que debería realizarse una separación por módulos, así se mejorarían los tiempos de respuesta debido a que no toda la información tendría que ser cargada en memoria, ni todos los módulos, así también se tendría una fácil localización de funcionalidades específicas al perfil asociado, esto con la idea de una rápida y fácil asimilación del sistema con los nuevos empleados de la SCJN que nunca antes habían utilizado el sistema.

En beneficio de la ecología, el digitalizar la información impresa y la información que aún no ha sido impresa se ahorrarán recursos naturales necesarios para la impresión de dicha información. El sistema tiene por objetivo conservar la información en forma digital sin tener que recurrir a los textos impresos como se hace actualmente. A su vez la SCJN ahorrará recursos monetarios al no imprimir dichos documentos, de cualquier forma se puede imprimir dicha información en caso de ser necesario a través del sistema, la cual en muchas de las ocasiones se requiere de un par de copias y solo de algunas secciones, de esta forma bastaría una sola impresión por cada grupo de trabajo y no una copia por cada empleado, ya que la información se encontraría de forma digital y puede ser consultada las veces que sea necesario, por otro lado los archivos PDF y Word de la gaceta o del diario oficial pueden ser descargados y así obtener una copia digital de los mismos.

Mi participación en este proyecto me ayudó en la formación profesional como Ingeniero en el que tuve que utilizar mis conocimientos y aptitudes para realizar las necesidades del sistema así como solucionar y mejorar los problemas que surgieron a lo largo del desarrollo del proyecto. Incrementé mis aptitudes y conocimiento en el área de software, a la par de la obtención de experiencia laboral, la cual servirá para incremento del currículum y así poder conseguir un mejor trabajo de acuerdo a mis expectativas.

Glosario

Acervo. Conjunto de bienes electrónicos o en papel que le ha sido donado a la Suprema Corte de Justicia de la Nación por instituciones diversas y el cual está bajo resguardo de la Dirección de Compilación de Leyes con la intención de ponerlo a disposición del público en general para su consulta.

Ado.Net. Es un conjunto de clases que exponen servicios de acceso a datos para el programador de .NET. Constituye una parte integral de .NET Framework y proporciona acceso a datos relacionales, XML y de aplicaciones. [26]

Ámbito. Es la clasificación del ordenamiento, y esta puede ser: Estatal, Distrito Federal, Federal, Legislación Informática, Tratados Internacionales.

Articulado. Conjunto de artículos que conformar a la reforma. Se exponen las partes en las que se divide la reforma, en forma numerada o por categoría.

Binding. Proporciona un nivel de acceso a la definición de un enlace, que conecta las propiedades de los objetos de destino (por lo general, los elementos de WPF), y cualquier fuente de datos (por ejemplo, una base de datos, un archivo XML, o cualquier objeto que contiene datos). [27]

BindingSource. Encapsula un origen de datos para enlazarlo a controles. Proporciona una capa de direccionamiento indirecto al enlazar los controles de un formulario a los datos. Actúa como un origen de datos con establecimiento inflexible de tipos. [28]

Casas de la Cultura Jurídica (CCJ). Oficinas ubicadas en los diferentes estados de la República Mexicana cuya misión es brindar consulta del acervo impreso (legislación y expedientes) al público en general.

Cuadernillo. Conjunto de copias del Diario Oficial de la Federación, o de la Gaceta Oficial o del Periódico Oficial de los Estados en donde se ven reflejados todas las reformas que se le han hecho a un ordenamiento y que posteriormente se pone a disposición del personal de la Suprema Corte de Justicia de la Nación para su préstamo o consulta en sala por medio del área de Servicio al Público.

Decreto. Disposición o resolución dictada por una autoridad en asuntos de su competencia. Ejemplos: creación, reforma, adición, abrogación, etc.

Diario Oficial de la Federación (DOF). Órgano del Gobierno Constitucional de los Estados Unidos Mexicanos, que tiene la función de publicar en el territorio nacional: leyes, reglamentos, acuerdos, circulares, órdenes y demás actos expedidos por los poderes de la Federación, a fin de que éstos sean observados y aplicados debidamente en sus respectivos ámbitos de competencia.

Digitalización. Proceso de escanear las hojas de la publicación oficial en donde se observa el extracto de la reforma.

Enterprise Resource Planning (ERP). Un ERP es un sistema de información integral que incorpora los procesos operativos y de negocio. El propósito fundamental de un ERP es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación. [29]

Extracto de reforma. Texto explicativo que indica el elemento del ordenamiento ha sido reformado.

Fe de erratas. Tipo de decreto que se hace posterior a la publicación de una reforma con la intención de enlistan los errores que aparecieron en la edición previa.

Historia Legislativa. Cronología de reformas que ha sufrido un ordenamiento.

Hito. Acción o hecho clave dentro de un ámbito o contexto. En el caso particular del sistema, un hito define una acción importante que se le ha aplicado al ordenamiento a lo largo del tiempo.

ID.

- Identificador del ordenamiento: Número único para identificar un ordenamiento en el sistema.
- Identificador de la reforma: Número consecutivo para identificar una reforma de un ordenamiento.

IUS. Sistema que recopila las *jurisprudencias* y tesis aisladas que publica el Semanario Judicial de la Federación y su Gaceta y que cual es puesto a la venta de la población en general o consultado a través de la red interna y de Internet.

Jurisprudencia. Es el conjunto de los fallos de los tribunales judiciales que sirven de precedentes. Todas las sentencias conforman la jurisprudencia.

No repudio. Servicio de seguridad (OSI ISO-7498-2) que previene que un emisor niegue haber remitido un mensaje (cuando realmente lo ha emitido) y que un receptor niegue su recepción (cuando realmente lo ha recibido). [30]

Ordenamiento. El término se utiliza como sinónimo de ley, reglamento, código u ordenanza, en el entendido, y en un sentido amplio, se refiere a ordenar e integrar todas las normas de un sistema jurídico para conocer el lugar que corresponde a cada norma jurídica y a cada conjunto de normas dentro de la unidad.

Periódico Oficial de los Estados (POE). Órgano de información de los Gobiernos Locales que tiene como función publicar las Leyes, Decretos, Reglamentos, Acuerdos, Circulares, Notificaciones, Avisos y demás actos expedidos por los tres órdenes de gobierno, en sus respectivos ámbitos de competencia, así como las actas, documentos o avisos de particulares que conforme a la Ley, deban de ser publicados o tengan interés en hacerlo.

Proceso legislativo. Es el proceso de creación de una disposición jurídica a través del poder legislativo.

Red Jurídica. Sistema en el cual se publica o controla información que no necesariamente es dada a conocer en el portal de Internet de la SCJN

Reforma. Cambios y mejoras realizados al ordenamiento, contiene un texto explicativo que indica el elemento del ordenamiento ha sido reformado, llamado extracto.

SCJN. Suprema Corte de Justicia de la Nación

Semanario Judicial de la Federación y su Gaceta (SJFG). Publicación impresa de la SCJN en donde es dado a conocer al público en general el texto de las tesis o jurisprudencias aisladas que se emiten en la SCJN.

Simple Mail Transfer Protocol (SMTP). Es un estándar para el envío de correo electrónico a través de Internet o una red IP. Es el protocolo más utilizado para el envío de correo electrónico saliente de un servidor a otro. [31]

Stored Procedure. Los procedimientos almacenados y funciones son nuevas funcionalidades de la versión de MySQL 5.0. Es un conjunto de comandos SQL que pueden almacenarse en el servidor. Una vez que se hace, los clientes no necesitan relanzar los comandos individuales pero pueden en su lugar referirse al procedimiento almacenado. [32]

Tomos. Volumen encuadernado que contiene un grupo de Diarios o periódicos oficiales organizados por fechas.

Tesis aislada. Tesis es el género, tesis aislada es un solo criterio y tesis jurisprudencial son cinco resoluciones en un mismo sentido sin ninguna en contrario (especie). Es el criterio jurisdiccional respecto a una figura jurídica que soporta una sentencia.

Testers. Son aquellas personas encargadas de las pruebas de software, en cuya actividad está dirigida a la evaluación de un atributo o una capacidad de un programa o sistema y la determinación de que cumple con sus resultados requeridos. [33]

Web Services. Permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores. [34]

Windows Presentation Foundation (WPF). Es un sistema de presentación, para crear aplicaciones cliente de Windows que proporcionen una experiencia impactante para el usuario desde el punto de vista visual. [35]

XAML. El lenguaje de marcado de aplicaciones extensible es un lenguaje de marcado basado en XML desarrollado por Microsoft. XAML es el lenguaje que se usa para la presentación visual de las aplicaciones desarrolladas con Microsoft Expression Blend, del mismo modo que HTML es el lenguaje que se usa para la presentación visual de las páginas Web. Forma parte de Microsoft Windows Presentation Foundation (WPF). [36]

XML. Es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. [37]

Anexos

Anexo A. Framework .NET

Es una infraestructura de desarrollo que está compuesta por diversos recursos, entre los cuales se destaca el más importante, que es una máquina virtual conocida como CLR (Common Language Runtime), sobre la cual se ejecutan las aplicaciones. De este modo, los programas ya no poseen código nativo de ningún microprocesador en particular, sino instrucciones MSIL (Microsoft Intermediate Language) que son traducidas a código nativo en el momento de su ejecución (por medio de un compilador Just In Time).

El framework también define una librería base de clases, BCL (Base Class Library), a la cual puede acceder cualquier desde lenguaje desarrollado para la plataforma. Por encima de la infraestructura se ubica un conjunto de reglas básicas que debe implementar un lenguaje para poder ser parte de la familia .NET. Esta especificación es conocida como CLS (Common Language Specificeation). [38]

Arquitectura del Framework .Net

El código fuente escrito en C# se compila en un lenguaje intermedio (IL) conforme con la especificación CLI. El código de lenguaje intermedio, junto con recursos tales como mapas de bits y cadenas, se almacena en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll generalmente. Un ensamblado contiene un manifiesto que ofrece información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Cuando se ejecuta un programa de C#, el ensamblado se carga en CLR, con lo que se pueden realizar diversas acciones en función de la información del manifiesto. Posteriormente CLR realiza una compilación Just In Time (JIT) para convertir el código de lenguaje intermedio en instrucciones de máquina nativas.

CLR también proporciona otros servicios relacionados con la recolección automática de elementos no utilizados, el control de excepciones y la administración de recursos. El código ejecutado por CLR se denomina algunas veces "código administrado", en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico. [39]

En el diagrama siguiente se muestran las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C#, las bibliotecas de clases base, los ensamblados y CLR:

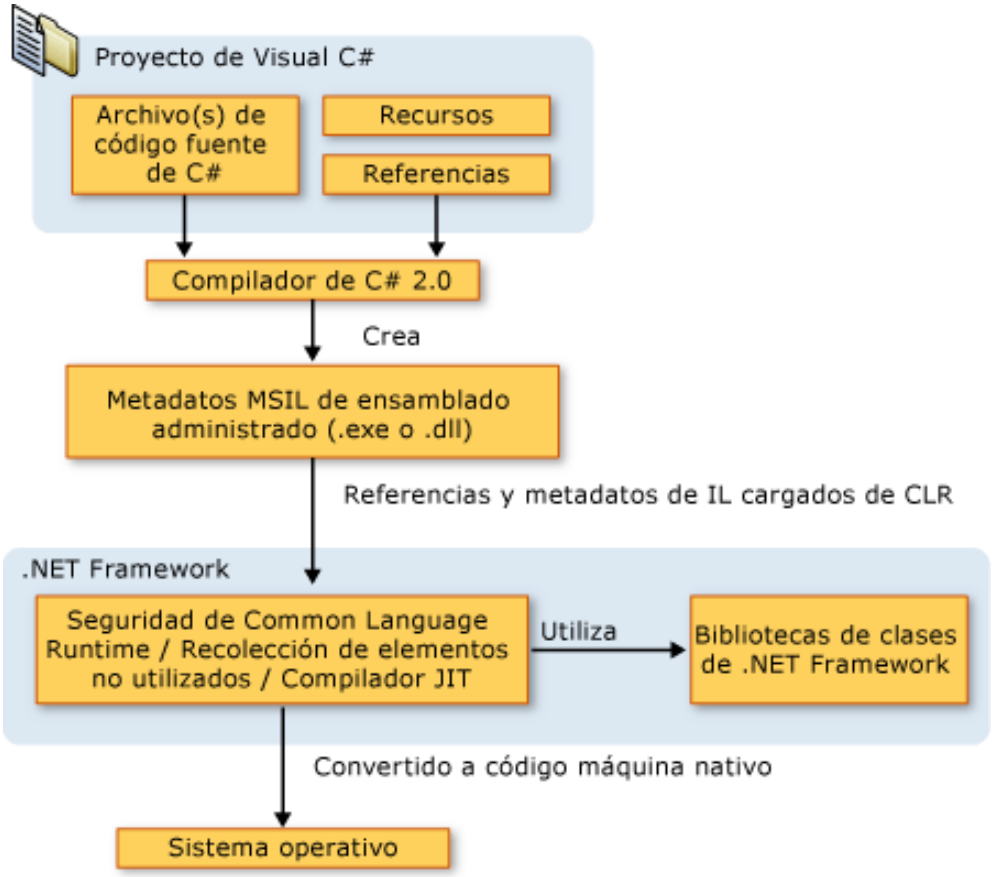


Figura A.1 Framework .Net.

Finalmente, se encuentra el conjunto de lenguajes que cumplan con la especificación CLS, como C#, VB.NET, C++, etc. [40]

En forma general la arquitectura de la infraestructura de desarrollo se muestra en la siguiente imagen:

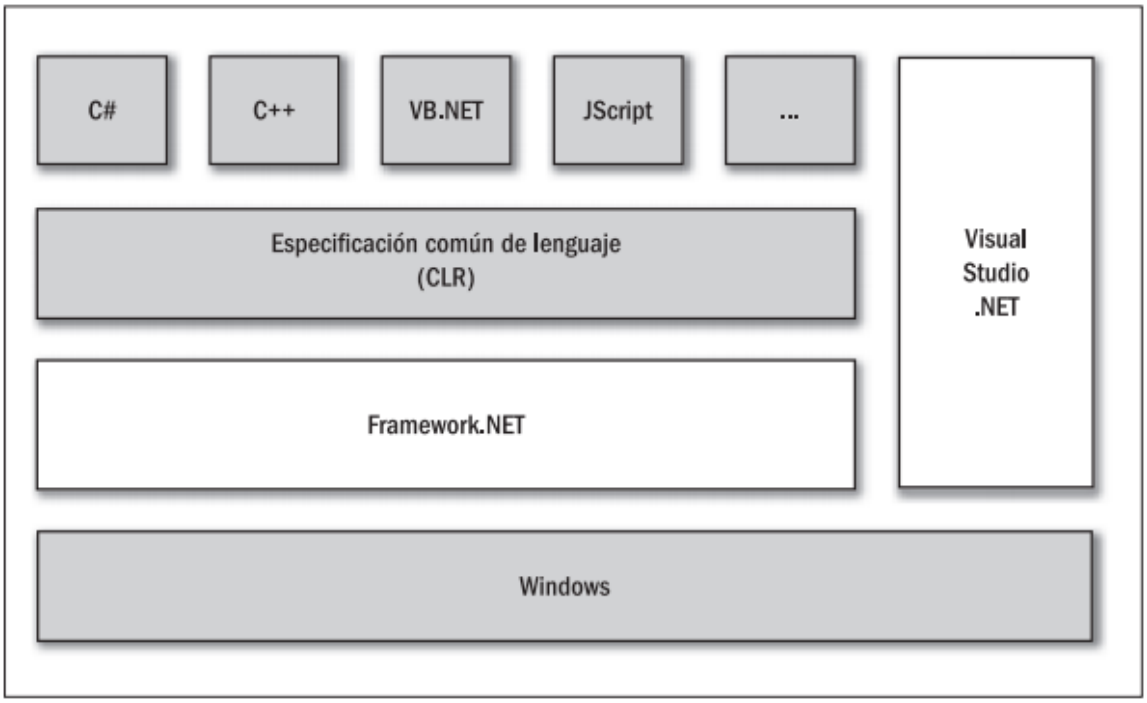


Figura A.2 Arquitectura de la infraestructura de desarrollo. [41]

Anexo B. Lenguaje C#

C# (C Sharp) es un lenguaje moderno y altamente expresivo que se ajusta al paradigma de programación orientada a objetos. Su sintaxis es similar a C++ y Java. El lenguaje fue desarrollado en gran parte por Anders Hejlsberg (reconocido por ser uno de los diseñadores líder del lenguaje de programación Delphi). [42]

C# es un lenguaje que cumple con la especificación del Framework .NET, en el cual el código creado es traducido a instrucciones MSIL para ser traducidas a código nativo no de manera monolítica, sino por métodos, módulos y componentes por el compilador JIT. [43]

Características fundamentales del lenguaje

La sintaxis de C# es muy expresiva, aunque cuenta con menos de 90 palabras clave; también es sencilla y fácil de aprender. La sintaxis de C# está basada en signos de llave, facilitando el reconocimiento por personas familiarizadas con C, C++ o Java.

La sintaxis de C# simplifica muchas de las complejidades de C++ y ofrece funciones eficaces tales como tipos de valores que aceptan valores NULL, enumeraciones, delegados, métodos anónimos y acceso directo a memoria. También admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que se pueden utilizar e implementar fácilmente. [44]

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. [45]

No existe el concepto de función global o variable fuera de una clase u objeto. Por su apego a la programación orientada a objetos (POO), es posible sobrecargar métodos y operadores. Soporta definición de interfaces. Ninguna clase puede poseer más de un padre (no se permite la herencia múltiple), pero sí puede suscribir un contrato con diversas interfaces. Soporta la definición de estructuras, a diferencia de C++ no son tan parecidas a las clases y poseen restricciones. En C#, una estructura es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia. [46]

Permite además la declaración de propiedades, eventos y atributos (que son construcciones declarativas).

C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen:

- Firmas de métodos encapsulados denominadas delegados, que permiten notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML. [47]

La interoperabilidad permite que los programas de C# realicen lo mismo que una aplicación de C++ nativa. C# admite incluso el uso de punteros y el concepto de código "no seguro" en los casos en que el acceso directo a la memoria es absolutamente crítico.

El proceso de generación de C# es simple en comparación con el de C y C++. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos. [48]

Anexo C. Visual Studio

El entorno Visual Studio es una herramienta de desarrollo, con la que se pueden crear aplicaciones que pueden contener uno o más proyectos de diversos lenguajes en función de los paquetes que se tienen instalados; en principio está a la disposición de C#, Visual Basic .NET y C++. También es posible crear aplicaciones C# utilizando el Framework .NET, que conforman un conjunto de herramientas que permiten compilar código fuente C# para crear aplicaciones. Visual Studio .NET hace uso del Framework .NET SDK y funciona como Front End para evitar la interacción con línea de comando.

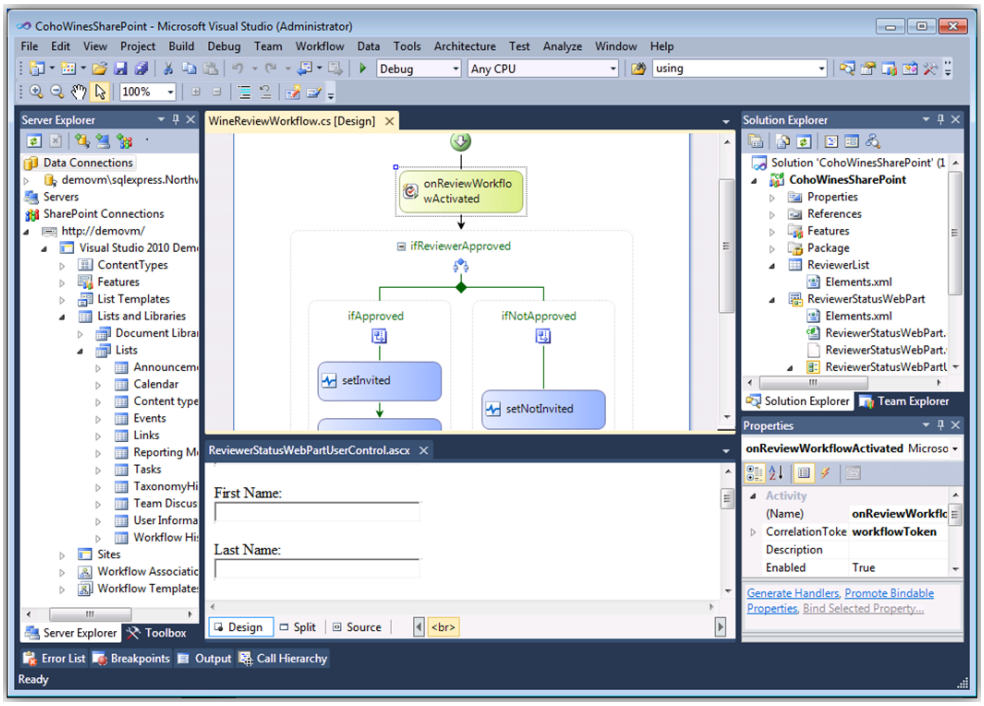


Figura C.1 Visual Studio.

Visual Studio es mucho más que un simple Front End. Algunas de las tareas que realiza son las siguientes:

- Navegar fácilmente por las clases por medio del visor de clases.
- Navegar por los archivos de uno o varios proyectos por medio del explorador de soluciones.
- Entender más rápidamente el código escrito debido al editor que resalta o colorea las palabras reservadas y los tipos de datos conocidos.

- Organizar múltiples proyectos y editar fácilmente sus propiedades.
- Configurar el entorno para que ejecute herramientas externas (como los precompiladores).
- Depurar los proyectos fácilmente y consultar valores de objetos de modo interactivo, así como realizar depuraciones remotas desde otras computadoras.
- Acceder a facilidades de búsqueda y reemplazo por hoja de código fuente activo y en archivos.
- Editar recursos (bitmaps, iconos, archivos binarios, etc.) por medio de herramientas integradas, y navegar por ellos por medio del visor de recursos.
- Agregar tareas por realizar haciendo uso de la lista de tareas pendientes, que además inserta automáticamente tareas a partir de comentarios prefijados.
- Generar documentación en formato HTML por medio de una herramienta provista con el entorno. Esta documentación es generada a partir del código fuente y comentarios con formatos específicos.
- Colapsar y expandir trozos de código para mejorar la legibilidad.
- Posibilidad de integrar herramientas al entorno por medio de un sistema de plug-ins. [49]

Un proyecto de Visual Studio se integrará por las hojas de código fuente C# de la siguiente manera:

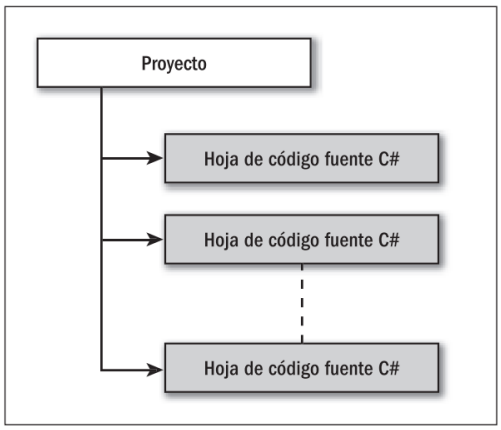


Figura C.2 Organización de un proyecto.

Usualmente, un proyecto construido tendrá como salida una aplicación ejecutable (archivo .EXE) o una librería (archivo .DLL).

Visual Studio soporta más de un proyecto abierto de manera simultánea en el mismo espacio de trabajo. Estos proyectos pueden relacionarse entre sí por medio de dependencias; de modo que al reconstruir la aplicación el entorno automáticamente reconstruirá todos los proyectos dependientes que se hayan modificado (por ejemplo una librería).

Visual Studio denomina solución a esta agrupación de proyectos (anteriormente llamado espacios de trabajo). Todo proyecto debe estar contenido en una solución; al crear un proyecto nuevo, Visual Studio crea automáticamente una solución que lo contiene.

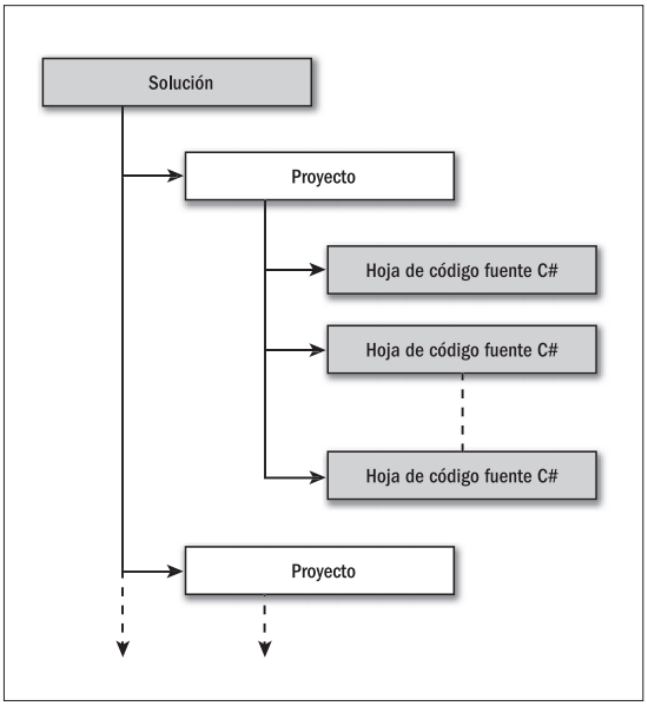


Figura C.3 Organización de una solución.

En la compilación de la solución se construyen todos los proyectos (que pueden estar relacionados directamente), es decir que se procesan todos los elementos de cada proyecto para generar los archivos de salida que correspondan.

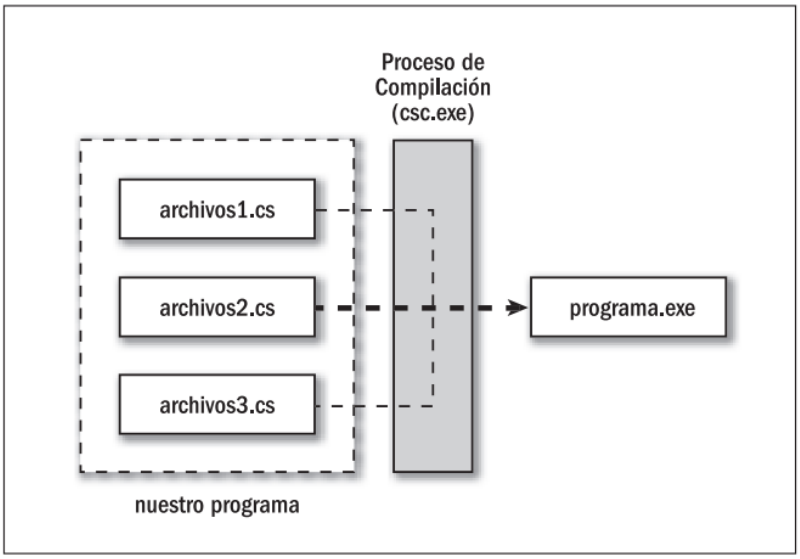


Figura C.4 Compilación de un proyecto.

Para los casos en los que no se posee un entorno Microsoft Visual Studio pueden utilizar Mono en ambientes Windows, GNU/Linux y Mac Os X. [50]

Anexo D. Developer Express

Developer Express es una empresa de desarrollo de software con sede en los Estados Unidos, la cual produce la codificación de herramientas de asistencia y componentes para desarrollodores de Delphi, C + + Builder y Microsoft Visual Studio. La mayor parte de su línea de productos son componentes VCL, .NET WinForms y ASP.NET que ayudan y mejoran la interfaz de usuario de Microsoft Windows y las aplicaciones de Microsoft Office. [51]

Productos

DXperience Enterprise y DXperience Universal incluyen una amplia gama de componentes visuales que ayudan a emular las interfaces de usuario más complejas de una manera práctica o a producir gráficos o informes de gran calidad. Los componentes .NET en DevExpress DXperience Enterprise y DXperience Universal están creados en Visual C# y se encuentran completamente optimizados para el Framework .NET. Se incluye el código fuente en C# de los componentes para ASP.NET, WinForms, Silverlight, WPF y también herramientas de productividad como CodeRush. [52]

DXperience Enterprise y Universal v2011 vol 1 (V11.1.6):

- Código Fuente de los productos para WinForms, ASP.NET AJAX, WPF y Silverlight.
- Librería ORM.
- 1 Licencia de Desarrollador por un año de suscripción.

Compatibilidad:

- Windows Azure.
- Windows 7.
- Windows Vista.
- Windows XP.
- Windows 2000

Arquitectura del producto:

- 32Bit.
- 64Bit.

Contenedores compatibles:

- Microsoft Visual Studio 2010.
- Microsoft Visual Studio 2008.
- Microsoft Visual Studio 2005.

Interfaz de Usuario

La interfaz de usuario puede ser compleja con diversas funcionalidades amigables, con un diseño de calidad y de fácil asimilación por el usuario final. Se puede observar un ejemplo en la siguiente imagen:

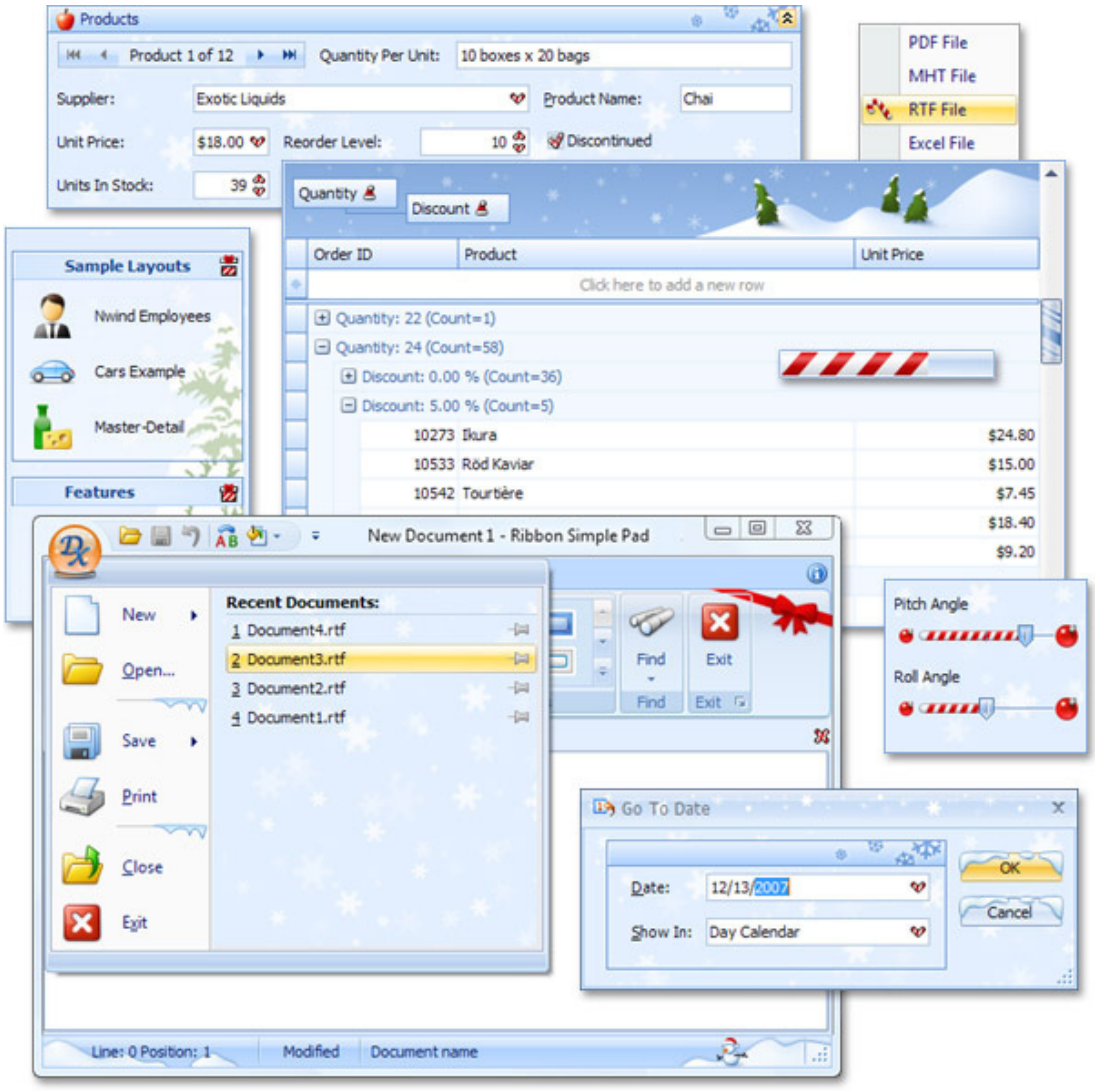


Figura D.1 Interfaz de Usuario.

Anexo E. Sharepoint

Microsoft SharePoint Server es una plataforma flexible basada en tecnologías web que amplía la capacidad de Microsoft Windows y Microsoft Office para permitir a las organizaciones publicar, compartir y encontrar información de la manera más sencilla, racionalizada y segura. Es una herramienta de administración de contenidos creando páginas web de forma muy rápida y sencilla.

La estructura funcional de Sharepoint se compone de la siguiente forma:

Figura E.1 Estructura Sharepoint.

Es la plataforma de colaboración empresarial que permite incrementar la productividad y administrar los contenidos a través de la conocida interfaz de Microsoft. La consolidación de soluciones mediante SharePoint contribuye a ahorrar mediante la reducción de los gastos de mantenimiento y formación y a través del aumento de la productividad del departamento de Tecnologías de la Información. [53]

Proporciona a los usuarios la mejor y la más flexible herramienta para organizar la información de forma efectiva, gestionar documentos y acceder rápidamente a los mismos, permitiendo una colaboración eficiente en un entorno amigable, basado en el explorador web y las herramientas de Microsoft.

A nivel de usuario, se presenta como una solución intuitiva para los ya conocedores de entornos de Microsoft Windows y Microsoft Office, que incorpora los procesos de negocio en la gestión documental, y que emplea el navegador web como interfaz de usuario, para los expertos en sistemas y desarrolladores, SharePoint Portal Server es una solución que incrementa espectacularmente el valor de su trabajo al combinar la capacidad de crear sitios web internos de una manera fácil con las funciones de gestión documental, indexación de contenidos desde múltiples orígenes y funciones de colaboración.

Las opciones de contenido de SharePoint permiten a todos los usuarios participar en la administración de contenidos de una forma regulada y conforme a las normativas. Estas funciones hacen posible un equilibrio perfecto entre la experiencia de los usuarios con respecto a los procesos y directivas.

Ventajas de utilizar Sharepoint:

- Compartir documentos.
- Encuestas.
- Foros.
- Creación de blogs.
- Búsquedas en el portal.
- Creación de páginas dinámicas
- Interfaz amigable.
- Control de versiones.
- Permisos a nivel de carpetas y documentos.
- Creación de formularios.
- Creación de flujos. [54]

Anexo F. Windows Communication Foundation (WCF)

Windows Communication Foundation es una parte de .NET Framework que proporciona un modelo de programación unificado para crear rápidamente aplicaciones orientadas a servicios que se comunican por la Web.

Los servicios Web que incluye los protocolos estándar para la comunicación de aplicación a aplicación incluyen seguridad, coordinación de transacciones distribuidas y una comunicación fiable. WCF está diseñado para ofrecer un enfoque manejable a la informática distribuida, interoperabilidad ancha y asistencia directa para la orientación sobre el servicio.

WCF simplifica el desarrollo de aplicaciones conectadas a través de un nuevo modelo de programación orientado a servicios. Admite muchos estilos de desarrollo de aplicaciones distribuidas proporcionando una arquitectura superpuesta, la arquitectura proporciona primitivos asíncronos de paso de aprobación de mensajes sin tipo, se crean funciones de protocolos para un intercambio de datos de transacción seguro y fiable, así como una amplia variedad de opciones de codificación y transporte.

El modelo de programación tipificada (llamado modelo de servicio) está diseñado para facilitar el desarrollo de aplicaciones distribuidas y proporcionar a los desarrolladores pericia en servicios Web ASP.NET, comunicación remota .NET Framework y Enterprise Services. El modelo de servicio presenta una asignación sencilla de conceptos de servicios Web para aquellos de Common Language Runtime (CLR) .NET Framework, incluyendo la asignación ampliable y flexible de mensajes para la implementación de servicios en lenguajes como Visual C# o Visual Basic.

Incluye funciones de serialización que habilitan el acoplamiento separado y el control de versiones y proporciona integración e interoperabilidad con sistemas distribuidos .NET Framework existentes, como Message Queue Server (MSMQ), COM+, servicios Web ASP.NET, Mejoras de servicios Web (WSE). [55]

Se incluyen referencias de las bibliotecas en el proyecto como se muestra en la siguiente figura:

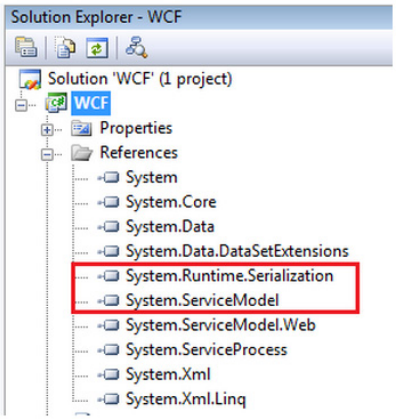


Figura F.1 Referencia de clases utilizadas por WCF.

La base para las nuevas aplicaciones basadas en Windows es .NET Framework. Por lo tanto, WCF se implementa principalmente como un conjunto de clases por encima de .NET Framework CLR. Dado que extiende su entorno familiar, WCF permite crear aplicaciones orientadas a objetos utilizando .NET Framework, para ser integradas las aplicaciones orientadas a servicios de una manera conocida.

Figura F.2 Servicio de WCF.

La figura muestra una vista de un cliente WCF y de un servicio. Los dos interactúan utilizando SOAP, WCF es la representación del mensaje nativo. WCF hace frente a un intervalo de desafíos para hacer que las aplicaciones se comuniquen. Sin embargo, hay tres cosas que destacan como los aspectos más importantes de WCF:

- Unificación de las tecnologías de comunicación .NET Framework existentes.
- Compatibilidad para interoperabilidad entre proveedores, incluyendo confiabilidad, seguridad y transacciones.
- Orientación explícita al servicio.

Debido a que el mecanismo de comunicación fundamental de WCF es un servicio Web basado en SOAP, las aplicaciones basadas en WCF pueden comunicarse con otro software que se ejecute en una variedad de contextos. Una aplicación generada en WCF puede interactuar con lo siguiente:

- Las aplicaciones basadas en WCF que se ejecutan en un proceso diferente en el mismo equipo de Windows.
- Las aplicaciones basadas en WCF que se ejecutan en otro equipo de Windows.
- Las aplicaciones generadas en otras tecnologías, como servidores de aplicaciones de J2EE, que son compatibles con los servicios Web estándar. Estas aplicaciones se pueden estar ejecutando en equipos con Windows o en los equipos que ejecutan otros sistemas operativos.

WCF se basa en la noción de comunicación basada en mensajes y cualquier cosa que se pueda modelar como un mensaje (por ejemplo, una solicitud HTTP o un mensaje de MSMQ), se puede representar de manera uniforme en el modelo de programación. Esto habilita una API unificada en todos los mecanismos de transporte diferentes.

El modelo distingue entre clientes, que son aplicaciones que inician la comunicación y servicios, que son aplicaciones que esperan a que los clientes se comuniquen con ellos y respondan a esa comunicación. Una única aplicación puede actuar como cliente y como servicio.

Los mensajes se envían entre extremos. Los extremos son los lugares donde los mensajes se envían o reciben (o ambos), y definen toda la información requerida para el intercambio de mensajes. Un servicio expone uno o más extremos de la aplicación (y a cero o más extremos de la infraestructura), y el cliente genera un extremo que es compatible con uno de los extremos del servicio.

En su forma final, un servicio es un programa. Como otros programas, un servicio se debe ejecutar en un ejecutable. Esto se conoce como un servicio con host propio.

- Los servicios también se pueden hospedar o ejecutar en un ejecutable administrado por un agente externo, como IIS o Servicio de activación de Windows (WAS). WAS permite activar automáticamente aplicaciones WCF cuando se implementan en un equipo.
- Los servicios también se pueden ejecutar manualmente como ejecutables (archivos .exe).
- Un servicio también se puede ejecutar automáticamente como un servicio de Windows. Los componentes COM+ también se pueden hospedar como servicios WCF. [56]

La arquitectura de WCF se encuentra definida en la siguiente figura:

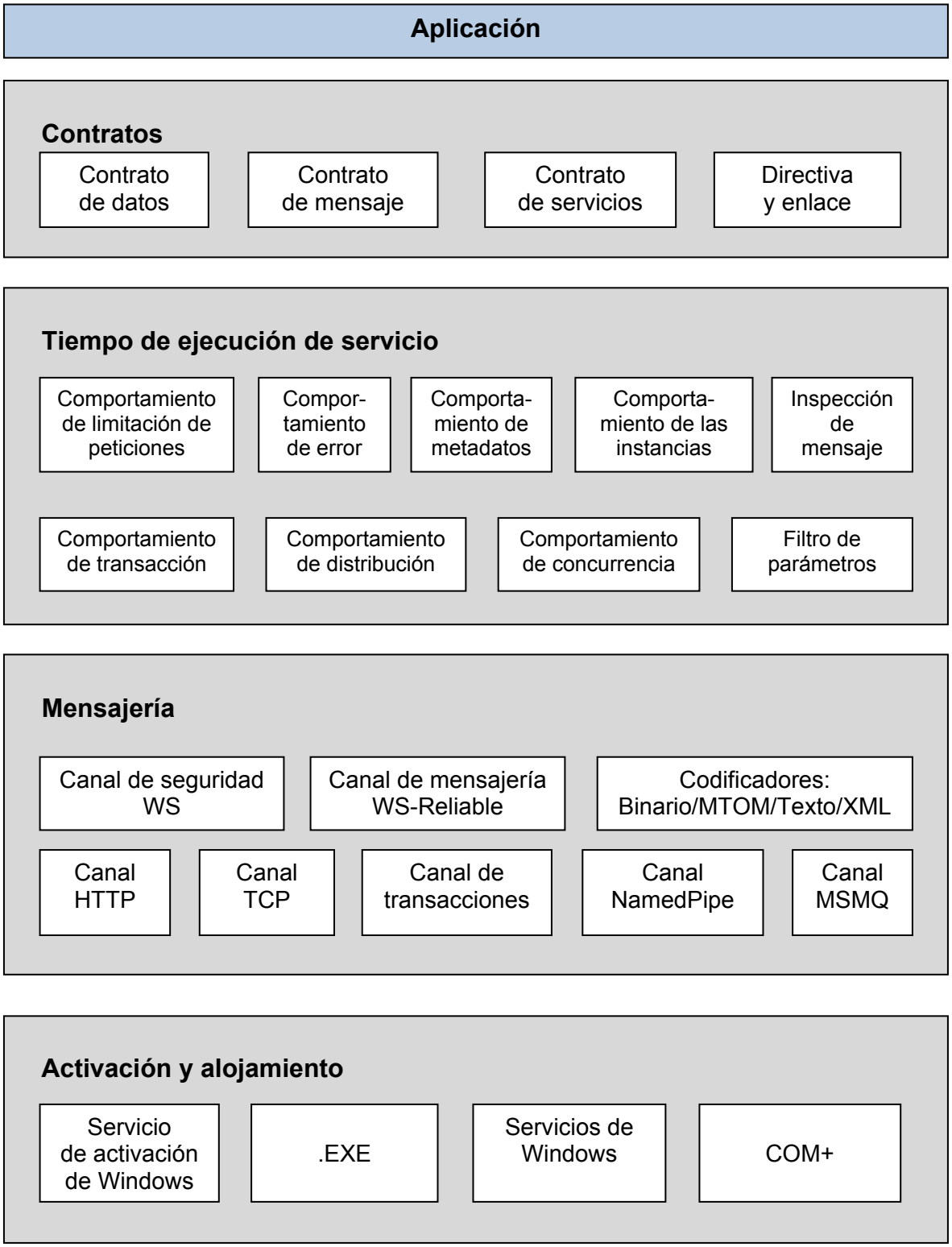


Figura F.3 Arquitectura de WCF.

Las herramientas Windows Communication Foundation de Microsoft están diseñadas para facilitar la creación, implementación y administración de aplicaciones WCF:

Tabla F.1 Herramientas de WCF.

Herramienta	Descripción
Herramienta de utilidad de metadatos de ServiceModel (Svcutil.exe)	Genera el código de modelo de servicio de los documentos de metadatos y documentos de metadatos del código de modelo de servicio.
Herramienta de búsqueda de clave privada (FindPrivateKey.exe)	Recupera la clave privada de un almacén especificado.
Herramienta de registro de ServiceModel (ServiceModelReg.exe)	Administra el registro y la anulación de registro de ServiceModel en un equipo único.
Herramienta de configuración del modelo de servicio COM+ (ComSvcConfig.exe)	Configura interfaces de COM+ que se van a exponer como servicio Web.
Herramienta del editor de configuración (SvcConfigEditor.exe)	Crea y modifica la configuración para los servicios WCF.
Herramienta del visor de seguimiento de servicio (SvcTraceViewer.exe)	Ver, agrupar y filtrar mensajes de seguimiento para poder diagnosticar, reparar y comprobar los problemas con los servicios WCF.
Utilidad de configuración de WS-AtomicTransaction (wsatConfig.exe)	Configura valores básicos de compatibilidad de WS-AtomicTransaction mediante una herramienta de línea de comandos.
Complemento MMC de configuración de WS-AtomicTransaction	Configura valores básicos de compatibilidad de WS-AtomicTransaction mediante un complemento MMC.
Herramienta de registro de servicio de flujo de trabajo (WFServicesReg.exe)	Registra un servicio de Windows Workflow.
WCF Service Auto Host	Hospeda servicios WCF contenidos en archivos de bibliotecas (.dll)
WCF Test Client	Herramienta GUI que permite introducir parámetros de tipos arbitrarios, enviar esa entrada al servicio y ver la respuesta que devuelve el servicio.

[57]

Anexo G. Windows Presentation Foundation (WPF)

Windows Presentation Foundation (WPF) es un sistema de presentación de la próxima generación, para crear aplicaciones cliente de Windows que proporcionen una experiencia impactante para el usuario desde el punto de vista visual.

Es una tecnología que permite construir aplicaciones con una potente (novedosa, interactiva, espectacular) interfaz de usuario. Tiene soporte para los sistemas operativos de Windows XP SP2, Windows 2003 SP1, Windows Vista y Windows 7.

El núcleo de WPF es un motor de representación basado en vectores e independiente de la resolución que se crea para sacar partido del hardware de gráficos moderno. WPF extiende el núcleo con un conjunto completo de características de desarrollo de aplicaciones que incluye Extensible Application Markup Language (XAML), controles, enlace de datos, diseño, gráficos 2-D y 3-D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en Microsoft .NET Framework con el cual es posible compilar aplicaciones que incorporen otros elementos de la biblioteca de clases de .NET Framework. WPF constituye un subconjunto de tipos de .NET Framework en su mayoría ubicados en el espacio de nombres System.Windows.

WPF incluye construcciones de programación adicionales que mejoran las propiedades propiedades de dependencia y los eventos enrutados, también proporciona mejoras de programación adicionales para el desarrollo de aplicaciones cliente de Windows debido a la capacidad para programar una aplicación mediante código de lenguaje marcado y subyacente. [58]

Se utiliza el lenguaje marcado Extensible Application Markup Language (XAML) para implementar la apariencia de una aplicación, y los lenguajes de programación administrados (subyacentes) para implementar su funcionalidad. La separación entre la apariencia y el funcionamiento aporta las ventajas siguientes:

- Se reducen los costos de programación y mantenimiento, al no estar el marcado específico de la apariencia estrechamente relacionado con el código específico del funcionamiento.

- La programación es más eficaz al implementar la apariencia de una aplicación al mismo tiempo que se implementa la funcionalidad.

- Se pueden usar varias herramientas de diseño para implementar y compartir el marcado XAML.

- La globalización y localización de las aplicaciones WPF se simplifica. [59]

XAML es un lenguaje de marcado basado en XML que se utiliza para implementar la apariencia de una aplicación mediante declaración, con el que se puede crear ventanas, cuadros de diálogo, páginas y controles de usuario, así como para rellenarlos con controles, formas y gráficos.

Se pueden crear elementos visibles de la UI en el marcado XAML declarativo y, a continuación, separar la definición de la UI de la lógica en tiempo de ejecución mediante archivos de código subyacente, que se unen al marcado mediante definiciones de clases parciales. XAML representa directamente la creación de instancias de objetos en un conjunto concreto de tipos de respaldo definidos en ensamblados. El código XAML habilita un flujo de trabajo en el que las partes independientes pueden funcionar en la UI y la lógica de una aplicación, a través de herramientas potencialmente diferentes.

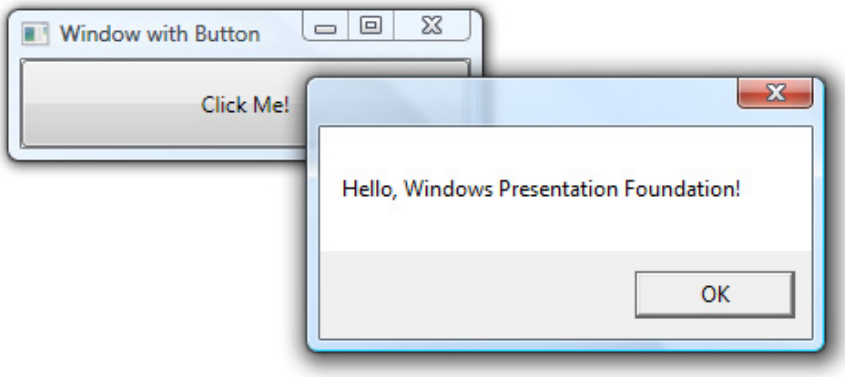


Figura G.1 Ejemplo de WPF.

En concreto, el código XAML define una ventana y un botón mediante los elementos Window y Button. Cada elemento se configura con atributos, como el atributo Title del elemento Window para especificar el texto de la barra de título de la ventana. En tiempo de ejecución, WPF convierte los elementos y atributos definidos en el marcado en instancias de clases de WPF.

El modelo de programación primario de WPF se expone a través de código administrado. El CLR proporciona varias características que pueden hacer el proceso más sólido y productivo (como son la administración de memoria, el control de errores, el sistema de tipos común, etc.). [60]

Los componentes principales de WPF se muestran en la siguiente figura:

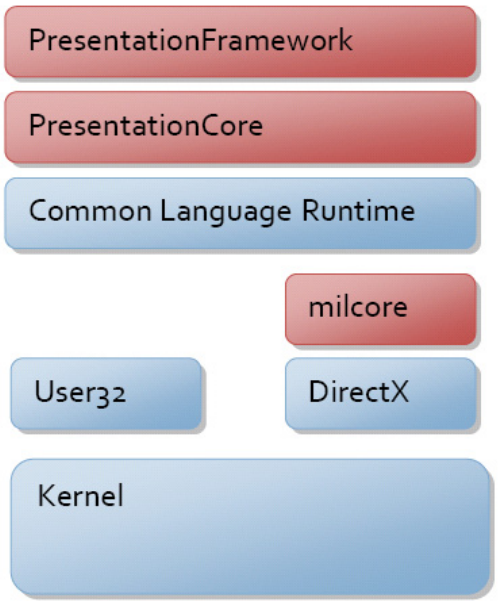


Figura G.2 Arquitectura de WPF.

PresentationFramework, PresentationCore y milcore son los componentes principales del código de WPF. Milcore se ha escrito en código no administrado para conseguir una estrecha integración con DirectX.

La visualización en WPF se realiza a través del motor de DirectX, lo que permite una presentación eficaz mediante hardware y software. El motor de composición de milcore es sumamente sensible al rendimiento. [61]

Referencias

[1] Universidad Nacional Autónoma de México, Facultad de Ingeniería. Proyectos realizados en conjunto con la SCJN.

[2] Suprema Corte de Justicia de la Nación, Dirección General de Tecnologías de la Información. Documentación generada durante los convenios 2009 y 2010 correspondientes al Sistema Integral Legislativo y al Sistema de Informática Jurídica.

[3] http://www.ifai.org.mx/transparencia/LFTAIPG.pdf
Ley Federal de Transparencia. 01 de Junio el 2011.

[4] http://www.scjn.gob.mx/2010/transparencia/Paginas/ProgramaAnualEjecucion2011.aspx
Programa Anual de Ejecución 2011. 01 de Junio del 2011.

[5] http://www.dotnetconsult.co.uk/weblog2/content/binary/SOA-WCF-WF.pdf
Arquitectura Orientada a Servicios (SOA). 19 de Septiembre del 2011.

[6] http://www.mastermagazine.info/articulo/3391.php
Conceptos básicos de SOA. 19 de Septiembre del 2011.

[7] http://www.accenture.com/SiteCollectionDocuments/Local_Spain/PDF/SOA.pdf
Implementación de SOA. 19 de Septiembre del 2011.

[8] http://www.iidea-solutions.com/consultoria-ti/arquitectura-aplicaciones/?gclid=CMrX8PzklasCFWJeTAodNWh7tw
Características fundamentales de SOA. 19 de Septiembre del 2011.

[9] http://www.campusmvp.com/libros/indices/Indice_Libro_SOA_Krasis_Press.pdf
Buenas prácticas y diseño de aplicaciones SOA. 19 de Septiembre del 2011.

[10] http://www.epicor.com/lac/Solutions/Pages/Serviceoriented.aspx
Beneficios de SOA. 19 de Septiembre del 2011.

[11] http://soloprogramadores.peruforo.org/t15-que-es-la-arquitectura-cliente-servidor
Arquitectura Cliente/Servidor. 27 de Septiembre del 2011.

[12] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf
Introducción a la Arquitectura Cliente/Servidor. 27 de Septiembre del 2011.

[13] http://antoniomtz.org/2009/07/programacion-en-3-capas/
Programación en tres capas. 27 de Septiembre del 2011.

[14]
http://icomparable.blogspot.com/2008/10/arquitectura-n-tier-o-arquitectura-n.html
Arquitecturas n-Tier y n-Layer. 27 de Septiembre del 2011.

[15] http://ccia.ei.uvigo.es/docencia/SCS/Tema1.pdf
Modelos y tipologías. 27 de Septiembre del 2011.

[16]
http://ingeniero-arielbustos.blogspot.com/2010/04/metologias-modernas-del-software.html
Metodologías modernas del software. 27 de Septiembre del 2011.

[17]
http://www.ibm.com/developerworks/ssa/websphere/library/techarticles/1005_defreitas/1005_defreitas.html
Desarrollo iterativo e incremental. 27 de Septiembre del 2011.

[18] http://www.proyectosagiles.org/desarrollo-iterativo-incremental
Beneficios del desarrollo iterativo e incremental. 27 de Septiembre del 2011.

[19] http://www.programacionextrema.org/
Programación Extrema. 27 de Septiembre del 2011.

[20] http://www.chuidiang.com/ood/metodologia/extrema.php
Prácticas básicas de la programación extrema. 27 de Septiembre del 2011.

[21] http://www.mitecnologico.com/Main/FormasNormalesBasesDatos
Formas Normales (Base de Datos). 27 de Septiembre del 2011.

[22] http://sistemas.itlp.edu.mx/tutoriales/basedat1/tema4_2.htm
Primera y segunda forma normal. 27 de Septiembre del 2011.

[23] http://cnx.org/content/m18350/latest/
Ejemplos de normalización (1FN, 2FN y 3FN). 27 de Septiembre del 2011.

[24] http://www.phlonx.com/resources/nf3/nf3_tutorial_spanish.pdf
Las tres formas normales. 27 de Septiembre del 2011.

[25] http://www.angelfire.com/my/jimena/bdat1/guia7.htm
Tercera forma normal. 27 de Septiembre del 2011.

[26] http://msdn.microsoft.com/es-es/library/e80y5yhx(v=vs.80).aspx
ADO.NET. 12 de Julio del 2011.

[27] http://msdn.microsoft.com/en-us/library/system.windows.data.binding.aspx
Clases Binding. 20 de Junio del 2011.

[28] http://msdn.microsoft.com/es-es/library/h974h4y2(v=vs.80).aspx
Componente BindingSource 03 de Julio del 2011.

[29] http://www.tuobra.unam.mx/publicadas/040702105342-ERP.html
Planeación de Recursos de la Empresa. 10 de Junio del 2011.

[30]
https://www.ccn-cert.cni.es/publico/serieCCN-STIC401/es/n/non_repudiation.htm
No repudio (definición). 12 de Julio del 2011.

[31] http://www.smtp.com/
Protocolo Simple de Transferencia de Correo. 10 de Junio del 2011.

[32] http://dev.mysql.com/doc/refman/5.0/es/stored-procedures.html
Procedimientos almacenados y funciones. 12 de Julio del 2011.

[33] http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/
Pruebas de software. 25 de Julio del 2011.

[34] http://homepages.mty.itesm.mx/al450951/
Web Services. 22 de Julio del 2011.

[35] http://msdn.microsoft.com/es-es/library/aa970268.aspx
Introducción a WPF. 20 de Junio del 2011.

[36]
http://www.vacationinnicaragua.com/microsoft/expression-blend/XAML/concept_xaml_whatis.htm
Introducción a XAML. 20 de Junio del 2011.

[37] http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml
Tecnología XML. 15 de Junio del 2011.

[38] http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_2/Lenguaje%20c.pdf
Lenguaje C# y la plataforma .NET. 04 de Agosto del 2011.

[39] http://msdn.microsoft.com/es-es/library/z1zx9t92(v=vs.80).aspx
Arquitectura del Framework .NET. 04 de Agosto del 2011.

[40] SHEPHERD, George. Microsoft ASP.NET 4.0 Paso a paso. España: Anaya Multimedia, primera edición, 2010.

[41] LIBERTY, Jesse; HORVATH, David. Aprendiendo C++ para Linux en 21 Días. México: Prentice Hall, primera edición, 2000.

[42] DEITEL, Harvey; DEITEL, Paul. Cómo programar en C/C++ y Java. México: Pearson/Prentice Hall, cuarta edición, 2004.

[43] DEITEL, Harvey; DEITEL, Paul. Cómo programar en C#. México: Pearson/Prentice Hall, segunda edición, 2007.

[44] http://www.devjoker.com/contenidos/Tutorial-C/125/Introduccion-a-C.aspx
Características del lenguaje C#. 10 de Agosto del 2011.

[45] CEBALLOS, Francisco. Enciclopedia de Microsoft® Visual C#. México: Alfaomega, tercera edición, 2010.

[46] http://www.scribd.com/doc/2984731/El-Lenguaje-C
Introducción a C#. 10 de Agosto del 2011.

[47] http://www.canalvisualbasic.net/manual-net/c-sharp/#cSharp
Manual C#. 10 de Agosto del 2011.

[48] GONZÁLEZ, Alfons. Programación de Bases de Datos con C#. México: Alfaomega, primera edición, 2010.

[49]
http://www.visualstudiotutor.com/2010/02/visual-studio-debugging-tutorial-basics/
Tutorial de Visual Studio básico. 13 de Agosto del 2011.

[50] http://www.microsoft.com/spain/visualstudio
Microsoft Visual Studio 2010. 13 de Agosto del 2011.

[51] http://www.componentsource.com/products/dxperience/summary.html
Resumen de componentes DevExpress. 01 de Agosto del 2011.

[52] http://www.devexpress.com/Products/NET/Controls/WinForms/#main|overview
WinForms de DevExpress. 01 de Agosto del 2011.

[53] http://sharepoint.microsoft.com/es-mx/Paginas/default.aspx
Microsoft Sharepoint 2010. 31 de Agosto del 2011.

[54] http://mssharepoint.multiply.com/
Arquitectura Sharepoint. 31 de Agosto del 2011.

[55] http://msdn.microsoft.com/es-mx/netframework/aa663324
Windows Communication Foundation. 03 de Septiembre del 2011.

[56] http://www.davidchappell.com/introducingwcfv1.2.1.pdf
Introducción a Windows Communication Foundation. 03 de Septiembre del 2011.

[57] http://msdn.microsoft.com/es-mx/library/ms733128(v=vs.90).aspx
Arquitectura de Windows Communication Foundation. 03 de Septiembre del 2011.

[58] http://msdn.microsoft.com/es-es/library/ms754130.aspx
Windows Presentation Foundation. 11 de Septiembre del 2011.

[59] http://msdn.microsoft.com/es-es/library/aa970268.aspx
Introducción a Windows Presentation Foundation. 11 de Septiembre del 2011.

[60] http://msdn.microsoft.com/es-es/library/ms750441.aspx
Arquitectura de Windows Presentation Foundation. 11 de Septiembre del 2011.

[61] http://www.onglasses.net/wwwroot/upfiles/User/manual_wpf.pdf
Tutorial de Windows Presentation Foundation. 11 de Septiembre del 2011.