

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DISTRIBUIDO PARA LOS PROCESOS DEL SERVICIO MÓVIL TERRESTRE Y DE LOS SISTEMAS DE MODULACIÓN DIGITAL DE BANDA ANCHA PARA LA SECRETARÍA NACIONAL DE TELECOMUNICACIONES (SENATEL)

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN**

**ANDREA PATRICIA PINTO RAMÍREZ
andrepin29@gmail.com**

**JOSÉ ALBERTO GAIBOR COLOMA
alebertogaibor89@gmail.com**

**DIRECTOR: ING. HENRY MANOLO ECHEVERRÍA CULQUI
henry.echeverria@epn.edu.ec**

**CODIRECTOR: ING. RAÚL DAVID MEJÍA NAVARRETE, MSc.
david.mejia@epn.edu.ec**

Quito, Diciembre 2014

DECLARACIÓN

Nosotros, Andrea Patricia Pinto Ramírez y José Alberto Gaibor Coloma, declaramos bajo juramento que el trabajo descrito es de nuestra autoría; no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en el documento.

A través de la presente declaración cedemos nuestros derechos de propiedad correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Andrea Patricia Pinto Ramírez

José Alberto Gaibor Coloma

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Andrea Patricia Pinto Ramírez y José Alberto Gaibor Coloma, bajo mi supervisión.

Ing. Henry Echeverría

Director del Proyecto

Ing. David Mejía, MSc.

Codirector del Proyecto

AGRADECIMIENTO

Doy gracias a Dios, a Quien le debo mi existencia, mi razón y mi corazón. Le doy las gracias por encontrar en Su palabra las respuestas a mis más grandes inquietudes, por ser mi fortaleza, felicidad y satisfacción. Amo a Dios por sobre todas las cosas y Le agradezco por todas las bendiciones que me ha concedido, y por los dones puestos en mí.

Agradezco a la mujer que me dio la vida y que me amó desde el momento que supo de mi existencia, le agradezco por enseñarme a poner el corazón en cada actividad que haga por muy pequeña que ésta sea, gracias a su valioso ejemplo de constancia y dedicación, sé con certeza que claudicar nunca será una opción, le agradezco a la mujer inquebrantable, llena de fe y alegría a la cual tengo el privilegio de llamar Madre.

Agradezco a mi Padre, quien para mí es el hombre más responsable, honesto, trabajador y perseverante que existe, le agradezco por haberme enseñado a ser una persona con valores y virtudes, independiente y capaz de vencer de cualquier dificultad y obstáculo. Le agradezco a mi padre, a quien le debo tanto y toda mi vida no me alcanzará para retribuirle todo el amor y apoyo entregado.

Agradezco a mi Hermana por estar siempre pendiente de mí, y por su inmenso deseo de que haga con mi vida algo memorable. Aunque somos dos personas muy diferentes, sé que nadie en el mundo me podría amar más. Quiero que tenga la certeza que del tiempo no disipará mi cariño, ella es la única persona por quien daría mi vida sin pensarlo dos veces. Agradezco a mi hermana por tantas cosas que ha dado y hecho por mí.

Agradezco a José Alberto por haber aceptado emprender este reto conmigo, nunca deje de quererle y confiar en él. Gracias por la ayuda y tolerancia durante el tiempo de desarrollo del proyecto. Todos los éxitos profesionales y personales que deseo obtener en mi vida también se los deseo a él, y pido a Dios que no deje de bendecirle e iluminar su camino para que tome decisiones acertadas en su vida.

Aunque sé que ninguna palabra será suficiente para expresar mi gratitud, le agradezco infinitamente al Ing. David Mejía por habernos ayudado desinteresada e incondicionalmente en todos los aspectos de este proyecto. Le agradezco además por tener ese alto sentido de colaboración y compromiso que entrega en cada uno de los proyectos de los cuales forma parte. Le expreso mi más profunda admiración, le agradezco todos sus consejos, enseñanzas y lecciones, y aunque sé bien que a veces no los merecía, nunca dudó en seguir ayudándonos.

Agradezco al Ing. Henry Echeverría por haber aceptado ser Director de este Proyecto y depositar su confianza en nosotros, le agradezco además por la disponibilidad que siempre tuvo conmigo y por ser la persona tan amable y atenta que es.

Agradezco al Ing. Harold Miranda quién pensó y propuso el desarrollo del presente Proyecto, le agradezco su valiosa ayuda, apoyo, confianza y amistad. A pesar de su trabajo y de sus ocupaciones siempre le dio prioridad a mis peticiones y requerimientos. Para mí Harold es un ser humano excepcional, quien va a tener mi eterna gratitud y siempre va a poder contar con mi sincera amistad.

Agradezco a los Ingenieros que forman parte de la Secretaría Nacional de Telecomunicaciones por la colaboración brindada en el desarrollo de este proyecto, especialmente: al Ing. Ramiro Andrade, quien siempre tuvo la predisposición para ayudarme y de quién siempre obtuve una respuesta positiva, a la Ing. Mónica Riofrío quien estuvo al pendiente de las cosas que podría necesitar para la implementación del proyecto, a la Ing. Paulina Zhunio, al Ing. Luis Ibarra, al Ing. Carlos Gallo, a la Ing. Jenny Velásquez, y a todos los Ingenieros que constituyen la Dirección General de Gestión del Espectro Radioeléctrico. De corazón, muchas gracias por todo.

Finalmente, agradezco a todos mis amigos y amigas por los momentos compartidos. Tal vez sea el destino el que determina a qué personas tengamos que conocer, pero definitivamente es nuestra voluntad decidir a quiénes les entregamos nuestra amistad y gratitud.

Andrea

AGRADECIMIENTO

Agradezco a Dios por las bendiciones y momentos felices brindados, agradezco los retos y dificultades que me ha presentado, que me han puesto a prueba, partícipes esenciales de mi formación y crecimiento, aliciente para seguir adelante, enseñándome siempre a levantarme y luchar, luchar porque al final el esfuerzo siempre vale la pena.

A mis padres, decirles que no tengo palabras para agradecerles por todo el amor que me han brindado, amor que es mi combustible para lograr mis objetivos, por la paciencia que solo un padre puede tener, por tener siempre los brazos abiertos, gracias por la educación que me han brindado, me siento orgulloso de la persona que soy y se los debo solamente a ustedes, quiero decirles que siempre los amaré, que los llevo en mi corazón en todo momento.

A mi hermano por ser el mejor amigo que he tenido, juntos lloramos, juntos reímos, nunca ha sido necesario decir nada para saber que podemos confiar uno en el otro bajo cualquier circunstancia, si pudiese elegir al mejor hermano, te elegiría a ti, gracias por brindarme tu amistad siempre podrás contar conmigo incondicionalmente.

Agradezco a Andrea por confiar en mí y brindarme la oportunidad de acompañarla en esta empresa, juntos supimos sobrellevar los malos momentos, juntos vivimos experiencias maravillosas que permanecerán conmigo para siempre, lo vivido nadie nos quita, el amor no se borra, muy a nuestra manera trabajábamos en equipo siempre logrando cumplir los objetivos.

“Gratitud al maestro que sufre y se afana, frente al joven la voz del mañana”, mi gratitud para todos los profesores que desinteresadamente han brindado el conocimiento necesario para crecer tanto como profesional y más importante aún como persona. A los profesores que nos acompañaron durante el desarrollo del proyecto mi admiración y aprecio ya que supieron motivar en mí el deseo de aprender, gracias por ir más allá de los viejos preceptos de enseñanza y tratar de establecer nuevos paradigmas.

Al Ing. David Mejía por confiar en Andrea y en mí, brindándonos su apoyo incondicional y desinteresado, que aún con el ahínco con el que afronta sus labores diarias siempre tuvo el tiempo para orientarnos, para motivarnos a seguir adelante y no desmayar en llevar el proyecto siempre por buen camino.

Al Ing. Henry Echeverría por el voto de confianza al iniciar el proyecto, por la apertura que nos brindó siempre, los oportunos llamados de atención cuando fueron necesarios, gracias por más que un director ser un guía y brindarnos sus consejos y experiencia.

Agradezco que de mi paso por las instituciones educativas, pudiese obtener mucho más que conocimiento, obtuve innumerables experiencias junto a maravillosas personas a las que hoy puedo llamar amigos.

Agradezco a la vida por el camino que ha trazado para mí, que me ha traído hasta este momento tan feliz, que siembra en mí nuevas metas y las ganas de seguir viviendo para cumplirlas.

Alberto

DEDICATORIA

Dedicado a mi Padre Celestial y a las personas que más amo: Jorge, M. Elena y Tatiana.

Andrea

DEDICATORIA

A quienes marcaron mi camino con sus enseñanzas, fascinaron mi mente con sus historias y anécdotas, nutrieron mi espíritu con sus risas, para quien aún permanece junto a nosotros y para los que partieron de este mundo creyendo hasta el último minuto en mí, en donde estén los extraño, este logro es dedicado para ustedes mis queridos abuelos, gracias por acercar el horizonte cada día un poco más.

Alberto

ÍNDICE

DECLARACIÓN	I
CERTIFICACIÓN.....	II
AGRADECIMIENTO.....	III
AGRADECIMIENTO.....	V
DEDICATORIA	VII
DEDICATORIA.....	VIII
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABLAS	XIX
RESUMEN.....	XX
PRESENTACIÓN	XXI
1. CAPÍTULO I.....	1
MARCO TEÓRICO	1
1.1. INTRODUCCIÓN.....	1
1.2. SERVICIOS DE RADIOCOMUNICACIONES	1
1.2.1. SERVICIOS DE TELECOMUNICACIONES.....	2
1.2.2. ESTADÍSTICAS DEL SECTOR	3
1.2.3. ESPECTO RADIOELÉCTRICO	5
1.2.4. SISTEMAS VHF y UHF.....	6
1.2.4.1. Aplicaciones en el rango VHF (30-300 MHz).....	7
1.2.4.2. Aplicaciones en el rango UHF (300-3000 MHz).....	7
1.2.4.3. Sistemas de Radio de Dos Vías	8
1.2.5. SISTEMAS DE MODULACIÓN DIGITAL DE BANDA ANCHA (MDBA) .	8
1.2.5.1. Aplicaciones MDBA.....	10
1.3. SERVICIOS WEB	10
1.3.1. INTRODUCCIÓN	10
1.3.2. DEFINICIÓN DE SERVICIOS WEB.....	11
1.3.3. EVOLUCIÓN DE LAS APLICACIONES INFORMÁTICAS.....	12
1.3.3.1. Aplicaciones Monolíticas.....	12

1.3.3.2. Aplicaciones Cliente-Servidor	13
1.3.3.3. Aplicaciones Web	13
1.3.3.3.1 <i>Diferencia entre Aplicación Web y Aplicación de Escritorio</i>	13
1.3.3.4. Aplicaciones de n-Niveles	14
1.3.4. HIPERTEXTO Y HTML	15
1.3.5. URL, URN Y URI.....	17
1.3.6. HTTP.....	18
1.3.7. TECNOLOGÍAS BASE DE LOS SERVICIOS WEB	20
1.3.7.1. XML.....	22
1.3.7.2. SOAP	23
1.3.7.3. WSDL.....	24
1.3.7.4. UDDI	25
1.3.8. REST	26
1.3.9. SEGURIDAD EN LOS SERVICIOS WEB	28
1.3.9.1. TLS Y SSL	29
1.3.9.2. RAID	30
1.3.10. SOA (SERVICE ORIENTED ARCHITECTURE)	32
1.4. METODOLOGÍA DE DESARROLLO	33
1.4.1. METODOLOGÍAS ÁGILES	34
1.4.2. SCRUM.....	35
1.4.2.1. Fases de Scrum	35
1.4.2.2. Roles de Scrum	36
1.4.2.3. Elementos de Scrum.....	37
1.4.2.4. Reuniones de Scrum.....	38
2. CAPÍTULO II.....	40
ANÁLISIS DE REQUERIMIENTOS.....	40
2.1. INTRODUCCIÓN.....	40
2.2. SECRETARÍA NACIONAL DE TELECOMUNICACIONES (SENATEL)....	40
2.2.1. MISIÓN	41
2.2.2. VISIÓN	41
2.2.3. ISO 9001:2008	41
2.2.4. ESTRUCTURA ORGÁNICA DE LA SECRETARÍA NACIONAL DE TELECOMUNICACIONES.....	42

2.2.5. FUNCIONES DE LA SENATEL	44
2.2.5.1. Otorgamiento de Títulos Habilitantes	44
2.2.5.2. Otorgamiento de Registros y Licencias.....	44
2.2.5.3. Procedimiento para el Otorgamiento de Títulos Habilitantes (OTH)	
45	
2.2.5.3.1. <i>Proceso de recepción y digitalización de solicitudes</i>	46
2.2.5.3.2. <i>Elaboración del Informe Jurídico</i>	46
2.2.5.3.3. <i>Elaboración del Informe Técnico</i>	47
2.2.5.3.4. <i>Revisión y Consolidación de Informes</i>	48
2.2.5.3.5. <i>Notificación al solicitante sobre la Concesión de Frecuencias</i>	
<i>aprobada</i>	48
2.2.5.3.6. <i>Elaboración del Contrato</i>	48
2.2.5.4. Procedimiento para la Administración de Títulos Habilitantes (ATH)	
49	
2.2.5.5. Registro de Enlaces de Modulación Digital de Banda Ancha (MDBA)	
50	
2.3. DESCRIPCIÓN Y ANÁLISIS DE LAS NORMATIVAS	51
2.3.1. INSTRUCTIVO FORMULARIOS DE CONCESIÓN DE FRECUENCIAS	
51	
2.3.1.1. Formulario para Información Legal (RC-1A)	52
2.3.1.2. Formulario para Información Legal para Sistemas de Modulación	
Digital de Banda Ancha (RC-1B).....	52
2.3.1.3. Formulario para Información de la Infraestructura del Sistema de	
Radiocomunicaciones (RC-2A)	52
2.3.1.4. Formulario para Información de Antenas (RC-3A).....	53
2.3.1.5. Formulario para Patrones de Radiación de Antenas (RC-3B)	53
2.3.1.6. Formulario para Información de Equipamiento (RC-4A)	53
2.3.1.7. Formulario para el Tipo de Operación y Número de Estaciones de	
Servicios FM-RDV (RC-5A).....	54
2.3.1.8. Formulario para Sistemas de Modulación Digital de Banda Ancha	
Enlaces Punto-Punto (RC-9A).....	55
2.3.1.9. Formulario para Sistemas de Modulación Digital de Banda Ancha	
Sistemas Punto-Multipunto (RC-9B)	55

2.3.1.10. Formulario para Cálculos de Propagación (RC-13A).....	55
2.3.1.11. Formulario para Esquema del Sistema de Radiocomunicaciones (RC-14A).....	56
2.3.1.12. Formulario para Estudio Técnico de Emisiones de RNI (RC-15A).	56
2.3.2. NORMA PARA LA IMPLEMENTACIÓN Y OPERACIÓN DE SISTEMAS DE MODULACIÓN DIGITAL DE BANDA ANCHA (RESOLUCIÓN NO. 417-15- CONATEL-2005).....	57
2.4. ANÁLISIS DEL FORMATO DE ARCHIVO XML QUE SOPORTA EL SOFTWARE SPECTRAPLUS	57
2.5. HERRAMIENTAS DE DESARROLLO	60
2.5.1. VENTAJAS DE .NET FRENTE A J2EE	61
2.5.2. VENTAJAS DE J2EE FRENTE A .NET	61
2.5.3. WCF (WINDOWS COMMUNICATION FOUNDATION).....	61
2.5.4. WPF (WINDOWS PRESENTATION FOUNDATION)	62
2.5.5. ASP.NET.....	62
2.5.6. SERVIDOR WEB	63
2.5.7. TEAM FOUNDATION SERVICE.....	64
3. CAPÍTULO III.....	65
DISEÑO Y DESARROLLO DEL SISTEMA.....	65
3.1. INTRODUCCIÓN.....	65
3.2. DESCRIPCIÓN DE LOS ARTEFACTOS UTILIZADOS EN EL SISTEMA DISTRIBUIDO	65
3.2.1. HISTORIAS DE USUARIO.....	65
3.2.2. TEAM FOUNDATION SERVICE (TFS).....	66
3.2.2.1. Historias de usuario en TFS.....	68
3.2.2.2. Sprints en TFS	69
3.2.2.2.1. <i>Sprint 1</i>	70
3.2.2.2.2. <i>Sprint 2</i>	72
3.2.2.2.3. <i>Sprint 3</i>	73
3.2.2.2.4. <i>Sprint 4</i>	74
3.2.2.2.5. <i>Sprint 5</i>	75
3.2.2.3. Diagramas de TFS.....	76
3.2.2.4. Control de código en TFS	77

3.3.	COMPONENTES DEL SISTEMA DISTRIBUIDO.....	78
3.3.1.	APLICACIÓN DE ESCRITORIO	79
3.3.1.1.	Validaciones.....	81
3.3.1.2.	Cálculos Matemáticos	82
3.3.1.3.	Controles Personalizados	83
3.3.1.4.	PDF.....	85
3.3.1.5.	XML.....	86
3.3.2.	SERVICIO WEB REST	88
3.3.2.1.	Archivo de Configuración	90
3.3.2.1.1.	<i>Transferencia de Archivos</i>	93
3.3.2.2.	Interfaz del Servicio Web	96
3.3.2.3.	Implementación de la Interfaz de Servicio	101
3.3.2.3.1.	<i>Consulta a Base de Datos</i>	101
3.3.2.3.2.	<i>Consulta a SENESCYT</i>	105
3.3.2.3.3.	<i>Consulta de Geocodificación a Google Maps</i>	108
3.3.3.	APLICACIÓN WEB	110
3.3.3.1.	Adjuntar documentos en ASP.NET	111
3.3.3.2.	Envío de Correos desde ASP.NET	112
3.4.	CAPAS ARQUITECTÓNICAS DEL SOFTWARE.....	114
3.4.1.	CAPA APLICACIÓN.....	116
3.4.1.1.	Interfaz de la Aplicación de Escritorio	116
3.4.1.1.1.	<i>Guardar y reconstruir información en la Aplicación de escritorio</i> 125	
3.4.1.2.	Interfaz de la Aplicación Web.....	127
3.4.2.	CAPA DE SERVICIOS.....	128
3.4.2.1.	Interfaz.....	132
3.4.2.2.	Data Binding	134
3.4.3.	CAPA PROCESOS DE NEGOCIO	135
3.4.3.1.	Validación de RUC.....	135
3.4.3.2.	Validación de antenas.....	136
3.4.3.3.	Validación de equipos	137
3.4.3.4.	Validación Profesional Técnico	138
3.4.3.5.	Validación de Coordenadas Geográficas.....	139

3.4.4. CAPA DE DATOS	140
4. CAPÍTULO IV	144
IMPLEMENTACIÓN, PRUEBAS Y SEGURIDAD	144
4.1. INTRODUCCIÓN.....	144
4.2. IMPLEMENTACIÓN	144
4.2.1. DESPLIEGUE DEL PROTOTIPO DE SISTEMA DISTRIBUIDO.....	144
4.2.1.1. Despliegue de la Aplicación de Escritorio	145
4.2.1.2. Despliegue del Servicio y Aplicación Web	147
4.2.2. INSTRUCTIVOS	151
4.3. SEGURIDAD Y TOLERANCIA A FALLOS.....	152
4.3.1. SEGURIDAD Y TOLERANCIA A FALLOS A NIVEL DE SOFTWARE	153
4.3.1.1. Recuperación de la información.....	153
4.3.1.2. Cuentas de usuario.....	156
4.3.1.3. MD5	157
4.3.1.4. HTTPS	159
4.3.1.4.1. <i>Servicio WCF con HTTPS</i>	160
4.3.1.5. RAID 5	162
4.3.2. SEGURIDAD Y TOLERANCIA A FALLOS A NIVEL FÍSICO	163
4.4. PRUEBAS.....	164
4.5. COSTOS REFERENCIALES	165
4.5.1. Costos referenciales del hardware.....	165
4.5.2. Costos referenciales del Software Desarrollado	167
4.5.3. Costos referenciales de licenciamiento	170
5. CAPÍTULO V	172
CONCLUSIONES Y RECOMENDACIONES	172
5.1. CONCLUSIONES	172
5.2. RECOMENDACIONES.....	175
REFERENCIAS BIBLIOGRÁFICAS.....	178

ÍNDICE DE FIGURAS

Figura 1-1 Espectro Radioeléctrico	6
Figura 1-2 Tipo de Aplicaciones.....	12
Figura 1-3 Porcentaje de Uso de los Navegadores Web [9]	17
Figura 1-4 Partes de URI	18
Figura 1-5 Comunicación entre el cliente (navegador) y el servidor	19
Figura 1-6 Protocolos de servicios Web.....	21
Figura 1-7 Estándares de Servicios Web.....	25
Figura 1-8 Funcionamiento general de SSL/TLS [12]	29
Figura 1-9 Fases de Scrum [19].....	36
Figura 1-10 Roles de Scrum [19]	37
Figura 1-11 Elementos de Scrum [19].....	38
Figura 1-12 Reuniones de Scrum [19]	39
Figura 2-1 Organismos Reguladores de las Telecomunicaciones en el Ecuador .	41
Figura 2-2 Estructura Orgánica de la Secretaría Nacional de Telecomunicaciones	43
Figura 2-3 Proceso para el Otorgamiento de Títulos Habilitantes.....	45
Figura 2-4 Nombres de las variables de los campos de SPECTRAplus	59
Figura 2-5 Fragmento del archivo XML exportado del SPECTRAplus.....	60
Figura 3-1 Roles de Scrum	67
Figura 3-2 Tareas TFS.....	68
Figura 3-3 Tablero de TFS.....	69
Figura 3-4 Administración de <i>Sprints</i>	69
Figura 3-5 <i>Sprint</i> 5 en TFS.....	70
Figura 3-6 Diagrama de flujo acumulativo del TFS	77
Figura 3-7 Conexión Visual Studio a TFS	78
Figura 3-8 Diagrama de casos de uso	79
Figura 3-9 Validación del control <i>textbox</i>	81
Figura 3-10 Controles personalizados	83
Figura 3-11 Código XAML del control personalizado que forma parte del Formulario RC-2A	84
Figura 3-12 Control Personalizado para el Formulario RC-2A.....	84

Figura 3-13 Fragmento de código que muestra el uso de iText Sharp	85
Figura 3-14 Ejemplo de uso de XmlTextWriter	86
Figura 3-15 Diagrama del esquema XML para Sistemas MDBA	87
Figura 3-16 Diagrama del esquema para Servicio MT-RDV	87
Figura 3-17 Creación de un nuevo proyecto WCF	89
Figura 3-18 Elementos por defecto de un proyecto WCF	90
Figura 3-19 Consulta a un Servicio Web RESTful	93
Figura 3-20 Opción para refactorizar nombres	93
Figura 3-21 Valores TransferMode, MaxBufferSize y MaxReceivedMessageSize	95
Figura 3-22 Atributos behaviorConfiguration y bindingConfiguration	95
Figura 3-23 Fragmento de código que describe un contrato.....	97
Figura 3-24 Contrato de datos del Concesionario.....	97
Figura 3-25 Métodos y su comportamiento.....	99
Figura 3-26 Solicitud de datos del concesionario respuesta formato JSON	100
Figura 3-27 Solicitud de datos del concesionario respuesta formato XML	101
Figura 3-28 Algunas de las tablas que forman parte de la Base de Datos de SPECTRAplus	102
Figura 3-29 Diagrama de clases para manipular la Base de Datos	102
Figura 3-30 Clase Conexión	103
Figura 3-31 Clases que heredan de Conexion.....	103
Figura 3-32 Diagramas de clases para manipular datos.....	104
Figura 3-33 Fragmento de código para consultar información del concesionario	105
Figura 3-34 Consulta de títulos registrados SENESCYT	106
Figura 3-35 Captura de solicitud con Wireshark	106
Figura 3-36 Consulta a la SENESCYT mediante el método POST de HTTP	107
Figura 3-37 Descargar información desde Google Maps.....	109
Figura 3-38 Interfaz gráfica del Formulario RC-2A	109
Figura 3-39 Fragmento de código ASP.NET.....	111
Figura 3-40 Referencia Word.....	111
Figura 3-41 Fragmento de código de la generación del Memorando.....	112
Figura 3-42 Envío de correo en ASP.NET	113

Figura 3-43 Interfaz de envío de correo	114
Figura 3-44 Algunas plataformas para el desarrollo de aplicaciones SOA	114
Figura 3-45 Capas de la arquitectura SOA	115
Figura 3-46 Diagrama de clases de los formularios 1A y 5A (Servicio MT-RDV).....	117
Figura 3-47 Diagrama de clases de los formularios MDBA.....	118
Figura 3-48 Diagrama de clases de los formularios y clases RC-1A y RC-1B....	119
Figura 3-49 Diagrama de clases de los formularios RC-2A y RC-3A.....	121
Figura 3-50 Diagrama de clases de los formularios RC-4A y RC-9A.....	122
Figura 3-51 Diagrama de clases del formulario RC-5A.....	123
Figura 3-52 Diagrama de clases del formulario RC-9B.....	124
Figura 3-53 Diagrama de clases del almacenamiento de la Información.....	125
Figura 3-54 Fragmento de la clase PersistenciaBinaria.....	126
Figura 3-55 Métodos GuardarBinMdba y AbrirBinMdba	127
Figura 3-56 Diagrama de clases de la Aplicación Web.....	128
Figura 3-57 Archivo Web.config contract y serviceMetadata	130
Figura 3-58 Fragmento del archivo WSDL, Types	130
Figura 3-59 Fragmento del archivo WSDL, Message	130
Figura 3-60 Fragmento del archivo WSDL, operaciones en Service	131
Figura 3-61 Fragmento del archivo WSDL, operaciones en PortType	131
Figura 3-62 Interfaz con los métodos a publicar y la clase que los implementa .	133
Figura 3-63 WebHttpBinding usado en el servicio para transferencia de archivos.	135
Figura 3-64 Diagrama de secuencia RUC	136
Figura 3-65 Diagrama de secuencia Antenas	137
Figura 3-66 Diagrama de secuencia Equipos	138
Figura 3-67 Diagrama de secuencia de la validación del Profesional Técnico ...	139
Figura 3-68 Diagrama de secuencia de la validación de Coordenadas Geográficas	139
Figura 3-69 Diagrama de colaboración del RUC	140
Figura 3-70 Consultar información del RUC al servicio.....	141
Figura 3-71 Método en el Servicio que busca la información de un RUC	141
Figura 3-72 Diagrama de colaboración de la antena	142
Figura 3-73 Diagrama de colaboración del equipo	142

Figura 3-74 Métodos para consultar modelos e Equipos y Antenas	143
Figura 4-1 Algunos componentes presentes en el prototipo	145
Figura 4-2 Diagrama de despliegue aplicación de escritorio.....	146
Figura 4-3 Diagrama de despliegue servicio y aplicación web.....	148
Figura 4-4 Agregar un Rol o característica al servidor	148
Figura 4-5 Agregar Net Framework al Servidor	149
Figura 4-6 Proceso de instalación de la característica IIS y Net Framework	149
Figura 4-7 Agregar Net Framework desde <i>PowerShell</i>	150
Figura 4-8 Versión de Net Framework Incompatible	150
Figura 4-9 DefaultAppPool en el Administrador de Internet Information Services	151
Figura 4-10 Versiones de Net Framework disponibles.....	151
Figura 4-11 Descarga de Instructivos desde la aplicación de escritorio	152
Figura 4-12 Instanciar System.Windows.Threading.DispatcherTimer	154
Figura 4-13 Configurar eventos DispatcherTimer	154
Figura 4-14 Función que permite la recuperación de la información	155
Figura 4-15 Ventana principal de la aplicación web	156
Figura 4-16 Fragmento de código de la autenticación de la aplicación web	157
Figura 4-17 Función para el cálculo del Hash.....	158
Figura 4-18 Clase que implementa el Algoritmo MD5.....	159
Figura 4-19 Comunicación Cliente-Servidor mediante protocolo HTTPS	160
Figura 4-20 Seguridad de transporte web.config	161
Figura 4-21 Cambiar mexHttpBinding por mexHttpsBinding.....	161
Figura 4-22 Cambiar httpGetEnabled por httpsGetEnabled.....	161
Figura 4-23 Archivo Web.config alojado en el servidor	162
Figura 4-24 RAID 5 [56]	163

ÍNDICE DE TABLAS

Tabla 1-1 Telefonía Fija	3
Tabla 1-2 Telefonía Móvil.....	4
Tabla 1-3 Servicio de Internet.....	4
Tabla 1-4 Servicios Portadores.....	5
Tabla 1-5 Diferencia entre Aplicación Web y Aplicación de Escritorio	14
Tabla 1-6 XML vs HTML [10]	22
Tabla 1-7 Métodos HTTP	27
Tabla 1-8 Comparación de Metodologías [17]	34
Tabla 3-1 Ejemplo de historia de usuario.....	66
Tabla 3-2 Estadísticas TFS.....	70
Tabla 3-3 <i>Sprint</i> 1	71
Tabla 3-4 <i>Sprint</i> 2	72
Tabla 3-5 <i>Sprint</i> 3	74
Tabla 3-6 <i>Sprint</i> 4	75
Tabla 3-7 <i>Sprint</i> 5	76
Tabla 3-8 Formularios de la Aplicación de Escritorio	80
Tabla 3-9 Algunos subelementos de endpoint Behaviors	92
Tabla 3-10 Algunos atributos de un elemento <ServiceBehaviors>.....	94
Tabla 3-11 Algunas propiedades de un elemento webHttpBinding.....	94
Tabla 3-12 Espacios de nombres para definir un contrato de servicio.....	96
Tabla 3-13 Algunas clases del <i>Namespace</i> System.ServiceModel.Web	98
Tabla 3-14 Propiedades de WebGet y WebInvoke	99
Tabla 3-15 Parámetros URI	100
Tabla 4-1 Requerimientos para la máquina que use la aplicación de escritorio .	146
Tabla 4-2 Requerimientos del servidor	147
Tabla 4-3 Seguridades Físicas SENATEL	164
Tabla 4-4 Comparación de tiempo de elaboración e ingreso de los trámites.....	165
Tabla 4-5 Características y costos del servidor	166
Tabla 4-6 Características y costos de discos duros.....	166
Tabla 4-7 Características y costos de computador	167
Tabla 4-8 Costos Referenciales.....	169
Tabla 4-9 Costos referenciales Licenciamiento	170

RESUMEN

La mejora continua es parte fundamental en toda organización cuyo propósito sea dar un servicio de calidad, el perfeccionamiento de cualquier proceso o servicio radica principalmente en la implementación de tecnologías y productos de vanguardia para satisfacer nuevos requerimientos de los usuarios.

Actualmente se requiere con mayor asiduidad de sistemas que permitan la automatización de la información, por lo cual es imperativo el desarrollo e implementación de nuevas aplicaciones o utilidades sobre sistemas ya existentes, es decir, adaptar antiguos sistemas a nuevas necesidades.

En virtud de lo antes mencionado el presente Proyecto plantea el desarrollo de un Prototipo de Sistema Distribuido para la Secretaría Nacional de Telecomunicaciones (SENATEL), con el fin de coadyuvar en los procesos de Otorgamiento de Títulos Habilitantes del Servicio Móvil Terrestre Radio de Dos Vías y Registros de Enlaces MDBA.

Los concesionarios no cuentan con una herramienta informática que ayude a realizar y validar sus trámites antes de ingresarlos a la SENATEL. Actualmente los concesionarios deben entregar sus trámites impresos para que los analistas de la SENATEL verifiquen toda la información técnica y legal para posteriormente ingresarla manualmente al software propietario denominado SPECTRAplus.

El presente Proyecto busca automatizar este proceso tanto del lado del concesionario como del lado del analista, utilizando un sistema distribuido conformado por tres componentes, en el cual el concesionario podrá realizar sus trámites a través de una interfaz amigable y enviarlos a la SENATEL, mientras que el analista podrá visualizar e importar la información al SPECTRAplus.

En este documento se describe la base teórica utilizada, los requerimientos de los usuarios, el desarrollo e implementación del Prototipo de Sistema Distribuido basado en la metodología de desarrollo Scrum, y finalmente las pruebas y resultados de operación de los tres componentes.

PRESENTACIÓN

Para generar nuevas concesiones, renovaciones, modificaciones o eliminaciones de una determinada concesión de frecuencias, la SENATEL ha establecido formularios que deben ser ingresados con determinadas especificaciones legales y técnicas de acuerdo al requerimiento del concesionario. Actualmente el ingreso de estos trámites se lo realiza físicamente a través del Centro de Atención al Usuario (CAU) de la SENATEL, para lo cual se presentan los formularios impresos, a continuación una versión escaneada del documento se distribuye entre los analistas para que validen la información para posteriormente ingresarla manualmente al sistema SPECTRAplus, obteniendo finalmente los registros y títulos habilitantes.

El presente Proyecto designado: "Diseño e implementación de un prototipo de Sistema Distribuido para los procesos del Servicio Móvil Terrestre y de los Sistemas de Modulación Digital de Banda Ancha para la Secretaria Nacional de Telecomunicaciones (SENATEL)" ofrece una herramienta informática que podría ayudar en el proceso de elaboración de trámites por parte del concesionario y el ingreso de la información técnica por parte de los analistas de la SENATEL al software SPECTRAplus.

Entre los objetivos específicos planteados están:

- Analizar los procesos administrativos y técnicos de la SENATEL empleados para el Otorgamiento y Administración de Títulos Habilitantes del Servicio Móvil Terrestre y el Registro de Sistemas MDBA.
- Establecer un esquema adecuado basado en XML que sea compatible con el esquema del software SPECTRAplus, para poder importar los datos que el prototipo genere, a dicho software.
- Diseñar e implementar un Sistema Distribuido manteniendo estándares de seguridad y tolerancia a fallos, constituido por tres componentes:
 - ✓ Servicios Web basados en SOA (*Service Oriented Architecture*) para validar la información.

- ✓ Aplicación de escritorio que permita el ingreso de los datos offline al concesionario.
- ✓ Aplicación web para presentar los datos al analista de la SENATEL.
- Desarrollar los procesos de automatización y validación para el ingreso al sistema de la información técnica.

El desarrollo de este Proyecto se lo presenta en cinco capítulos, como se explica a continuación:

En el **Capítulo I**, se realiza una descripción de los servicios de telecomunicaciones, conceptos de Servicio Móvil Terrestre, sistemas VHF/UHF, Radio de Dos vías y Sistemas de Modulación Digital de Banda Ancha (MDBA). Se presenta principios de funcionamiento de los Servicios Web, archivos XML y protocolos de seguridad TLS y SSL.

En el **Capítulo II**, se realiza un estudio de los procesos internos relacionados al Otorgamiento y Administración de Títulos Habilitantes y a los Registros de Sistemas MDBA para automatizar los procesos y cumplir con los objetivos propuestos. Finalmente se justifican todas las herramientas utilizadas para la creación y desarrollo del software.

En el **Capítulo III**, se realiza el diseño de los tres componentes del Sistema Distribuido sobre un único Entorno de Desarrollo Integrado (IDE), utilizando la metodología de desarrollo SCRUM.

En el **Capítulo IV**, se explica el procedimiento de desarrollo, implementación y despliegue llevado a cabo para conseguir un funcionamiento adecuado y seguro. Además de costos referenciales.

En el **Capítulo V**, se plantea conclusiones y recomendaciones en base a la experiencia obtenida en la realización del presente proyecto.

Al final de este documento se encuentran los Anexos A y B, los Anexos C, D, E, F, G, H, I, J y K se encuentran en el DVD adjunto.

ANEXO A: ANÁLISIS DEL FORMATO XML

ANEXO B: HISTORIAS DE USUARIO

ANEXO C: FUNCIONES PARA EL CÁLCULO DE LA DISTANCIA AZIMUT Y
ÁNGULO DE ELEVACIÓN

ANEXO D: MANUALES DE USUARIO DE MDBA

ANEXO E: MANUALES DE USUARIO DE MT-RDV

ANEXO F: CONFIGURACIÓN DE HTTPS (Certificado Autofirmado)

ANEXO G: PRUEBAS DE CONFIGURACIÓN DE HTTPS

ANEXO H: IMPLEMENTACION DE RAID 5

ANEXO I: MANUAL DE PRUEBAS

ANEXO J: COTIZACIONES

ANEXO L: INSTALADORES

ANEXO K: CÓDIGO

CAPÍTULO I

1. MARCO TEÓRICO

1.1. INTRODUCCIÓN

En este capítulo se realiza una descripción de los Servicios de Telecomunicaciones, orientándose a los procesos para los cuales se desarrolla el presente proyecto: Servicio Móvil Terrestre, sistemas VHF/UHF (*Very High Frequencies/Ultra High Frequencies*) Radio de dos vías (MT-RDV) y Sistemas de Modulación Digital de Banda Ancha (MDBA).

Se presentan principios de funcionamiento de los Servicios Web, REST (*Representational State Transfer*), SOA (*Service Oriented Architecture*), protocolos de seguridad TLS/SSL (*Transport Layer Security/Secure Sockets Layer*).

Finalmente se realiza una introducción a la metodología de desarrollo de software SCRUM detallando las características principales por las cuales se la seleccionó para el desarrollo de este proyecto.

1.2. SERVICIOS DE RADIOCOMUNICACIONES

El actual Reglamento de Radiocomunicaciones emitido por el Concejo Nacional de Telecomunicaciones [1], establece que los servicios de radiocomunicaciones implican: transmisión, emisión o recepción de ondas radioeléctricas para fines específicos de telecomunicación.

Debido a la expansión y convergencia de sistemas de radiocomunicaciones, la UIT¹, como organismo internacional de normalización ha diferenciado y definido a cada uno de los servicios a fin de estandarizar las distintas operaciones de radiocomunicación.

¹ **UIT (Unión Internacional de Telecomunicaciones)** es el organismo especializado de Telecomunicaciones de la Organización de las Naciones Unidas, encargado de regular las telecomunicaciones a nivel internacional entre las distintas administraciones y empresas operadoras.

Dentro de este contexto en nuestro país se manejan varios servicios; como: fijos, móviles, radiodifusión, seguridad, especiales, aficionados, exploración entre otros, con el fin de promover el desarrollo de los servicios de telecomunicaciones para beneficio de la comunidad.

1.2.1. SERVICIOS DE TELECOMUNICACIONES

Los servicios de telecomunicaciones implican la transmisión y recepción de información para fines científicos, industriales, gubernamentales, etc. En el Ecuador los Servicios de Telecomunicaciones están definidos y normados por la Ley Especial de Telecomunicaciones reformada [2] pudiendo ser estos: finales, portadores o de valor agregado.

Servicios finales realizan toda la comunicación entre usuarios finales, como por ejemplo:

- **Telefonía fija** permite la comunicación bidireccional entre usuarios fijos.
- **Telefonía móvil (Servicio Móvil Avanzado)** permite la comunicación bidireccional entre usuarios móviles.

Servicios portadores brindan la capacidad para la transmisión de las señales entre los puntos de terminación de una red, que pueden hacer uso de redes conmutadas o no conmutadas para los enlaces de los puntos de terminación.

Servicios de valor agregado son los servicios que siendo finales o portadores añaden otras facilidades o satisfacen nuevas necesidades, como por ejemplo:

- **Acceso a Internet** es un servicio de valor agregado, que permite a los usuarios conectarse a la red de Internet.
- **Cibercafés** son locaciones de acceso público que brindan servicios de Internet a cambio de una tarifa fija.

1.2.2. ESTADÍSTICAS DEL SECTOR

Dentro del esquema de Servicios de Telecomunicaciones en el Ecuador, cabe mencionar que los más destacados son: telefonía fija, telefonía móvil, servicios portadores y acceso a Internet. Las Tabla 1-1, Tabla 1-2, Tabla 1-3 y Tabla 1-4 presentan las estadísticas del crecimiento de estos servicios en los últimos años. Esta información fue tomada de la página web de la Superintendencia Nacional de Telecomunicaciones (SUPERTEL) [3].

Telefonía Fija: Se cuenta con siete entidades que proveen al país de servicio de telefonía fija: CNT EP., ETAPA EP, LINKOTEL S.A., ECUADORTELECOM S.A., SETEL S.A., GLOBAL CROSSING S.A. y GRUPOCORIPAR S.A. (Ver Tabla 1-1).

Según [4] hasta el año 2013 el 39.6% de los hogares ecuatorianos poseía una línea telefónica fija, el crecimiento es relativamente bajo en comparación con el servicio móvil avanzado.

Telefonía Fija	
Año	Número de Líneas Fijas
2006	1'775.231
2007	1'823.050
2008	1'906.401
2009	2'011.605
2010	2'133.642
2011	2'219.425
2012	2'295.592
2013	2'404.008

Tabla 1-1 Telefonía Fija

Telefonía Móvil: Existen tres operadoras OTECEL, CONECEL y CNT EP, las cuales han tenido una gran acogida en el país (Ver Tabla 1-2).

Según [4] hasta el año 2013 el 86.4% de los hogares ecuatorianos poseía un teléfono celular.

Telefonía Móvil	
Año	Número de Líneas Móviles
1994	18.920
1998	242.812
2002	1'572.142
2004	3'439.349
2006	8'126.397
2008	10'934.712
2010	14'986.886
2012	16'816.018
2013	17'621.178

Tabla 1-2 Telefonía Móvil

Servicio de Internet: Actualmente el Internet es el principal servicio de telecomunicación, existiendo solo en nuestro país, cerca de 176 proveedores de Internet. Como se puede apreciar en la Tabla 1-3 desde el año 2010 se muestra un rápido incremento en el número de usuarios. La SUPERTEL, estima que por cada cuenta conmutada existen 4 usuarios.

Según [4] hasta el año 2013 el 28.3% de los hogares ecuatorianos tenía acceso a Internet.

Internet	
Año	Número de Usuarios
2008	1'627.916
2009	1'977.687
2010	4'913.359
2011	5'499.193
2012	9'011.105
2013	11'085.782

Tabla 1-3 Servicio de Internet

Servicios Portadores: Existen cerca de 21 operadores de servicios portadores (CNT EP, SURATEL S.A., MEGADATOS, PUNTONET, etc.) que cubren las

necesidades de varios usuarios a nivel nacional, las estadísticas muestran un rápido crecimiento desde el año 2009 (Ver Tabla 1-4).

De modo general las citadas cifras muestran un acelerado desarrollo de los servicios de telecomunicaciones en nuestro país, motivo por el cual es necesario desarrollar ciertos procesos que permiten operar a las distintas entidades dedicadas al servicio de las telecomunicaciones y de la división del espectro radioeléctrico en el territorio nacional.

Servicios Portadores	
Año	Número de Usuarios
2009	291.469
2010	380.394
2011	567.943
2012	719.523
2013	931.215

Tabla 1-4 Servicios Portadores

1.2.3. ESPECTO RADIOELÉCTRICO

Debido a la creciente demanda de servicios de telecomunicaciones (televisión, radiodifusión, seguridad, defensa, emergencias, transporte e investigaciones científicas) surge la necesidad de administrar el espectro radioeléctrico de manera eficiente teniendo en cuenta lo siguiente:

- Que pueda ser utilizado por el mayor número de usuarios sin problemas de interferencias.
- Que presente flexibilidad ante nuevas tecnologías y servicios.
- Además asegurar que exista la disponibilidad del espectro para servicios comunitarios de seguridad, salud, entre otros.

Esto implica que la asignación de frecuencias sea un proceso complejo, por lo cual la administración del espectro radioeléctrico depende de las regulaciones de cada país, que realiza la asignación teniendo en cuenta el Plan de Frecuencias de la UIT.

Según la UIT, el espectro radioeléctrico es el conjunto de ondas electromagnéticas, cuya frecuencia se fija convencionalmente por debajo de los 3.000 GHz, que se propagan por el espacio sin guía artificial.

El espectro radioeléctrico es un recurso natural, de carácter limitado, que constituye un bien de dominio público, sobre el cual el Estado ejerce su soberanía. Se divide en nueve bandas de frecuencia desde los 3 kHz, hasta los 300 GHz de frecuencia, que son asignadas a varios servicios, dependiendo de los diferentes modos de propagación de las ondas radioeléctricas que pueden ser por onda de superficie, onda ionosférica, onda espacial-tropósfera, entre otras.

Por otra parte la asignación de bandas de frecuencias, se divide en: frecuencias compartidas donde varios usuarios comparten las mismas frecuencias, o frecuencias exclusivas en la cual un único usuario posee asignada una o más frecuencias.

En la Figura 1-1 se muestra la división del Espectro Radioeléctrico y sus usos.

300-3000(Hz)	3-30(KHz)	30-300(KHz)	0,3-3(MHz)	3-30(MHz)	30-300(MHz)	0,3-3(GHz)	3-30(GHz)	30-300(GHz)
VF	VLF	LF	MF	HF	VHF	UHF	SHF	EHF
								
Frecuencias de Voz	Muy Baja Frecuencia	Baja Frecuencia	Frecuencia Media	Alta Frecuencia	Muy Alta Frecuencia	Ultra Alta Frecuencia	Super Alta Frecuencia	Frecuencia Extremadamente Alta
	Usada para Navegación a larga distancia Comunicación Submarina	Radiofaros de Comunicación Marina Navegación Larga Distancia	Radiodifusión AM Localización de direcciones	Radioaficionado Telefonía Comunicaciones aéreas Comunicaciones Marítimas de larga distancia Comunicaciones Militares	Televisión VHF Radiodifusión FM Comunicaciones aéreas AM	Televisión UHF Telefonía celular Radar Enlaces microonda	Microondas Enlaces de Radar Comunicación satelital	Radar Satélites Experimentales

Figura 1-1 Espectro Radioeléctrico

1.2.4. SISTEMAS VHF y UHF

Dentro del espectro radioeléctrico se encuentran las bandas: VHF (*Very High Frequencies*) y UHF (*Ultra High Frequencies*), que generalmente se usan para

enlaces de radio a corta distancia, televisión, frecuencia modulada enlaces de radio (radio de dos vías), radar, ayuda a la navegación aérea, entre otros.

La asignación dependerá principalmente del tipo de sistema o servicio radioeléctrico y de la saturación de la banda de frecuencia en la zona geográfica en la que se desee instalar dicho sistema o servicio.

En VHF y UHF predomina el modo de propagación por línea de vista directa a través de la ionósfera, por lo cual se tiene un máximo alcance de 200 Km. de radio. Cabe mencionarse que estas bandas se encuentran casi saturadas debido a la gran cantidad de aplicaciones que se emplean.

Para mayor detalle de las asignaciones de frecuencias se puede consultar el Plan Nacional de Frecuencias 2012 [5]. A continuación se mencionan algunas aplicaciones de estas bandas.

1.2.4.1. Aplicaciones en el rango VHF (30-300 MHz)

Esta banda también es conocida como la banda de las ondas métricas, hay diversas aplicaciones en VHF, como: fijo, móvil, móvil marítimo, móvil aeronáutico, radiodifusión, radioaficionados, aficionados por satélite, móvil por satélite, radioastronomía, radionavegación aeronáutica, meteorología por satélite, investigación espacial, radiolocalización, radionavegación por satélite y operaciones espaciales [5].

1.2.4.2. Aplicaciones en el rango UHF (300-3000 MHz)

Esta banda también es conocida como la banda de las ondas decimétricas, hay diversas aplicaciones en UHF ya que por sus características físicas de propagación brinda mayores alcances sin atenuación de la señal, razón por la cual considerables servicios de radiocomunicación requieren transmitir en este espacio de frecuencias.

UHF es usado para los servicios: fijo, móvil, móvil aeronáutico, radiodifusión, radiolocalización, radionavegación, aficionados, satelital, móvil por satélite, ayudas

a la meteorología, exploración de la tierra por satélite, investigación espacial, frecuencias patrón y señales horarias por satélite, meteorología por satélite, operaciones espaciales, radioastronomía, radio determinación por satélite, radionavegación aeronáutica, radionavegación por satélite, operaciones espaciales entre otras [5].

1.2.4.3. Sistemas de Radio de Dos Vías

De acuerdo al Reglamento de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones y el Plan Nacional de Frecuencias [5] los sistemas de Radio de Dos Vías están definidos de la siguiente manera:

“Son sistemas de radiocomunicaciones que transmiten y reciben comunicaciones en modo simplex² o semidúplex³ y utilizan tecnología push to talk⁴”

Radio de dos vías es un tipo de servicio de UHF/VHF, especialmente para la transmisión de voz, este servicio proporciona un medio de comunicación de mayor calidad, además de ser más confiable, seguro y económico, en comparación con las soluciones de redes celulares o públicas que existen en el mercado. Este servicio es usado por las cooperativas de taxis, empresas de mensajería, empresas de seguridad, constructoras, entre otras.

1.2.5. SISTEMAS DE MODULACIÓN DIGITAL DE BANDA ANCHA (MDBA)

Se realiza determinado tipo de modulación para que una señal pueda ser transmitida, en el caso de la modulación digital se deben transformar los símbolos digitales en formas de onda que sean compatibles con el medio en el cual se van a transmitir. Se modula para: compartir el canal que en este caso es el aire, poder usar antenas de tamaños relativamente pequeños, reducir las interferencias o ruidos, obtener mayor alcance y para conseguir transmisiones más robustas.

² **Simplex:** Es un modo de transmisión, permite que la información discorra en un solo sentido y de forma permanente.

³ **Semidúplex:** En este modo la transmisión fluye en los dos sentidos, pero no simultáneamente, solo una de las dos estaciones del enlace punto a punto puede transmitir.

⁴ **Push to Talk:** Pulsar para Hablar, es un método para hablar en líneas simplex o semidúplex de comunicación, apretando un botón para transmitir y liberándolo para recibir.

De acuerdo a la Resolución N° 417-15-CONATEL-2005 [6], los sistemas de Modulación Digital de Banda Ancha se caracterizan por:

- Distribución de la energía media de la señal de transmisión dentro de una anchura de banda mucho mayor que la convencional y con un nivel bajo de potencia.
- Utilización de técnicas de modulación que proporcionan una señal resistente a las interferencias.
- Permitir a diferentes usuarios utilizar la misma banda de frecuencias simultáneamente.
- Coexistir con sistemas de Banda Angosta, lo que hace posible aumentar la eficiencia de utilización del Espectro Radioeléctrico.
- Operar en bandas de frecuencias inscritas en el cuadro de atribución de bandas de frecuencias.

Según el Plan Nacional de Frecuencias 2012 [5]:

“En las bandas 902-928 MHz, 2400 – 2483.5 MHz, 5150 – 5350 MHz, 5470 – 5725 MHz, 5725 – 5850 MHz, también operan sistemas de Modulación Digital de Banda Ancha y enlaces auxiliares de radiodifusión sonora que utilizan técnicas de modulación digital de banda ancha sin protección contra interferencias perjudiciales.”

Los sistemas MDBA pueden operar en varias configuraciones, siendo las más utilizadas:

- Sistemas Punto-Punto⁵
- Sistemas Punto-Multipunto⁶
- Sistemas Móviles⁷

⁵ **Sistemas Punto-Punto** Son sistemas de radiocomunicaciones que permiten enlazar dos estaciones fijas distantes, empleando antenas direccionales en ambos extremos.

⁶ **Sistemas Punto-multipunto** Son sistema de radiocomunicaciones que permiten enlazar una estación fija central con varias estaciones fijas distantes, las estaciones fijas distantes emplean antenas direccionales para comunicarse con la fija central.

⁷ **Sistemas Móviles** Son sistemas de radiocomunicaciones que permiten enlazar una estación fija central con una o varias estaciones móviles. Las estaciones móviles deberán contar con equipamiento de comunicación con la estación fija central como un sistema de Wi-Fi.

1.2.5.1. Aplicaciones MDBA

Existen diversas aplicaciones en donde se puede emplear Modulación Digital de Banda Ancha como:

- Radioenlaces
- Telefonía inalámbrica y enlaces satelitales
- Sistema de Posicionamiento Global (GPS)
- 802.118, estándares asociados y Bluetooth
- Sistemas de comunicaciones militares
- Sistemas celulares
- Telemetría y Control Remoto (Sensores Industriales, Control de Procesos, Monitoreo de Medio Ambiente, Telemetría Médica, SCADA9)
- Terminales de datos portátiles (Puntos de Venta, Órdenes de Pedidos, Administración de Inventarios, Verificación de Tarjetas de Crédito)
- Redes LAN10 y WLAN11 (conexión de computadores de oficina a alta velocidad, acceso a Internet, etc.)
- PBX12 y sistemas de radiolocalización

1.3. SERVICIOS WEB

1.3.1. INTRODUCCIÓN

El Internet es una herramienta potente que permite acceder de manera simple a una gran cantidad de recursos informáticos, presentando la información en forma de hojas electrónicas, lo que se conoce también como la *web*, la cual mantiene en gran parte el funcionamiento de Internet e incorpora nuevas ideas, como por ejemplo, posibilita la lectura universal, desligando al usuario de la necesidad de

⁸ **802.11** y sus estándares asociados definen el uso de capa uno y dos del modelo ISO/OSI para redes de área local inalámbricas.

⁹ **SCADA** (Supervisory Control And Data Acquisition) software para controlar procesos industriales a distancia.

¹⁰ **LAN** (Local Area Network) una LAN es una red que conecta los computadores en un área relativamente pequeña y predeterminada.

¹¹ **WLAN** (Local Area Network) es una LAN inalámbrica.

¹² **PBX** (Private Branch Exchange) central telefónica privada conectada a la red de telefonía pública.

manipular innumerables programas y conocer detalles de sistemas operativos para efectuar una búsqueda en Internet.

El Internet es la interconexión de dispositivos a nivel mundial, para permitir la comunicación entre ellos se establecieron protocolos estandarizados, siendo necesario además estandarizar las plataformas y lenguajes de programación existentes para poder compartir los recursos, dando paso al surgimiento de los servicios Web como una opción para lograr este objetivo, haciendo uso de varias tecnologías que se mencionarán más adelante; los servicios Web pueden usarse sobre cualquier protocolo de Internet.

Los primeros intentos por unificar estos elementos fueron fallidos, algunos de ellos tuvieron como principal limitante el hecho de ser dependientes de la implementación de sus respectivos vendedores, otra gran dificultad fue el uso de RPC¹³ (*Remote Procedure Call*) para las llamadas entre nodos; el estilo de programación basado en interfaces que promueve RPC a pesar de presentar muchos beneficios, posee desventajas entre ellas, no poder usar direcciones de un proceso como referencia en un proceso remoto, además la semántica de las llamadas asociadas y la transparencia relacionada a RPC. Otro punto conflictivo en la estandarización es el acceso a recursos que son incompatibles entre sí y que además pueden ser de diversos tipos.

Por estas razones toda la información existente en Internet debía identificarse claramente, siendo precisamente esa la función que desempeñaría el Localizador de Recursos Uniformes (URL).

1.3.2. DEFINICIÓN DE SERVICIOS WEB

Se puede entender a un Servicio Web como un conjunto de funciones que ante el usuario se presentan como una sola entidad y se publican en red para ser usadas por medio de otros programas, los servicios Web permiten el acceso rápido a los recursos digitales disponibles en la red mundial.

¹³ **Llamada a Procedimiento Remoto (RPC)**, es un protocolo que permite a un programa ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

Según la W3C¹⁴, un “Servicio Web es un sistema de software diseñado para permitir interoperabilidad máquina a máquina en una red” [7].

Estos avances en interoperabilidad marcaron un gran desarrollo y promovieron el uso de Internet, de esta manera algo que comenzó como un proyecto de investigación gubernamental, proporciona hoy en día a la humanidad un importante medio de comunicación.

1.3.3. EVOLUCIÓN DE LAS APLICACIONES INFORMÁTICAS

En un principio había aplicaciones monolíticas o mejor conocidas como aplicaciones de escritorio, pero el deseo de mejorar la capacidad de los sistemas para satisfacer nuevos requerimientos ha impulsado el desarrollo de los sistemas hasta obtener lo que ahora se conoce como aplicaciones de n-niveles (Ver Figura 1-2).

El diseño de las aplicaciones generalmente conlleva a la división de la aplicación en niveles; la interfaz de usuario, la lógica de la aplicación y el acceso a los datos.



Figura 1-2 Tipo de Aplicaciones

1.3.3.1. Aplicaciones Monolíticas

Estas son las primeras aplicaciones en la cual la interfaz, los datos y la lógica de la aplicación están en un solo computador y generalmente se tienen estructuras de

¹⁴ **W3C** (World Wide Web Consortium), es un consorcio internacional que produce recomendaciones para la World Wide Web.

software fuertemente acopladas, como los juegos monousuario y los sistemas operativos.

1.3.3.2. Aplicaciones Cliente-Servidor

Estos son sistemas donde la interfaz y parte de la lógica de la aplicación están en la máquina del usuario, y los datos y parte de la lógica en un servidor, al cual se accede a través de protocolos de comunicación, es decir el cliente realiza peticiones al servidor, generándose así un balance de carga operacional y una mejora en el desempeño del sistema.

1.3.3.3. Aplicaciones Web

En las aplicaciones web el usuario solo cuenta con la interfaz web, y realiza sus peticiones al servidor web, el cual contiene la lógica de la aplicación y la base de datos. Una aplicación web está constituida por páginas web, que son archivos de texto o formularios HTML¹⁵, por medio de los cuales el usuario realiza peticiones que se envían a través del protocolo HTTP¹⁶ hacia el servidor web en donde está alojada la aplicación web. Existen dos tipos de páginas web:

- **Páginas Web Estáticas:** son páginas fijas que se almacenan de forma local en el servidor web.
- **Páginas Web Dinámicas:** Cada vez que se realiza alguna petición, se construye la página en tiempo real. Se usan menos recursos pero el tiempo de ejecución puede ser mayor.

1.3.3.3.1. Diferencia entre Aplicación Web y Aplicación de Escritorio

Existen varias diferencias entre una aplicación de escritorio y una aplicación web, en la Tabla 1-5 se analizan las ventajas y desventajas de este tipo de aplicaciones

¹⁵ **HyperText Markup Lenguaje** es un lenguaje utilizado para describir páginas web.

¹⁶ **Protocolo de Transferencia de Hipertexto** (HTTP) es un protocolo de nivel de aplicación para los sistemas de información distribuidos colaborativos, hipermedia.

Tipo de aplicación Características	Aplicación de Escritorio	Aplicación Web
Portabilidad	Debe ser instalado en cada dispositivo	Solo requiere de un navegador
Actualizaciones	Se debe instalar en cada máquina.	Es transparente para el usuario
Usabilidad (Interfaz de usuario)	Se dispone de muchos componentes (<i>textbox, label, etc.</i>)	Se deben instalar <i>plugins</i> si se desean manejar nuevos componentes.
Seguridad	Es relativamente sencilla	Es más complicado ya que la aplicación se encuentra en la web
Multiplataforma e Interoperabilidad	Generalmente solo funcionan bajo el sistema para el que fueron diseñadas	Se puede acceder a la aplicación bajo cualquier sistema operativo.

Tabla 1-5 Diferencia entre Aplicación Web y Aplicación de Escritorio

Cada una de estas aplicaciones tiene ventajas y desventajas por ejemplo en las aplicaciones web es más sencillo realizar actualizaciones y mantenimiento, además de la facilidad en la escalabilidad y movilidad, aunque en el tema de la seguridad se deba tener más precauciones. En cuanto a las aplicaciones de escritorio la principal ventaja radica en la velocidad, seguridad y en el uso de diversos componentes y entornos visuales disponibles, aunque si se requiere de alguna actualización o modificación se debe realizar en cada máquina en la que este instalada la aplicación, incluso si esta aplicación trabaja con datos a través de la web.

1.3.3.4. Aplicaciones de n-Niveles

Las aplicaciones multiniveles son una evolución de las aplicaciones cliente-servidor, al desarrollar una aplicación basada en una arquitectura multinivel el objetivo es que la lógica de diseño este claramente separada de la lógica de negocio, este tipo de aplicaciones distribuye los datos y los separa de la presentación al usuario, las comunicaciones se pueden dar en una red de área local o usando Internet.

- **El nivel de presentación** consiste en las aplicaciones cliente que acceden al servidor intermedio, en esta arquitectura no existe la posibilidad de que un

cliente se comunice directamente con el servidor de base de datos, varios clientes pueden acceder al mismo servidor de aplicación permitiendo mejorar la redundancia.

- **El nivel de negocio** es en donde se encuentran los programas a ejecutarse, se reciben las peticiones del usuario y se envían las respuestas después de ser procesadas, en este nivel se establecen todas las reglas, se comunica con el nivel de presentación, recibe las solicitudes y presenta los resultados, se comunica con el nivel que maneja los datos, para solicitar al gestor de base de datos realice las tareas indicadas por el cliente ya sea una consulta o almacenar información.
- **El nivel de datos** es donde reside la información y es el encargado de acceder a los datos. Puede estar formado por un gran número de gestores de bases de datos que almacenan la información y reciben indicaciones de operaciones desde el nivel de negocio.

Distribuyendo una gran cantidad de procesos o tareas pesadas en los niveles intermedios se puede desarrollar aplicaciones de cliente más simples ocupando menos recursos, de esta manera se permite que la aplicación se instale en máquinas de bajo rendimiento. Las bases de datos solo tienen una conexión directa al servidor de aplicaciones, de esta forma se evitan problemas de concurrencia o latencia de datos que suelen ocurrir en las aplicaciones cliente.

1.3.4. HIPERTEXTO Y HTML

El hipertexto constituye un concepto fundamental para la fácil compartición de recursos por medio de Internet, es la forma de comunicación de la red que marca la independencia entre el usuario y la tecnología que se use para realizar una búsqueda, impulsando la lectura universal ya que los recursos (texto, imágenes, etc.) son diseñados para poder explorarse con facilidad.

Según [8]:

“El hipertexto es una herramienta de software con estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos.”

El hipertexto, presenta dos formas de realizar una búsqueda: el *surf* o modo libre que consiste en seguir los *links* o enlaces hasta conseguir un resultado satisfactorio o la asistida o indexada en la que participan los computadores, mediante el uso de índices creados automáticamente que simplifican la búsqueda.

HTML (*HyperText Markup Language*), no es un lenguaje de programación como tal. HTML está diseñado para estructurar texto y presentarlo como hipertexto, se lo utiliza principalmente en el desarrollo de páginas web.

Como HTML no posee un compilador, los errores de sintaxis no se detectan ni son visualizados, sin embargo solo hace falta un procesador de texto para desarrollarlo, mismo que la mayoría de sistemas operativos incorporan.

HTML se basa principalmente en la referencia, es decir, si un objeto existe en una ubicación externa es necesario para el desarrollo de una página solamente se agrega su referencia como texto, estas referencias se escriben en archivos de texto haciendo uso de etiquetas que consisten en breves instrucciones de comienzo y final, por lo general se guardan con extensión `.htm` o `.html`.

Al solicitar un archivo HTML en la web, estos son presentados por los programas conocidos como navegadores o *browsers*, que son los responsables de interpretar el código HTML, unir los elementos y ponerlos a disposición del usuario, para que finalmente pueda “navegar en la red”, es decir pasar de una página a otra haciendo uso de las facilidades del sistema hasta encontrar la información o página deseada.

En la Figura 1-3 se aprecian algunos de los navegadores web más conocidos y la aceptación que tienen dentro de los usuarios de Internet desde el año 2009 hasta el 2013.

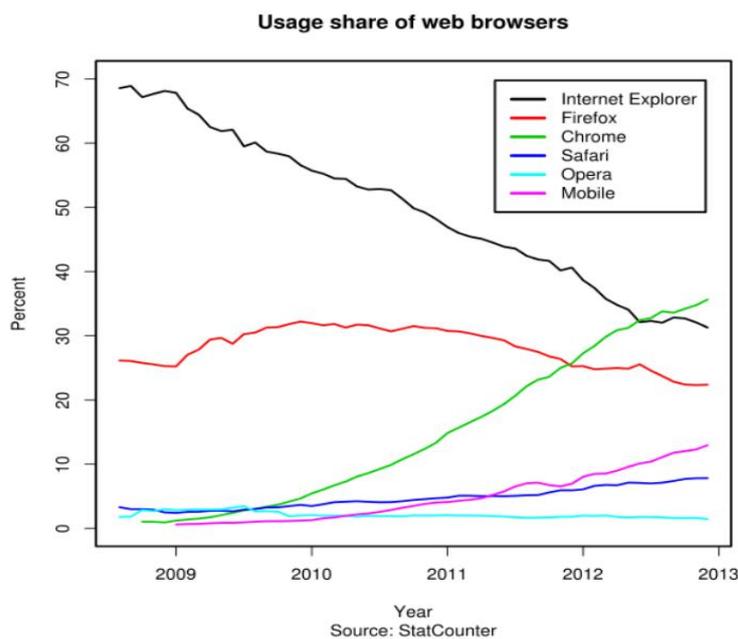


Figura 1-3 Porcentaje de Uso de los Navegadores Web [9]

1.3.5. URL, URN Y URI

Un Localizador de Recursos Uniforme (URL) es una cadena de caracteres mediante la que se asigna una dirección única a cada uno de los recursos presentes en Internet, haciendo uso del nombre del computador en el que se encuentra la información, el directorio dentro del computador, el nombre del archivo, el tipo de protocolo, entre otros, de esta manera un navegador puede acceder fácilmente a la información requerida por el usuario.

Un nombre de Recurso Uniforme (URN) funciona de manera similar a un URL, permiten diferenciar los recursos existentes en el Internet con la diferencia de que en un URL se puede especificar la ubicación del archivo.

Un identificador Uniforme de Recurso (URI) es un identificador más completo puede ser un URL, un URN o puede ser ambos, se acostumbra a llamar URL a todas las direcciones de la web, estas direcciones identifican el protocolo que se va a usar, en una sección denominada esquema, aunque comúnmente se usa HTTP, existe un gran número de protocolos lo que implica una variedad de esquemas.

La parte jerárquica de la dirección puede contener una dirección numérica o la información de dominio para acceder al equipo servidor y la ruta para acceder a la información dentro del equipo; para acceder a la información puede ser necesario pasar variables que están en el campo solicitud delimitado por el signo de interrogación este campo finaliza al encontrar el caracter numeral (#) que determina el inicio del campo fragmento, este campo marca la diferencia entre URL y URI, permite indicar una subdirección dentro de la información a la que se desea acceder.

Aunque hay una diferencia sutil entre URL y URI puede decirse que a su vez todo URL es una URI permitiendo ambos la identificación de recursos, por tanto es evidente que para el diseño de servicios Web, páginas dinámicas o páginas estáticas es importante contar con identificadores claros, pueden aparecer variables técnicas en la URL, pudiendo ser muchos los motivos; mostrar contenido en función de variables que ingresa el usuario o por simple omisión del desarrollador, lo importante es contar con una URL amigable que permita a los navegadores llenar formularios y extraer datos, además de una fácil navegación (Ver Figura 1-4).

Protocolo	Dominio	Puerto	Ruta	Consulta	Fragmento
http:// www.epn.edu.ec : 45719/2012/JUNIO.html? SEMANA=3#LUNES					

Figura 1-4 Partes de URI

1.3.6. HTTP

Inicialmente la información se intercambiaba en Internet utilizando HTTP, versiones posteriores del protocolo permitían el intercambio de mensajes con encabezados con la finalidad de ayudar al intercambio de archivos HTML entre un navegador (cliente web) y servidores HTTP localizados mediante una cadena de caracteres (URL).

HTTP basa su funcionamiento en operaciones Petición-Respuesta sencillas y en texto plano, lo cual le permite al usuario ver información legible y fácil de depurarla aunque esto implique que el mensaje sea más largo. El cliente establece una conexión con el servidor envía datos de la solicitud en un mensaje, el servidor

responde con el estado de la solicitud y su posible resultado, esto no implica que exista una sesión, de hecho una característica del protocolo es no almacenar estados de los intercambios de información, se establece una conexión se transfiere la información y se libera la conexión.

La solicitud debe distinguir el método que se aplicará a la información, el identificador de la información y la versión de protocolo para el intercambio de información. Los métodos permitidos varían dependiendo de la versión del protocolo (Ver Figura 1-5).

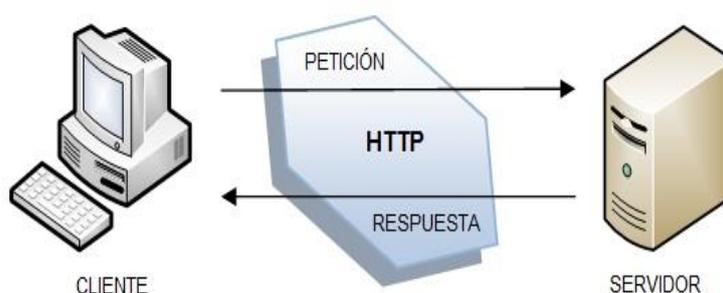


Figura 1-5 Comunicación entre el cliente (navegador) y el servidor

La respuesta consiste en un mensaje HTTP, indica la versión de protocolo que se utiliza para el intercambio de información, un código de estado representado por tres dígitos que indican si la solicitud fue atendida o no, además proporciona las razones por las que no se atendió la solicitud, existen cinco tipos de estados los mismos que se identifican y agrupan por el primer dígito;

- Los que inician por uno (1) son Informativos e indican que se recibió la solicitud y que está siguiendo el proceso.
- Los que inician por dos (2) indican Éxito, es decir que la acción señalada por el usuario en la solicitud ha sido identificada y aceptada.
- Los que inician por tres (3) indican Redirección, es decir que se deben realizar acciones adicionales para completar la solicitud del usuario.
- Los que inician por cuatro (4) Error del Cliente, permiten notificar al usuario que la solicitud es incorrecta y no se puede proceder.
- Los que inician por cinco (5) Error del Servidor, que indican problemas en el servidor al procesar la solicitud.

En cada petición se especifican los métodos que se deben realizar, pueden ser métodos seguros que no provoquen una acción o pueden ser métodos para realizar acciones inseguras.

Método *Options*, el cliente conoce de las operaciones que se pueden efectuar sobre un recurso pero estas son indicadas por el servidor.

Método *Get*, implica devolución de información al usuario, esta información es manejada por medio del respectivo URI.

Método *Head*, este método es similar al método *Get*, con la excepción de que no se devuelve contenido, tan solo las cabeceras, las cabeceras devueltas por el servidor son las mismas que si se tratara del método *Get*.

Método *Post*, este método se creó para que el servidor acepte el contenido enviado en las solicitudes, el método *Post* desempeña tareas determinadas por el servidor basado en la URI contenida en la solicitud, el resultado no necesariamente tiene un URI asociado.

Método *Put*, este método permite que el servidor almacene las solicitudes usando el URI contenido en las mismas, en caso de que el URI ya exista se interpreta como una actualización, puede parecer que el método *Post* y *Put* son similares, lo que los diferencia es lo que significa el URI en cada uno de ellos, mientras que para *Post* identifica el recurso que será manejado, en *Put* identifica el contenido.

Método *Delete*, el servidor elimina un recurso indicado mediante un URI en la solicitud, puede que se obtenga una respuesta favorable pero no garantiza al usuario que la operación se efectuará con éxito.

Método *Trace*, este método sirve para determinar la existencia o no del receptor, se determina la ruta, se limita el número de nodos intermediarios evitando los lazos.

1.3.7. TECNOLOGÍAS BASE DE LOS SERVICIOS WEB

El intercambio de mensajes sobre los protocolos de Internet como HTTP, es la base de los servicios Web (Ver Figura 1-6).

Este mecanismo de comunicación es entendido y empleado por cualquier lenguaje o plataforma, facilitando así la interoperabilidad.

Al momento de implementar una aplicación existen muchas tecnologías relacionadas a los servicios Web; sin embargo a continuación se menciona a las más conocidas:

- El diseñador del Servicio Web es libre de escoger el lenguaje o plataforma para la implementación.
- El Servicio Web es definido en WSDL (*Web Services Description Language*), un documento XML (*eXtensible Markup Language*) que describe de forma estándar al servicio y sus interfaces.
- El Servicio Web se registra mediante UDDI (*Universal Description, Discovery and Integration*). UDDI almacena la descripción de los servicios Web, es decir almacena los archivos WSDL.
- Las llamadas a operaciones de los servicios Web se realizan mediante SOAP, lo que permite a una aplicación de usuario mantener un vínculo con los servicios Web.



Figura 1-6 Protocolos de servicios Web

1.3.7.1. XML

XML no solo es el formato de mensaje sino también es la manera en la que se definen los servicios. De ahí la importancia de conocer algo sobre XML, sobre todo de la manera en que se usa para definir e implementar los servicios Web.

Las limitaciones presentadas por HTML, por ejemplo el manejo de contenido dinámico, determinaron el surgimiento de XML. En la Tabla 1-6 se comparan las características de XML y HTML.

XML	HTML
Permite trabajar con los datos en tiempo real.	Se puede realizar la comparación de información.
Es estructurado, está centrado en los datos en sí.	Se centra solamente en el despliegue de los datos.
Las etiquetas de XML son prácticamente ilimitadas.	Las etiquetas que se pueden usar son solo aquellas compatibles con HTML.
Creado para manejar datos dinámicos.	Los datos que maneja son estáticos.
XML es básicamente texto, fácil de depurar.	Posee un conjunto finito de etiquetas y sintaxis simple.
Es estructurado, varias estructuras de datos.	Facilita el diseño de elementos multimedia al permitir incorporar imágenes, video, etc.

Tabla 1-6 XML vs HTML [10]

Con XML el significado asociado a los datos se puede describir de manera clara definiendo el número de elementos necesarios para este fin, describiendo además lo que se puede realizar con ellos.

De XML se puede destacar que no existe la necesidad de mejoras para que funcione de manera adecuada con las nuevas versiones que presentan los navegadores, la gran variedad de aplicaciones existentes en Internet tampoco implica un problema debido a que XML es extensible y flexible.

XML permite generar documentos mejor estructurados, esto implica que al realizar una búsqueda se obtendrán resultados más precisos y satisfactorios, el propio usuario (cliente web) puede manipular los datos, se le permite además el intercambio de documentos entre aplicaciones locales (PC) y aplicaciones distribuidas en Internet.

XML permite desarrollar de mejor manera el hipertexto, por ejemplo, enlaces que se especifican independientes de su ubicación, permite especificar atributos para los enlaces, etc. Al manejar XML no se está limitando el tipo de documento ya que se puede exportar en otros formatos, es más se podría considerar a XML una especie de llave maestra, que se integra en el formato deseado.

1.3.7.2. SOAP

SOAP (*Simple Object Acces Protocol*) fue el primer protocolo que permitió la comunicación mediante XML sobre cualquier protocolo de Internet. Proceso que involucra mensajes simples y un protocolo ampliamente difundido, se considera neutral, eliminando inconvenientes respecto a sistemas operativos, lenguajes de programación o plataformas distintas ofreciendo al usuario libertad en el desarrollo de aplicaciones distribuidas.

Describe un modelo de comunicación unidireccional permitiendo el proceso típico de petición/respuesta estilo RPC. Cuando se realiza una petición, el usuario envía todos los datos necesarios para la petición, debiendo ser transmitidos en un formato específico basado en XML al servidor. Una vez que el mensaje llega al punto terminal en la comunicación de SOAP la codificación XML es convertida a datos que son manejados por el lenguaje de programación en el que está implementado el servidor.

Aun cuando existían diversas opciones de protocolos que podrían facilitar la comunicación entre aplicaciones, con el tiempo, gracias al respaldo de importantes industrias, SOAP pasó a convertirse en el núcleo de los servicios Web. Por muchas razones SOAP predominó sobre las otras opciones disponibles:

- SOAP es un protocolo que permite la comunicación entre aplicaciones, no se asocia a ningún lenguaje de programación en particular, la API¹⁷ se determina por la elección del lenguaje que realice el desarrollador.
- SOAP nació como un protocolo que permitía la comunicación de parámetros encapsulados en XML para una petición, siendo solamente líneas en un documento, podía transmitirse sobre cualquier protocolo de transporte que soporte texto (HTTP, SMTP¹⁸), es decir SOAP no estaba ligado a ningún protocolo de transporte específico.

La estructura del mensaje SOAP se basa en un sobre que contiene el mensaje y una cabecera que lo describe así:

- **Envelope (Envoltorio):** Define el contenido de un mensaje y como procesarlo, es decir identifica el inicio y el fin del mensaje.
- **Header (Cabecera):** Es información adicional de la cabecera del mensaje, que puede considerarse como opcional, ayuda al procesamiento del mensaje tanto en puntos intermedios como en su destino.
- **Body (Cuerpo):** Aquí se encuentran los principales datos del mensaje XML, ayuda a interpretar las llamadas y respuestas.

1.3.7.3. WSDL

Un usuario obtiene acceso a las aplicaciones en Internet a través de procesos en los que se envía parámetros que son conocidos por el usuario ya que las aplicaciones previamente muestran sus funcionalidades, por lo cual existe un intercambio de información entre el cliente y la aplicación lo que implica que debe existir un protocolo que facilite este proceso ya que SOAP no brinda el soporte adecuado para realizar las especificaciones que son necesarias.

¹⁷ **API** (Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

¹⁸ **SMTP** (Protocolo para la transferencia simple de correo electrónico), es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

El Lenguaje de Descripción de Servicios Web (WSDL) es un formato de esquema XML que fue desarrollado principalmente por Microsoft e IBM y fue sometido por estas compañías al W3C para su aceptación y publicación.

WSDL permite describir a una aplicación de manera que se conozca lo que hace, de qué manera cumple sus funciones y la forma en que los clientes que la requieran puedan acceder a ella.

1.3.7.4. UDDI

Una vez que el servicio ha sido descrito y se ha definido claramente al medio para el intercambio de mensajes, resta especificar una manera para publicar el servicio que ofrece una aplicación en particular y encontrar los servicios que otras aplicaciones ofrecen. Esta precisamente es la función que cumple UDDI (*Universal Description, Discovery and Integration*).

UDDI es un catálogo compuesto por documentos XML, en el que se ofertan los servicios Web, promovido por consorcios independientes. UDDI es un estándar para el registro de los servicios mediante WSDL. Similar a la agenda de contactos telefónicos de un celular, UDDI almacena las interfaces de los servicios Web descritas en WSDL.

Provee dos tipos de API, de publicación y de consulta. La API de publicación facilita el registro y publicación de los servicios por parte del mismo proveedor. Mientras que la API de consulta está orientada a los usuarios de los servicios, a los que facilita dos llamadas diferentes, una para efectuar la búsqueda y una para obtener los resultados referentes a un servicio (Ver Figura 1-7).

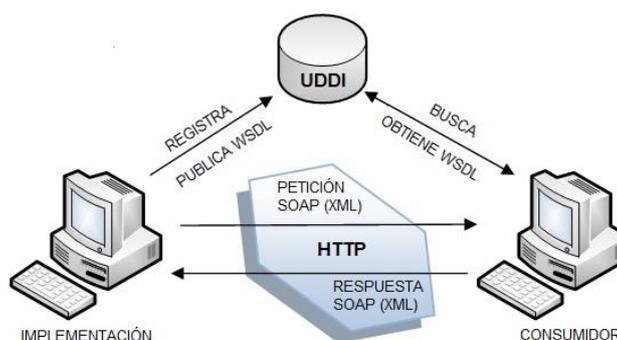


Figura 1-7 Estándares de Servicios Web

1.3.8. REST

La Transferencia de Estado Representacional (REST), es una arquitectura orientada principalmente para sistemas distribuidos como el *World Wide Web* (WWW). Es un conjunto de principios agrupados para formar una interfaz que permite definir claramente recursos de un dominio y transmitirlos mediante HTTP, surge como alternativa simple a SOAP y a los servicios Web basados en el Lenguaje de Descripción de servicios Web (WSDL).

Según [11]:

“REST es un intento de mostrar cómo debe comportarse una aplicación web bien diseñada: una red de páginas web (una máquina de estados virtual) donde el usuario progresará seleccionando enlaces (transiciones de estado) que devuelven la página siguiente (el siguiente estado de la máquina) que el usuario manejará a su gusto”

REST no es algo nuevo, busca acoplarse a las principales características que han permitido el desarrollo de Internet, se apoya en formatos y estándares existentes y utilizados en la mayoría de aplicaciones, principalmente se debe destacar el sistema de direccionamiento que permite una identificación única mediante las URL.

En los últimos años tal es el éxito de REST como tecnología para el desarrollo de servicios Web que ha logrado desplazar a estándares predominantes como SOAP.

REST se basa en las siguientes ideologías:

- Soporte de crecimiento y escalabilidad. Internet ha mostrado un crecimiento exponencial a pasos agigantados desde su surgimiento, es lógico que una tecnología que pretende funcionar sobre Internet comparta esta característica, REST simplifica los estándares y protocolos para mejorar el rendimiento del sistema, de esta manera la comunicación no estará supeditada al servidor ni mucho menos al cliente.
- Un mecanismo de acceso común o generalizado, interfaces conocidas que permitan un fácil acceso.

- El servidor no debe almacenar datos de los clientes que realizan las solicitudes, *stateless*.

HTTP permite que REST alcance sus objetivos:

- Los recursos en Internet a los que los usuarios envían los mensajes se identifican de forma única y global por medio de los URI. Los recursos se manejan por medio de representaciones.
- Se transfiere el estado completamente mediante el uso de un conjunto limitado de operaciones que define HTTP.
- REST plantea el uso claro y adecuado de los métodos de HTTP y las respectivas operaciones. En la Tabla 1-7 Métodos HTTP se mencionan las operaciones asociadas a los métodos HTTP.

Es recomendable que los servicios REST se implementen contemplando el uso de un conjunto de tipos de contenido representados por medio de un tipo MIME¹⁹ garantizando a los clientes que sin importar el lenguaje en el que estén implementados, puedan utilizar el Servicio Web.

MÉTODO HTTP	OPERACIÓN
POST	crear recurso nuevo en el servidor
GET	leer, obtener un recurso
PUT	actualizar un recurso
DELETE	borrar un recurso

Tabla 1-7 Métodos HTTP

- El estado no se mantiene, en cada mensaje HTTP se envía toda la información necesaria para completar la solicitud de un cliente. Como se mencionó REST brinda escalabilidad para ello hace uso de una topología

¹⁹ **MIME** (Multipurpose Internet Mail Extensions), son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.

compuesta por un sin número de elementos intermediarios, por lo que se hace evidente que para reducir al mínimo el tiempo de respuesta a una solicitud, el usuario debe enviar todos los datos necesarios para que el Servicio Web lleve a cabo con éxito el pedido, es decir que al no mantener estado se mejora el rendimiento de un Servicio Web.

- La navegación entre recursos REST se realiza siguiendo enlaces sin necesitar de registros, el estado de una aplicación web REST es almacenado en documentos HTML o XML, que pueden residir tanto en el cliente como en el servidor.

A pesar de ser una arquitectura y no un estándar, REST parece definir el uso de varios estándares en la implementación de servicios Web, tal es el caso de HTTP, URL, XML, HTML, MIME.

1.3.9. SEGURIDAD EN LOS SERVICIOS WEB

En Internet existen muchas amenazas debido a que se maneja información crítica como: números de seguro e identificaciones, dinero electrónico, claves bancarias. Los *hackers* no han escatimado esfuerzos por tener acceso a esta información, el propósito de cifrar los mensajes y de construir canales seguros de comunicaciones es precisamente conseguir que solo las personas autorizadas tengan acceso a determinados datos.

Por esta razón, para brindar seguridad a las aplicaciones y servicios que corren sobre HTTP se creó HTTPS. HTTPS permite hacer segura la conexión impidiendo que se sustraiga información confidencial, ya que establece comunicaciones cifradas en sitios web. HTTPS es un protocolo de capa aplicación muy utilizado, emplea los protocolos TLS/SSL (*Transport Layer Security/Secure Sockets Layer*). Como TLS y SSL son protocolos muy similares generalmente se suele hacer referencia a estos dos como si fueran un solo conjunto de protocolos.

1.3.9.1. TLS Y SSL

TLS/SSL son un conjunto de protocolos criptográficos basados en criptografía simétrica²⁰ y asimétrica²¹, certificados²² y firmas digitales para obtener comunicaciones seguras en aplicaciones cliente-servidor a través de Internet.

TLS/SSL funciona de forma transparente para el usuario. Cuando se intenta acceder a un sitio web, por ejemplo, Facebook usando HTTPS, se lleva a cabo el procedimiento de la Figura 1-8.

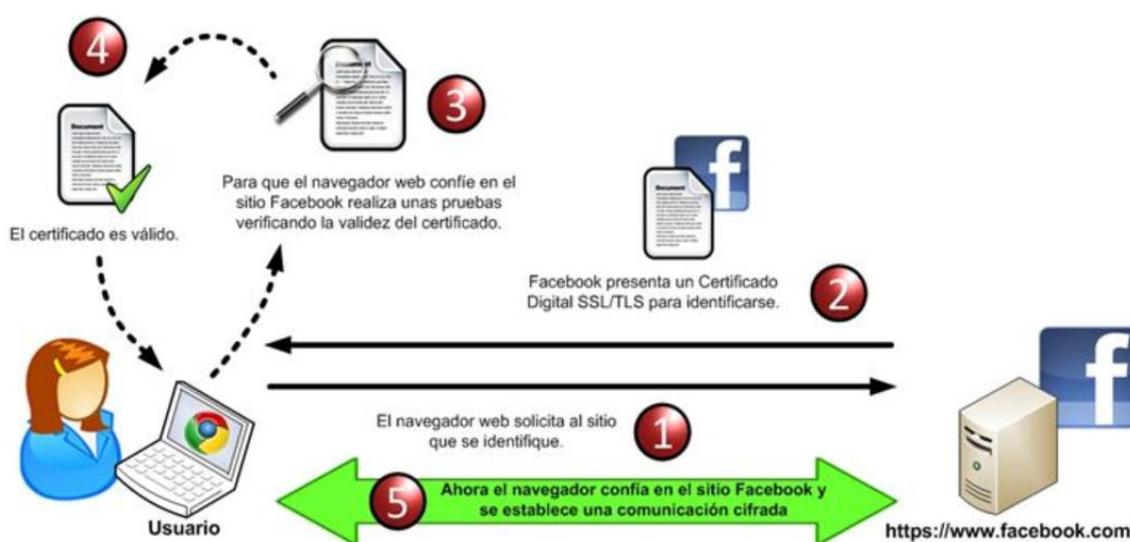


Figura 1-8 Funcionamiento general de SSL/TLS [12]

Cuando el navegador hace una petición al sitio seguro de Facebook, éste envía un mensaje indicando que quiere establecer una conexión segura y envía datos sobre la versión del protocolo TLS/SSL que soporta. En base a esta información enviada por el navegador, el servidor Web de Facebook responde con un mensaje informando que está de acuerdo en establecer la conexión segura con los datos de SSL/TLS proporcionados.

²⁰ **Criptografía Simétrica:** Es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes.

²¹ **Criptografía Asimétrica:** Es un método criptográfico que usa un par de claves para el envío de mensajes.

²² **Certificado digital:** Es un archivo electrónico que funciona como una contraseña en línea para comprobar la identidad de un usuario o equipo. También se utiliza para crear un canal cifrado de SSL que se utiliza para las comunicaciones entre un servidor y un equipo o dispositivo cliente. Un certificado digital es un comunicado emitido por una entidad de certificación (CA) que garantiza la identidad del poseedor del certificado y permite las comunicaciones cifradas.

Una vez que ambos conocen los parámetros de conexión, el sitio de Facebook presenta su certificado digital al navegador web para identificarse como un sitio confiable [12].

1.3.9.2. RAID

RAID (*Redundant Array of Independent Disks*) es una tecnología que combina varios discos duros para formar una única unidad lógica, donde los mismos datos son almacenados en todos los discos (redundancia). En otras palabras, es un conjunto de discos que funcionan como si fueran uno solo. Eso permite tener una tolerancia alta contra fallas, pues si un disco tiene problemas, los demás continúan funcionando, teniendo el usuario los datos a su disposición (transparencia) [13].

Para conformar el RAID, es preciso utilizar por lo menos 2 discos. El sistema operativo, en este caso, mezclará los discos como una única unidad lógica. Cuando se graban datos, los mismos se reparten entre los discos del RAID (dependiendo del nivel). Con eso, además de garantizar la disponibilidad de los datos en caso de fallo de un disco, es posible también equilibrar el acceso a la información, de forma que no haya "cuellos de botella" [13].

Aunque existan gran variedad de tipos o niveles de RAID disponibles en el mercado, elegir el nivel RAID adecuado dependerá del tipo de información que se maneje, además del nivel de relevancia y cantidad de la misma. A continuación se citan los 6 niveles relevantes de funcionamiento de RAID [14].

- **RAID 0**, aquí los datos se reparten en pequeños segmentos y se distribuyen entre los discos. Este nivel no ofrece tolerancia a fallos, pues no existe redundancia. Eso significa que un fallo en cualquiera de los discos puede ocasionar la pérdida total de la información. Se recomienda el uso de RAID 0 cuando se requiere de mayor rendimiento del almacenamiento y cuando no sea un gran problema la pérdida de la información.
- **RAID 1** funciona duplicando todos los datos de cada unidad de forma sincronizada a una unidad de duplicación exacta. Así, si por ejemplo, una computadora posee 2 discos, se puede anexar un disco para cada uno,

totalizando 4. Los discos que fueron añadidos, trabajan como una copia del primero, si el disco principal recibe datos, el disco anexado también los recibe. De esa forma, si uno de los discos presenta una falla, el otro disco continúa en funcionamiento. Aunque la grabación de datos es más lenta, pues es realizada dos veces, la lectura de esa información es más rápida, pues puede ser accedida de dos fuentes.

- **RAID 2** adapta el mecanismo de detección de fallas en discos duros. Actualmente, el RAID 2 es poco usado, ya que prácticamente todos los discos rígidos nuevos salen de fábrica con mecanismos de detección de fallas implantados.
- **RAID 3** aquí los datos son divididos entre los discos del *array*, excepto uno, que almacena información de paridad. Así, todos los bytes de los datos tienen su paridad (aumento de 1 bit, que permite identificar errores) almacenada en un disco específico. A través de la verificación de esta información, es posible asegurar la integridad de los datos, en casos de recuperación.
- **RAID 4** divide los datos entre los discos, siendo uno de esos discos exclusivo para paridad. La diferencia entre el nivel 4 y el nivel 3, es que en caso de falla de uno de los discos, los datos pueden ser reconstruidos en tiempo real a través de la utilización de la paridad calculada a partir de los otros discos, siendo que cada uno puede ser accedido de forma independiente. El RAID 4 es el indicado para el almacenamiento de archivos grandes, donde es necesario asegurar la integridad de la información.
- **Raid 5** es semejante a RAID 4 excepto por el hecho de que la paridad no está destinada a un único disco, sino a todo el arreglo de discos. Eso hace que la grabación de datos sea más rápida, pues no es necesario acceder a un disco de paridad en cada grabación. RAID 5 es el nivel más utilizado y ofrece resultados satisfactorios en aplicaciones no muy pesadas. Este nivel necesita de por lo menos 3 discos para funcionar.

1.3.10. SOA (SERVICE ORIENTED ARCHITECTURE)

SOA es de utilidad tanto para el proceso de desarrollo de aplicaciones como para el respectivo proceso de implementación, permitiendo la integración de tecnologías y aplicaciones independientes.

Según [15]:

“Aunque las iniciativas SOA normalmente se abordan desde el punto de vista tecnológico, SOA no es una tecnología, sino un enfoque o manera de hacer las cosas que aporta grandes beneficios al negocio. De forma simplificada, SOA consiste en crear elementos software discretos, modulares y reutilizables llamados servicios.”

Antes era muy complicado integrar sistemas aún más cuando se trataba de sistemas fuertemente acoplados, es decir sistemas que no permitían la escalabilidad. SOA es una arquitectura desacoplada que permite la reutilización de recursos y componentes de software, que generalmente se emplea en los servicios Web. Cabe mencionar que SOA y servicios Web son dos conceptos diferentes.

Para tener éxito en el desarrollo de una aplicación SOA el proceso de desarrollo debe ser orientado a obtener servicios comunes.

A continuación se describen las características más importantes de la arquitectura SOA:

- Contrato de Servicio, se refiere a la forma en que el servicio puede ser usado por otro servicio o programa que lo requiera, es decir que defina una interfaz.
- Debe garantizar su reusabilidad es decir que debe ser compatible con otros servicios.
- La funcionalidad que brinda debe ser propia, es decir que no deba acoplarse a otro servicio para funcionar, debe tener una mínima dependencia.
- Sin estado, que un servicio pueda ser invocado sin ninguna condición.
- Independiente, es decir que este separada la lógica de la aplicación de los datos. La idea es que el servicio no guarde datos, y solo contenga la lógica del negocio.

- Distribuible, es decir que los servicios estén ubicados en lugares locales o remotos y que para el usuario final sea como un solo sistema compacto.
- Débilmente acoplado, es decir que se pueda integrar con facilidad a nuevas aplicaciones.
- Deben poder ser descubiertos, con el fin de evitar redundancia en los servicios.

Según [16], SOA permite la integración y organización de sistemas de información independientemente de la tecnología que se use, brindando muchos beneficios como:

- Mejorar la capacidad de respuesta; rápida adaptación y despliegue de servicios, clave para responder a las demandas de clientes, *partners* y empleados.
- Adaptabilidad; facilita la adopción de cambios añadiendo flexibilidad y reduciendo esfuerzos
- Reduce la complejidad gracias a la compatibilidad basada en estándares frente a la integración punto a punto.
- Reutiliza los servicios compartidos que han sido desplegados previamente.
- Integra aplicaciones heredadas limitando así el coste de mantenimiento e integración.
- Beneficios en el desarrollo, ya que las aplicaciones son reutilizables, más fácil de mantener y tienen la capacidad de ampliación de las funcionalidades del sistema, exponiéndolas de una forma segura.

1.4. METODOLOGÍA DE DESARROLLO

Las metodologías permiten abstraer la secuencia lógica en la que se debe llevar a cabo un proceso y plasmar estas ideas en la implementación de una aplicación. El objetivo que toda metodología persigue es de obtener un producto funcional en el tiempo establecido y con el presupuesto estimado.

Se crearon metodologías de desarrollo de software para dar solución a problemas de tiempo, presupuesto y funcionamiento en sistemas de software. Antes de los años 90 se empleaban metodologías tradicionales seguían modelos basados en

procesos definidos y caracterizados por exhaustivas planificaciones previas al desarrollo, sin embargo surgieron las metodologías ágiles como alternativa a las metodologías tradicionales.

En la Tabla 1-8 se comparan las metodologías ágiles con las tradicionales.

Metodologías ágiles	Metodologías tradicionales
Existe un solapamiento de fases de desarrollo.	Desarrollo secuencial.
El cliente es parte del equipo de desarrollo.	El cliente sólo interactúa con el grupo de trabajo en reuniones.
Los procesos se ejecutan de manera concurrente.	Los procesos se ejecutan de manera sistemática.
Es pensada para permitir cambios durante el desarrollo del proyecto.	Se establecen parámetros de desarrollo que no se pueden cambiar.
No se parte de los requerimientos sino de la visión de un producto final.	Parte del análisis de los requisitos.
Se cuenta con un equipo multidisciplinario.	Equipo especializado.
Se basan en experiencias adquiridas en la práctica de desarrollo de aplicaciones.	Se basan en estándares ya definidos y en un plan pre-elaborado.
Se trabaja con Historias de Usuario o <i>Product Backlog</i> .	Se trabaja con Casos de Uso.
Da prioridad a las personas y las interacciones.	Da prioridad a los procesos y herramientas.

Tabla 1-8 Comparación de Metodologías [17]

1.4.1. METODOLOGÍAS ÁGILES

A pesar de la heterogeneidad de las metodologías, el desarrollo de sistemas web y en general el de cualquier tipo de aplicación, se caracteriza por la premura con la que son requeridas, dentro de este esquema prevalecen las metodologías ágiles, que brindan la reducción en el tiempo de desarrollo de una aplicación sin que exista una pérdida de calidad en el producto final, además de permitir el uso de procesos

que facilitan el *feedback* (realimentación) en determinado aspecto o elemento del software.

Las metodologías ágiles se basan en cuatro conceptos desarrollados en el “Manifiesto Ágil” [18]:

- Valorar más a los individuos y su interacción, por encima de los procesos y las herramientas.
- Valorar más el software que funciona, por encima de la documentación exhaustiva.
- Valorar más la colaboración con el cliente, por encima de la negociación contractual.
- Valorar más la respuesta al cambio, por encima del seguimiento de un plan

Existen muchas metodologías que siguen estos conceptos como: Scrum, Kanban, AM (*Agile Modeling*), XP (*eXtreme Programming*), entre otras.

1.4.2. SCRUM

Scrum es una metodología ágil que se centra en la gestión del proyecto de manera iterativa, flexible e incremental, enfocándose en *qué* cosas se debe hacer y no en el *cómo* se las debe hacer.

Scrum no se fundamenta en el seguimiento de un plan preestablecido, si no en la adaptación continua a las situaciones que se puedan presentar en el desarrollo del proyecto. Se basa en pequeños períodos de trabajo llamados *sprints*, los cuales se realizan cada 15 o 60 días, además de reuniones diarias para determinar los avances del proyecto. A continuación se mencionan las fases que se deben seguir, los roles que se manejan, como se llevan a cabo las reuniones y los elementos que conforman esta metodología ágil de desarrollo de software.

1.4.2.1. Fases de Scrum

Scrum está conformado por cinco fases que se indican en la Figura 1-9, las cuales deberán permitir la adaptabilidad y flexibilidad en el desarrollo del proyecto.

- **Concepto:** Definir una visión global del producto final, es decir conocer lo que se va a desarrollar.
- **Especulación:** Formular hipótesis, definir los requerimientos establecer la prioridad de las tareas, una vez identificadas las de mayor importancia asignarlas a la primera Iteración.
- **Exploración:** Establecer el número total de iteraciones para entrega del sistema listo, establecer los riesgos de que las iteraciones se acumulen.
- **Revisión:** realizar actividades de documentación, preparación en el manejo del sistema y revisión de la funcionalidad del sistema.
- **Cierre:** Entrega del producto terminado en la fecha establecida.

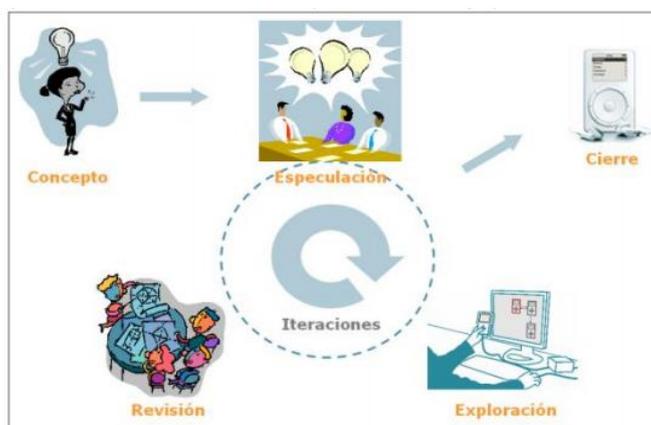


Figura 1-9 Fases de Scrum [19]

1.4.2.2. Roles de Scrum

Para que se puedan realizar las tareas con solvencia todos los miembros del equipo deberán conocer la metodología, además de tener espíritu de colaboración y un propósito en común, el de generar un producto de calidad.

En la Figura 1-10 se muestran los tres roles principales de Scrum.

- **Product Owner:** es el cliente. A través del *Product Owner* se realizará el *product backlog* (lista de requerimientos).
- **Scrum Master:** es el encargado de guiar al equipo para que se cumpla con la metodología, se encarga de la gestión del proyecto en general.

- **Team o Equipo:** está conformado por el equipo de desarrolladores encargados de la creación del producto. El equipo deberá asegurar el uso de técnicas y tecnologías ágiles para el desarrollo del sistema.

Cabe mencionar que el número de miembros del equipo (*Product Owner*, *Team* y *Scrum Manager*) puede estar conformado mínimo por cuatro personas y máximo por ocho, con el fin de mantener la agilidad en los *sprints*.



Figura 1-10 Roles de Scrum [19]

1.4.2.3. Elementos de Scrum

Scrum se compone de tres elementos, de los cuales los dos primeros forman los requisitos del sistema y el tercero forma el resultado del sistema.

- **Product Backlog (Pila del producto):** La realiza el *Product Owner* o cliente con el fin de poner en manifiesto sus requerimientos. Deberá presentar las funcionalidades del sistema las cuales deberán ser muy específicas, concretas y deberán estar priorizadas. Se puede emplear un formato de lista donde se incluya un identificador, la descripción de la funcionalidad requerida, la prioridad y el tiempo estimado, entre otros. En Scrum generalmente el *Product Backlog* se denomina Historias de Usuario.
- **Sprint Backlog (Pila del sprint):** Lo realiza el equipo el cual descompone a las funcionalidades generales del proyecto en unidades para asignar tareas determinadas a cada miembro del equipo. Se puede usar una pizarra, una

hoja de cálculo o una herramienta colaborativa o de gestión de proyectos, en donde se incluya la lista de tareas, la persona asignada a cada tarea, el estado, la prioridad y el tiempo, cabe señalar que solo se debe incluir lo necesario. Durante el *sprint* se actualiza el *sprint backlog* y a través de la reunión diaria los tiempos de cada miembro del equipo.

- **Incremento:** Es el resultado del *sprint*.

En la Figura 1-11 se muestran los elementos de Scrum.

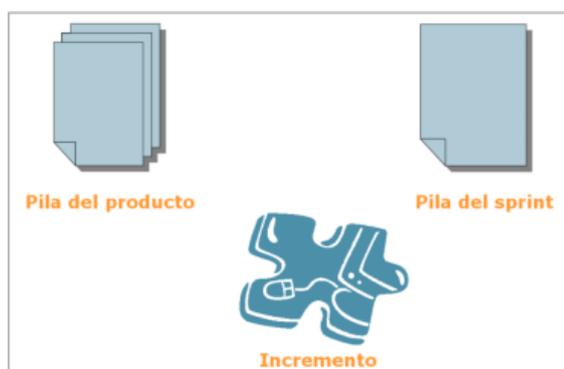


Figura 1-11 Elementos de Scrum [19]

1.4.2.4. Reuniones de Scrum

La gestión y el progreso de un proyecto basado en Scrum se determinan en tres tipos de reuniones, como se muestra en la Figura 1-12.

- **Planificación del *Sprint*:** Es una reunión cada 15 o 60 días, en la cual se decide que elementos del *product backlog* se van a desarrollar, quien las va a desarrollar y en cuánto tiempo. Dara como resultado el *sprint backlog*, las fechas, duraciones y objetivo del *sprint*. En cada *sprint* deberán intervenir todos los miembros del equipo, incluido el *Product Owner* el cual deberá aclarar dudas y comprobar que el equipo comprende las necesidades de negocio del cliente. El *Scrum Manager* actuará como moderador del *sprint*. Cabe mencionar que el *sprint* es un trabajo de colaboración activa entre todos los miembros del equipo.

- **Seguimiento del *Sprint*:** Es una pequeña reunión diaria que no deberá tomar más de 15 minutos, en la cual cada miembro del equipo de desarrollo menciona su avance.
- **Revisión del *Sprint*:** Es la reunión final en la cual el equipo de desarrollo presenta el producto final al *Product Owner* totalmente funcional, probado y operando en el entorno del cliente. Proporcionará de ser el caso documentación técnica o de usuario.

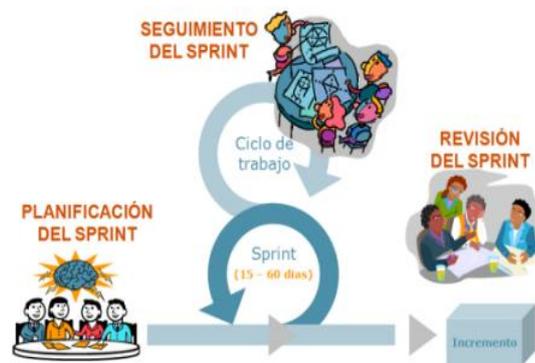


Figura 1-12 Reuniones de Scrum [19]

CAPÍTULO II

2. ANÁLISIS DE REQUERIMIENTOS

2.1. INTRODUCCIÓN

En este capítulo se realiza un estudio de los procesos internos ISO 9001:2008, relacionados al Otorgamiento y Administración de Títulos Habilitantes para los Servicio Móvil Terrestre para sistemas VHF/UHF Radio de Dos Vías (MT-RDV), y a los Registros de Sistemas de Modulación Digital de Banda Ancha (MDBA), con el objeto de conocer los procesos que se realizan en la actualidad en la Secretaría Nacional de Telecomunicaciones (SENATEL). Se hace una descripción de las normas para la implementación y operación de Sistemas MDBA Resolución No. 417-5-CONATEL-2005 y del Instructivo Formularios de Concesión de Frecuencias para analizar la información a ser validada. Además se analiza el formato adecuado de archivo XML que soporta el software SPECTRAplus. Finalmente se justifican las herramientas utilizadas para la creación y desarrollo del software.

2.2. SECRETARÍA NACIONAL DE TELECOMUNICACIONES (SENATEL)

A partir del año 1995, con la reforma de la Ley de Telecomunicaciones se separaron e independizaron las funciones de administración y regulación de las telecomunicaciones a nivel nacional, de esta manera el Consejo Nacional de Telecomunicaciones (CONATEL) se erigió como organismo de administración y regulación de las telecomunicaciones en el país y representante del Ecuador ante la UIT, el cual está apoyado por la Secretaría Nacional de Telecomunicaciones (SENATEL) como ente encargado de la ejecución de las políticas del CONATEL, así como organismo administrador y regulador del espectro radioeléctrico, los servicios de telecomunicaciones, radio y televisión. La Superintendencia de Telecomunicaciones (SUPERTEL), realiza las funciones de control y monitoreo del espectro radio eléctrico, de los operadores que exploten servicios de telecomunicaciones, entre otros. Y finalmente el Ministerio de Telecomunicaciones

y de la Sociedad de la Información (MINTEL) que es el organismo rector de las telecomunicaciones. En la Figura 2-1 Organismos Reguladores de las Telecomunicaciones en el Ecuador se muestran los organismos reguladores de las telecomunicaciones en el Ecuador.

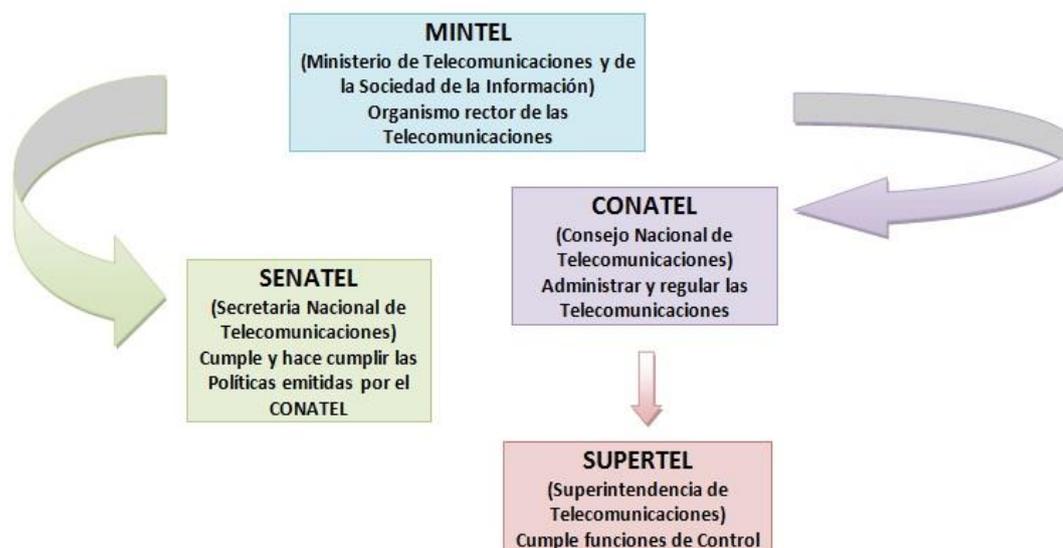


Figura 2-1 Organismos Reguladores de las Telecomunicaciones en el Ecuador

2.2.1. MISIÓN

“Fomentar el desarrollo de las Telecomunicaciones en un marco regulatorio convergente, y administrar eficientemente los recursos estratégicos relacionados, a fin de garantizar a la sociedad el acceso a servicios con calidad, diversificados y a precios justos, en todo el territorio nacional.” [20]

2.2.2. VISIÓN

“Seremos el organismo técnico en el ámbito de regulación y administración de recursos estratégicos relacionados a las Telecomunicaciones, con reconocimiento nacional e internacional, por el apoyo al desarrollo competitivo y el acceso igualitario de los servicios.” [20]

2.2.3. ISO 9001:2008

La SENATEL ha establecido sus procedimientos en función de la ISO 9001:2008 que es una norma creada por la ISO (Organización Internacional para la Estandarización), que especifica los requisitos para manejar un Sistema de Gestión de Calidad interno en organizaciones públicas o privadas, grandes medianas o pequeñas.

Esta norma define las responsabilidades de cada área, los objetivos y políticas organizacionales, la gestión de los recursos, y establece de forma sistemática los procesos para la elaboración del producto o servicio final, además de permitir un proceso de medición, análisis de resultados y mejora continua.

Todos estos procesos se deberán llevar a cabo con el único fin de proporcionar satisfacción al cliente al obtener un producto o servicio de calidad.

2.2.4. ESTRUCTURA ORGÁNICA DE LA SECRETARÍA NACIONAL DE TELECOMUNICACIONES

A la SENATEL le corresponde formular, orientar y ejecutar las políticas y estrategias del sector de las telecomunicaciones, estando integrada por tres niveles estructurales:

- **Nivel Asesor.**- Constituye la instancia de consulta y asesoramiento para la toma de decisiones del Nivel Ejecutivo.
- **Nivel de Apoyo.**- Brinda el soporte administrativo y de coordinación adecuado para cumplir la misión y los objetivos de la Secretaría.
- **Nivel Operativo.**- Es el encargado de la ejecución de los planes estratégico y operativo de la Secretaría Nacional de Telecomunicaciones.

En la Figura 2-2 se muestra la estructura orgánica de la SENATEL.

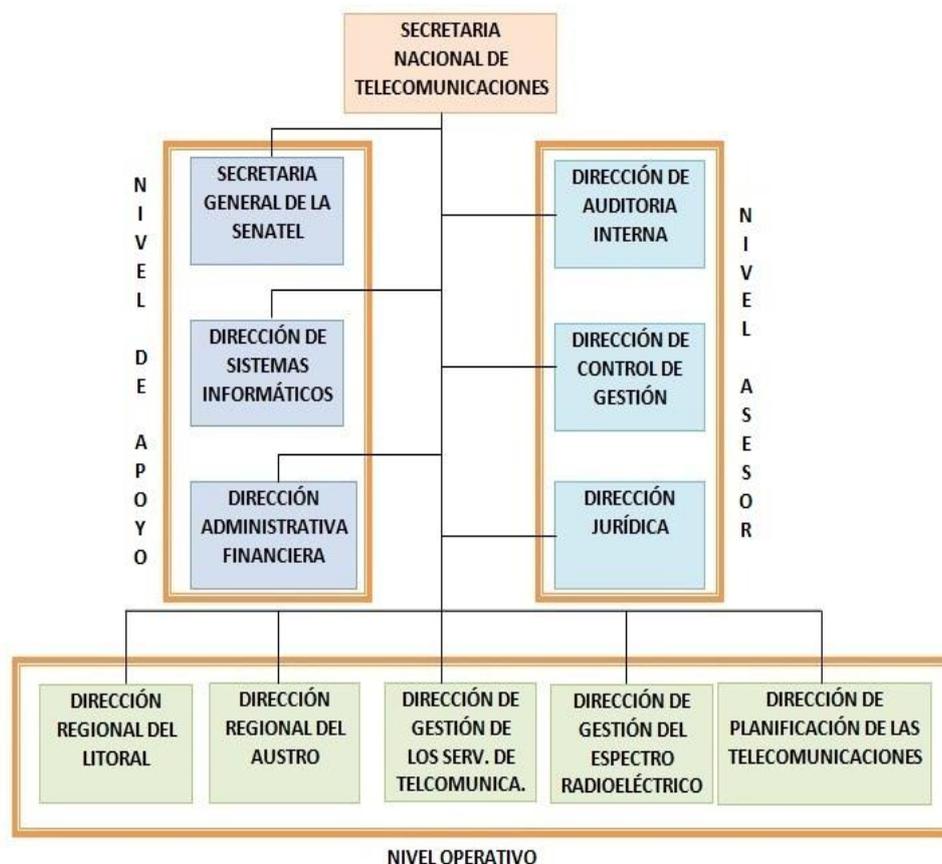


Figura 2-2 Estructura Orgánica de la Secretaría Nacional de Telecomunicaciones

Existen 11 unidades distribuidas en diferentes niveles y son:

- **CAU:** Centro de Atención al Usuario
- **DGGST:** Dirección General de Gestión de los Servicios de Telecomunicaciones
- **DGGER:** Dirección General de Gestión del Espectro Radioeléctrico
- **DGSI:** Dirección General de Sistemas Informáticos
- **DGAF:** Dirección General Administrativa Financiera
- **DGAI:** Dirección General de Auditoría Interna
- **DGJ:** Dirección General Jurídica
- **DGP:** Dirección General de Planificación de las Telecomunicaciones
- **SG:** Secretaría General de la SENATEL
- **DRA:** Dirección Regional del Austro
- **DRL:** Dirección Regional del Litoral

2.2.5. FUNCIONES DE LA SENATEL

La SENATEL tiene bajo su responsabilidad el otorgamiento de Títulos Habilitantes, de Registros y Licencias, etc. [20].

2.2.5.1. Otorgamiento de Títulos Habilitantes

Consiste en:

- Concesión de Servicios de Telecomunicaciones
- Permiso de Servicios de Valor Agregado
- Permiso de Redes Privadas
- Permiso y Registro de Operación de Entidades de Acreditación
- Concesión de Frecuencias
- Renovación de Frecuencias
- Permisos para la Provisión del Segmento Espacial

2.2.5.2. Otorgamiento de Registros y Licencias

Consiste en:

- Registro de Ciber Cafés
- Registro de Redes Físicas
- Registro de Convenios de Conexión
- Registro de Convenios de Reventa
- Registro de Acuerdos de Interconexión
- Emisión y Registro de Disposiciones de Interconexión
- Autorización Temporal de Frecuencias
- Registro de Enlaces de Modulación Digital de Banda Ancha (Explotación)
- Registro de Enlaces de Modulación Digital de Banda Ancha (Privado)
- Licencia para Banda Ciudadana y Radioaficionados
- Renovación de Licencia para Banda Ciudadana y Radioaficionados

Para el cumplimiento de estas funciones todas las unidades administrativas son responsables de un procedimiento general, ya sea para el Otorgamiento de Títulos Habilitantes o para el Otorgamiento de Registros y Licencias, que consta entre otros de la recepción, digitalización de documentos en el CAU, el ingreso del

trámite a los respectivos servidores, y la entrega de los trámites a las diferentes direcciones tanto técnica como jurídica, la aprobación y la elaboración del contrato.

A continuación se presenta en resumen los procedimientos internos para el Otorgamiento de Títulos Habilitantes para el Servicio Móvil Terrestre-Radio de Dos Vías (FM-RDV) y para el Registro de Licencias de Sistemas MDBA, ya que el desarrollo del presente Proyecto se basa en estos dos sistemas.

2.2.5.3. Procedimiento para el Otorgamiento de Títulos Habilitantes (OTH)

El propósito del OTH es de concesionar y renovar frecuencias del espectro radioeléctrico a personas naturales o jurídicas que la requieran en el territorio nacional. Este procedimiento es aplicable a la concesión de frecuencias para el Servicio FM-RDV. En la Figura 2-3 se muestra el procedimiento para el OTH que en los siguientes ítems se detallan.

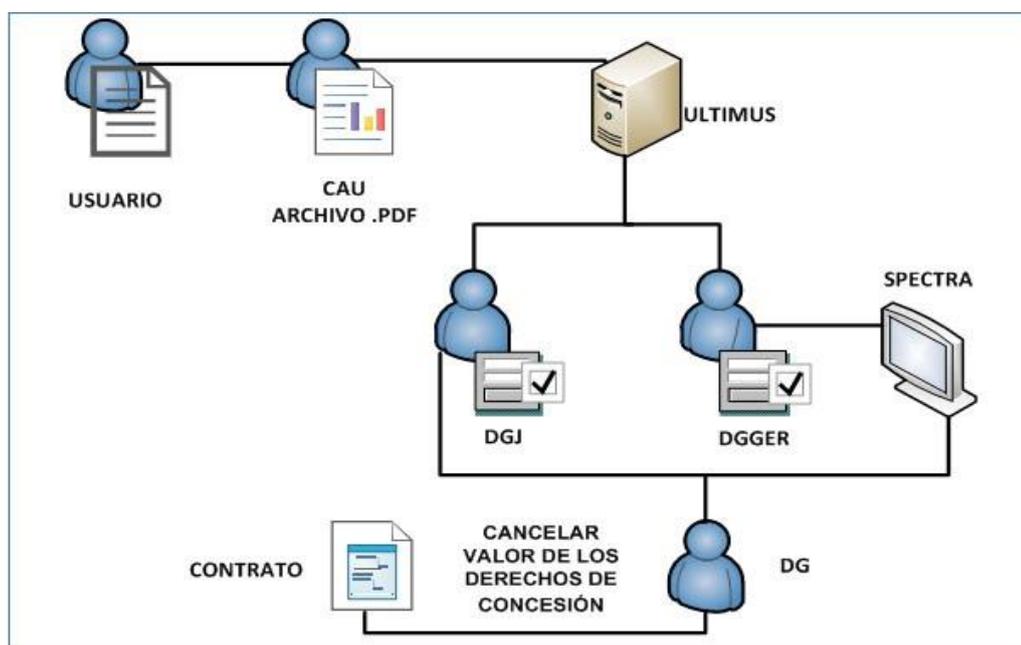


Figura 2-3 Proceso para el Otorgamiento de Títulos Habilitantes

2.2.5.3.1. *Proceso de recepción y digitalización de solicitudes*

Este proceso se lo lleva a cabo tanto para el Otorgamiento de Títulos Habilitantes, como para el Otorgamiento de Registros y Licencias.

- El concesionario se dirige a la SENATEL donde un funcionario del CAU revisa los anexos presentados según el cuadro de requisitos para los diferentes servicios.
- El funcionario del CAU organiza los documentos anexos a las solicitudes conforme a los requisitos de cada trámite.
- En el caso de que la documentación se encuentre incompleta e ilegible, se explica al usuario del incumplimiento de requisitos.
- Si el concesionario ha cumplido con los requerimientos para el ingreso de su solicitud, el funcionario procede a:
 - ✓ Asignar automáticamente el número de trámite.
 - ✓ Imprimir el comprobante de ingreso y se lo entrega al concesionario.
 - ✓ Recibir las solicitudes con sus anexos y enumerarlas hoja por hoja.
- El funcionario del CAU digitaliza los documentos físicos y realiza un comparativo del total de hojas escaneadas con la cantidad de hojas de los documentos físicos y procede a subir el trámite a las aplicaciones informáticas de la Institución pudiendo ser el sistema ULTIMUS²³ o el sistema DTS²⁴.
- El funcionario envía diariamente la documentación física al área de Documentación y Archivo.

2.2.5.3.2. *Elaboración del Informe Jurídico*

El CAU remite digitalmente los trámites a través del ULTIMUS al analista de la DGJ, quién verifica los antecedentes del trámite, y si no está adeudando, se puede continuar con el trámite.

²³ **ULTIMUS:** Sistema computacional empleado por la SENATEL para la administración de trámites.

²⁴ **DTS:** Sistema computacional empleado por la SENATEL para la administración de trámites.

Si están incompletos se elabora el anexo jurídico y se remite digitalmente a través del sistema ULTIMUS a la SG para la consolidación y notificación al concesionario. En caso de no existir respuesta a lo solicitado dentro del plazo concedido se solicitará se proceda con el archivo del trámite.

Con los requisitos completos el analista de la DGJ procede a la elaboración del Informe Jurídico, el cual una vez firmado por el director regresa al analista de la DGJ para separar la documentación y remitir los informes física y digitalmente a la SG.

2.2.5.3.3. Elaboración del Informe Técnico

El subdirector de la DGGER asigna el trámite al ingeniero analista usando ULTIMUS, quién realiza un análisis basándose en los Instructivos Formularios de Concesión de Frecuencias [21] y en la RESOLUCIÓN NO. 417-15-CONATEL-2005 [6], para cada servicio de radiocomunicaciones, en los que revisa parámetros tales como; tipo de servicio, banda de frecuencias, ancho de banda, potencia, ubicación de las estaciones repetidoras y fijas, equipos, antenas, cálculos de propagación y de RNI²⁵, perfiles topográficos, área de cobertura, entre otros.

Si el trámite no tiene inconvenientes, se ingresa la información de forma manual en el software SPECTRAplus para obtener un informe final, el cual es remitido físicamente y magnéticamente, para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director de la DGGER.

En caso de que en el estudio de ingeniería se verifique que el trámite esté mal realizado o incompleto, el ingeniero analista elabora digitalmente un anexo técnico en el sistema ULTIMUS, detallando las correcciones que se deben implementar así como los documentos faltantes, este anexo es remitido por medio magnético para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director

²⁵ **Las Radiaciones No Ionizantes (RNI)** son las radiaciones electromagnéticas que no tienen la energía suficiente para ionizar la materia y por lo tanto no pueden afectar el estado natural de los tejidos vivos.

de la DGGER, luego de lo cual se lo remite en forma magnética a través del sistema ULTIMUS a la SG, para que elabore el respectivo oficio dirigido al solicitante de frecuencias, adjuntando el respectivo anexo técnico de falta de requerimientos.

2.2.5.3.4. Revisión y Consolidación de Informes

Si no hubo inconvenientes en los dos informes tanto técnico como legal, se envían los informes a la SG, donde el especialista jefe de la SG utilizando los sistemas electrónicos institucionales, normativa legal vigente, cuadro de requisitos y documentos digitales del peticionario que proporciona el sistema, realiza la verificación de los datos consignados en los informes, pre aprueba e imprime el oficio. Una vez firmado el oficio, se entrega al jefe del área de Documentación y Archivo, quién una vez numerado y fechado el oficio, envía al CONATEL.

En el caso de existir inconsistencias en los informes, se procede al llenado en el registro de Producto No Conforme.

2.2.5.3.5. Notificación al solicitante sobre la Concesión de Frecuencias aprobada

Una vez que el CONATEL emite la Resolución autorizando al usuario la concesión de frecuencias, se asigna a un analista de la DGJ, el cual procede a generar el oficio para el concesionario al que debe adjuntarse la Resolución aprobada. Finalmente se envía el oficio de notificación al Archivo Central para el despacho del documento al usuario a través de correo electrónico.

2.2.5.3.6. Elaboración del Contrato

El concesionario cancela en recaudación el valor de los derechos de concesión indicado en el oficio de notificación, luego se traslada a la DGJ, con la copia de la factura, tres copias de la cedula de ciudadanía y papeleta de votación a color para la elaboración del contrato.

El analista de la DGJ procede a estructurar y sellar el contrato de concesión con el respectivo dato técnico, pastas correspondientes al tipo de sistema, Resolución y documentos habilitantes.

2.2.5.4. Procedimiento para la Administración de Títulos Habilitantes (ATH)

El propósito del ATH es realizar todas las modificaciones técnicas que se requieran en los títulos habilitantes para la concesión de frecuencias. Los trámites son recibidos por el CAU, donde un funcionario del CAU revisa los anexos presentados según el cuadro de requisitos para los diferentes servicios y lo envía a través del DTS a la DGGER en donde se realiza un informe técnico de estos trámites que luego serán remitidos a la DGJ.

El subdirector de la DGGER asigna el trámite al ingeniero analista usando DTS, para atender las modificaciones solicitadas o requeridas. El responsable técnico asignado analiza la información técnica, y de no existir errores ingresa manualmente las modificaciones en el sistema SPECTRAplus correspondiente y elabora el oficio de modificaciones técnicas, el cual es remitido físicamente y magnéticamente, para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director de la DGGER.

En caso de no ser pertinente se elabora el oficio de notificación al solicitante indicando la causa de la negación, el cual es remitido físicamente y magnéticamente, para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director de la DGGER.

Una vez que el oficio de modificación es aprobado, se envía a la DGJ. El director de la DGJ sumilla el oficio de modificación al responsable del Registro Público de Telecomunicaciones.

El responsable del Registro verifica si se trata de una modificación técnica o una cancelación e ingresa los datos en el sistema y archiva el oficio de modificación.

2.2.5.5. Registro de Enlaces de Modulación Digital de Banda Ancha (MDBA)

El propósito es conocer las bandas de frecuencias atribuidas en el Plan Nacional de Frecuencias a los sistemas que emplean técnicas de Modulación Digital de Banda Ancha (explotación), dentro del territorio nacional.

El Proceso de Recepción y Digitalización de Solicitudes, es mismo que para OTH. Se asigna el trámite a través del ULTIMUS al ingeniero analista de la DGGER, quien revisa la información técnica ingresada y verifica si corresponde a un Sistema de MDBA, revisa parámetros tales como; tipo de servicio, configuración del sistema, banda de frecuencias, modo de operación, ubicación de las estaciones fijas, potencia de las estaciones, homologación y características de equipos y antenas, cálculos de Radiaciones No Ionizantes, entre otros.

Si la información presentada en el estudio de ingeniería es correcta, entonces se realizan los siguientes pasos:

1. Se ingresan los datos en SPECTRAplus.
2. Se genera el Certificado de Registro de Enlaces de MDBA (Explotación) y se elabora el oficio de notificación de pago por los enlaces registrados.
3. El certificado y el oficio se remiten en medio físico y magnético para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director.
4. Se despacha el oficio de notificación de pago y el Certificado de Registro de Enlaces de MDBA (Explotación) se almacena en el archivo de la DGGER.
5. Una vez que el usuario cancela el valor correspondiente en Recaudaciones de la SENATEL, automáticamente se genera una alarma en el sistema ULTIMUS, a fin de que el ingeniero analista proceda a la elaboración de dos comunicaciones; una para la Superintendencia de Telecomunicaciones y otra al archivo general de la SENATEL, adjuntando el Certificado de Registro de Enlaces de MDBA (Explotación); comunicaciones que luego de ser revisadas y aprobadas por el subdirector y director son despachadas.

Si la información presentada en el estudio de ingeniería es incorrecta o incompleta, entonces el proceso es el siguiente.

1. Se elabora el oficio, detallando los problemas encontrados, así como los documentos faltantes.
2. El oficio es remitido por medio físico y magnético para su respectiva revisión y aprobación, por parte del supervisor, subdirector y director.
3. Finalmente la secretaria despacha el oficio y lo almacena en el archivo de la DGGER.
4. En caso de que no se corrijan los errores, en la primera petición de la SENATEL de documentos faltantes, el trámite se archiva.

2.3. DESCRIPCIÓN Y ANÁLISIS DE LAS NORMATIVAS

A continuación se presenta un análisis de los dos documentos técnicos y legales los cuales permiten realizar la validación de la información que ingresa el concesionario. Cabe mencionar que este análisis fue necesario para la elaboración de las historias de usuario presentes en el Capítulo III.

2.3.1. INSTRUCTIVO FORMULARIOS DE CONCESIÓN DE FRECUENCIAS

La SENATEL ha establecido formatos para los formularios concernientes a cada concesión, renovación o modificación de una banda de frecuencias. De acuerdo al requerimiento del usuario se debe llenar determinados formularios que se encuentran disponibles en la página web de la SENATEL [20], tanto como para Servicio MT-RDV, como para Sistemas MDBA. El Instructivo Formularios de Concesión de Frecuencias [21] describe detalladamente cómo llenar cada uno de estos formularios.

A continuación se presenta un resumen del contenido de cada formulario y de cómo los analistas realizan las validaciones antes de ingresar el trámite al software SPECTRAplus, es necesario mencionar que según lo tratado en el Plan de Proyecto de Titulación el presente Proyecto no validará imágenes ni los formularios RC-3B, RC-13A y RC-14A.

Todos los formularios son para el Servicio MT-RDV y Sistemas MDBA, a menos que se indique lo contrario.

2.3.1.1. Formulario para Información Legal (RC-1A)

Este formulario es solo para Servicio MT-RDV, en el que se debe ingresar: el objeto de la solicitud que puede ser una concesión, renovación, modificación, el tipo de uso, sistema y servicio, además de los datos del solicitante y profesional técnico.

Se debe adjuntar la copia de la cédula de identidad del solicitante, así como una copia de la licencia profesional del profesional técnico el cual debe ser Ingeniero en Electrónica y Telecomunicaciones. El analista de la DGGER deberá verificar que se encuentre la documentación adjunta y las respectivas firmas del solicitante y profesional técnico avalando la información presentada.

2.3.1.2. Formulario para Información Legal para Sistemas de Modulación Digital de Banda Ancha (RC-1B)

Este formulario es solo para sistemas de MDBA ya sea para servicio privado o de explotación, en el cual se debe especificar el objeto de la solicitud, es decir si es un registro (primera vez), renovación o modificación, además del tipo de uso (privado o explotación). De la misma manera que en el formulario RC-1A se deberán verificar los datos del solicitante y profesional técnico, con sus respectivos documentos legales adjuntos.

2.3.1.3. Formulario para Información de la Infraestructura del Sistema de Radiocomunicaciones (RC-2A)

En este formulario se ingresa información de la infraestructura de los sistemas que se van a usar, como:

- La estructura del sistema de radiocomunicaciones y sus características.
- Ubicación geográfica: en la cual debe constar la provincia, ciudad o cantón y detallar la localidad exacta.
- Coordenadas geográficas: en el formato de grados minutos y segundos, utilizando **N** o **S** para indicar latitud norte o sur y **W** para longitud oeste. Y se debe adjuntar una copia de un mapa cartográfico escala 1:50.000 donde se indique la ubicación de la estructura de transmisión.

- Tipo de fuente a utilizar y protecciones eléctricas, entre otras.

El analista de la DGER usa herramientas de cartografía como Google Earth donde ingresa las coordenadas geográficas y verifica las direcciones de cada una de las estructura.

2.3.1.4. Formulario para Información de Antenas (RC-3A)

En este formulario se ingresa información de las características técnicas de las antenas como tipo, marca, modelo, rango de frecuencias, impedancia, polarización, ganancia, diámetro, ángulo de elevación y alturas. Deberá adjuntarse los catálogos *data sheets* de las antenas a ser usadas.

El analista de la DGER deberá revisar los rangos de cada una de las características de las antenas, verificando que no sobrepasen los valores establecidos en los respectivos *data sheets*, además de asegurarse que la antena se encuentre en la base de datos del software SPECTRAplus. Si no se encuentra en la base de datos se procede al ingreso de la antena requerida por parte del ingeniero de la DGER encargado.

2.3.1.5. Formulario para Patrones de Radiación de Antenas (RC-3B)

En este formulario se deben presentar los gráficos de los patrones de radiación de las antenas presentadas en el formulario RC-3A. Especificando la marca y el modelo. Cabe mencionar que en el presente Proyecto no se validarán imágenes, por lo tanto no se validará este formulario.

2.3.1.6. Formulario para Información de Equipamiento (RC-4A)

En este formulario se ingresa el tipo de estación (repetidora, transmisor, estación base, fija, móvil, portátil, etc.), así como la marca, modelo, ancho de banda, tipo de modulación, velocidad de transmisión, potencia de salida, rango de operación, entre otras. El equipo deberá estar homologado por la SUPERTEL y deberá adjuntarse los catálogos (*data sheets*) de todos los equipos a ser usados. De igual manera que con las antenas el analista deberá verificar valores y además asegurarse de que el

equipo se encuentre en la base de datos del SPECTRAplus. De no encontrarse el equipo en la base de datos se envía un memo a la SUPERTEL para la respectiva homologación del equipo.

2.3.1.7. Formulario para el Tipo de Operación y Número de Estaciones de Servicios FM-RDV (RC-5A)

Este formulario es solo para Servicio MT-RDV, en el cual se debe especificar:

- Características de operación por cada circuito, el número de circuitos, la banda de frecuencia y el rango de frecuencia requerido.
- La región a operar, se debe especificar la provincia o provincias y la ciudad o ciudades principales donde se desea operar.
- El modo de operación, es decir si es una comunicación simplex, semidúplex o full dúplex.
- El horario de operación, en donde se debe especificar el valor exacto, en caso de tener operación en horas fraccionarias debe aproximarse al entero superior.
- Potencia de salida, en la que se debe especificar la mayor potencia en *Watts* de operación del sistema.
- Se debe especificar el ancho de banda con la que operará el sistema.
- La clase de emisión debe especificarse considerando la normativa del Reglamento de Radiocomunicaciones de la UIT.
- Se debe especificar el número de estaciones por circuito (repetidoras, fijas, móviles, portátiles).
- Las características de las estaciones repetidoras: en donde se debe ingresar el indicativo, la acción que se va a realizar con la estación indicada, que puede ser autorización, modificación, incremento o eliminación, la estructura asociada de acuerdo al formulario RC-2A, la antena asociadas de acuerdo al formulario RC-3A, la altura efectiva de la antena y el equipo utilizado de acuerdo al formulario RC-4A.
- Las características de las estaciones fijas y móviles: se debe completar la misma información que en las estaciones repetidoras.

El analista de la DGGER hace uso del Instructivo de Formularios [21] para validar toda la información requerida en este formulario.

2.3.1.8. Formulario para Sistemas de Modulación Digital de Banda Ancha Enlaces Punto-Punto (RC-9A)

Este formulario es solo para sistemas MDBA, en donde se registra la siguiente información:

- Clase de sistema: pudiendo ser privado o de explotación.
- Características técnicas y de operación del sistema fijo punto-punto: Se debe completar el número de enlace, la banda de frecuencia y el tipo de operación y la distancia del enlace.
- Características de las estaciones fijas: en donde se debe ingresar el indicativo, la acción que se va a realizar con la estación indicada, que puede ser autorización, modificación, incremento o eliminación, la estructura asociada a esa estación de acuerdo al formulario RC-2A, la antena asociada a esa estación de acuerdo al formulario RC-3A, la potencia máxima de salida y el equipo utilizado de acuerdo al formulario RC-4A.
- Perfil topográfico y gráfica del perfil topográfico.
- Esquema del sistema: se debe esquematizar todos los equipos de telecomunicaciones a utilizar.

El analista de la DGGER hace uso del Instructivo de Formularios [21] y de la RESOLUCIÓN NO. 417-15-CONATEL-2005 [6] para validar toda la información requerida en este formulario.

2.3.1.9. Formulario para Sistemas de Modulación Digital de Banda Ancha Sistemas Punto-Multipunto (RC-9B)

Se debe ingresar la misma información que se requiere para el formulario RC-9A, pero para sistemas punto-multipunto.

2.3.1.10. Formulario para Cálculos de Propagación (RC-13A)

Este formulario solo se usa para sistemas MDBA, en el cual se ingresa el número de circuito, los perfiles topográficos, el área y radio de cobertura y el esquema del sistema. Cabe mencionar que en el presente Proyecto no se validarán las imágenes, por lo cual los perfiles topográficos y gráficos del sistema deberán ser verificados por el analista de la DGER.

2.3.1.11. Formulario para Esquema del Sistema de Radiocomunicaciones (RC-14A)

Este formulario se usa para Servicio MT-RDV en el cual se realiza un esquema general del sistema, con todas sus respectivas estaciones. Al igual que el formulario RC-3A, RC-13A este formulario no será validado en el presente proyecto.

2.3.1.12. Formulario para Estudio Técnico de Emisiones de RNI (RC-15A)

En este formulario se deben detallar los datos del usuario, como el nombre y la dirección de la empresa además de la ubicación exacta. Se debe adjuntar una copia de un mapa cartográfico escala 1:50.000 donde se indique la ubicación de la estructura de transmisión.

Además debe tener la siguiente información:

- Se debe indicar el rango de frecuencias de operación y los valores correspondientes a la Slim²⁶ Ocupacional y Slim Poblacional de acuerdo a la siguiente tabla
- Cálculo de R^2 , donde $R = \sqrt{x^2 + (h - d)^2}$, siendo x una constante, h la altura y d la distancia
- Cálculo del PIRE²⁷ (máximo).
- Cálculo del *slim* teórico.
- Certificación del profesional técnico (responsable técnico).

²⁶ **Slim (Densidad de Potencia Límite):** es la potencia por unidad de superficie normal a la dirección de la propagación de onda electromagnética, en Watts por metro cuadrado (W/m^2).

²⁷ **PIRE (Potencia Isotrópica Radiada Equivalente):** es la cantidad de potencia que emitiría una antena isotrópica teórica (es decir, aquella que distribuye la potencia exactamente igual en todas direcciones) para producir la densidad de potencia observada en la dirección de máxima ganancia de una antena.

- Certificación de la persona natural, representante legal o persona debidamente autorizada.

El analista de la DGGER hace uso del Instructivo de Formularios [21] para verificar valores y después realizar los respectivos cálculos matemáticos necesarios para la validación de este formulario.

2.3.2. NORMA PARA LA IMPLEMENTACIÓN Y OPERACIÓN DE SISTEMAS DE MODULACIÓN DIGITAL DE BANDA ANCHA (RESOLUCIÓN NO. 417-15-CONATEL-2005)

En esta Norma [6], además de las definiciones y disposiciones generales en esta norma se especifica las características técnicas de los sistemas MDBA.

Las validaciones más prioritarias a las que hace hincapié esta resolución radican en:

- **Potencia máxima:** de acuerdo a la ganancia de la antena, al tipo de configuración del sistema (punto-punto o punto-multipunto) y a la banda de operación (en el caso de MDBA son las bandas ICM²⁸) se considera una potencia pico máxima del transmisor.
- **Homologaciones:** todos los equipos deberán estar homologados por la SUPERTEL (Superintendencia de Telecomunicaciones).
- **Emisiones no deseadas:** se consideran los límites de emisiones no deseadas en las Bandas de Operación de los sistemas MDBA que están especificados en la referencia [6].

2.4. ANÁLISIS DEL FORMATO DE ARCHIVO XML QUE SOPORTA EL SOFTWARE SPECTRAPLUS

²⁸ **ICM** son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.

El software SPECTRAplus es una herramienta informática para la gestión del espectro radioeléctrico que ofrece una solución para los procedimientos de concesión de licencias de algunos servicios de radiofrecuencias. Cumple con funciones como:

- Almacenar toda la información en una base de datos de gestión centralizada.
- Procesamiento y gestión de solicitudes de frecuencias.
- Procesamiento de solicitudes específicas de servicios de radiodifusión.
- Cálculo automático de tarifas y procesamiento de facturas.
- Generación de documentos de títulos habilitantes, licencias, facturas, recordatorios, etc.
- Sistema integrado de gestión de documentos.
- Preparación de estadísticas sobre el uso del espectro y derechos de licencia.
- Permite la importación y exportación de los trámites en formato XML.

El software SPECTRAplus posee diferentes interfaces gráficas dependiendo del tipo de trámite a realizar, en cada una de estas interfaces existen campos en los que el analista de la DGER debe ingresar manualmente la información presentada a través de documentos físicos por el concesionario.

El software SPECTRAplus requiere de información específica presentada por el concesionario, razón por la cual el presente Proyecto realiza una validación en los formularios que poseen información indispensable para que se pueda generar un archivo en formato XML importable a mencionado software y de esta manera generar el Título Habilitante de Servicio MT-RDV y el registro de enlaces de Sistemas MDBA. Cabe señalar que el SPECTRAplus no permite el ingreso de imágenes, razón por la cual no se realiza el procesamiento de imágenes.

Como se mencionó, el SPECTRAplus permite la exportación de un archivo XML con información concerniente a cada trámite realizado, de esta manera el nombre de un campo, por ejemplo de un *textbox*, de la interfaz gráfica del SPECTRAplus corresponderá a una etiqueta del archivo XML exportado. Teniendo en cuenta lo antes mencionado se procedió a identificar los nombres de los campos requeridos para la elaboración del archivo XML.

Un ejemplo de la relación entre los nombres de los campo y las etiquetas del archivo XML se puede ver en la Figura 2-4, en el cual se aprecia las propiedades de un control de un formulario que son los que debe tener el archivo XML.

Una vez identificados los campos se realizó la exportación de un trámite tanto de Servicio MT-RDV, como de MDBA con el fin de analizar el formato XML que soporta el SPECTRAplus y ver qué información presentada por el concesionario debe ir en cada etiqueta del archivo XML, para su posterior importación a través del prototipo desarrollado.

En el Anexo A se indican los nombres de las etiquetas del archivo XML exportado correspondiente a la información presentada en la documentación física por el concesionario.

The image shows a software window titled 'Propiedades of Item LI_CAT' with a table of attributes and values. A red circle highlights the 'LI_CAT' attribute with the value '18'. Another red circle highlights the '<LI_CAT>2</LI_CAT>' tag in the XML code snippet to the right, with a red arrow pointing from the dialog box to the XML tag.

Attributes	Values
Name	LI_CAT
Sequence Number	18
X coordinate	525.000
Y coordinate	68.000
Width	110.000
Height	18.000
Displayed	TRUE
Automatic Hint	FALSE
Hint text	
Base Table	TRUE
Primary Key	FALSE
Updateable	TRUE
Queryable	TRUE
Required	FALSE
Update if NULL	FALSE
Enabled	TRUE
Insert Allowed	TRUE

```

</LETTER>
<LICENCE>
  <LI_ID>14840</LI_ID>
  <AD_AD_ID>21863</AD_AD_ID>
  <LI_LWRDATE>2011-02-21T00:00:00-05:00</LI_LWRDATE>
  <LI_REQ_DATE>2011-02-21T00:00:00-05:00</LI_REQ_DATE>
  <LI_LIC_DATE>2011-02-21T00:00:00-05:00</LI_LIC_DATE>
  <LI_CANCEL_DATE>2016-02-20T00:00:00-05:00</LI_CANCEL_DATE>
  <LI_PAY_CATEGORY>00</LI_PAY_CATEGORY>
  <LI_CAT>2</LI_CAT>
  <LI_PERIOD_UNIT>6</LI_PERIOD_UNIT>
  <ACD_ACD_ID>1991</ACD_ACD_ID>
</LICENCE>
<LINK_EQ>
  <LEQ_NUM>1302974</LEQ_NUM>
  <EQ_EQ_ID_TX>1030727</EQ_EQ_ID_TX>
  <EQ_EQ_ID_RX>1030724</EQ_EQ_ID_RX>
  <LEQ_TX_TYPE>1</LEQ_TX_TYPE>
  <LEQ_TRANSFER_RATE_DUNIT>Mbit/s</LEQ_TRANSFER_RATE_DUNIT>
  <ACD_ACD_ID>1991</ACD_ACD_ID>
</LINK_EQ>

```

Figura 2-4 Nombres de las variables de los campos de SPECTRAplus

Un esquema resumido del archivo en formato XML que acepta la aplicación SPECTRAplus es la que se muestra en la Figura 2-5.

El archivo XML exportable presenta la información de forma estructurada, contiene varias etiquetas, las cuales tienen los datos de determinado formulario, como por ejemplo la etiqueta <ADDRESS> indica la dirección de las estructuras, información que el concesionario llena en el Formulario-2A.

```

<?xml version="1.0" standalone="yes" ?>
-<SPECTRAPLUS>
-<xs:schema id="SPECTRAPLUS" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
-<xs:element name="SPECTRAPLUS" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
-<xs:complexType>
-<xs:choice minOccurs="0" maxOccurs="unbounded">
+<xs:element name="ADDRESS">
+<xs:element name="ADDRESS_LINK">
-<xs:sequence>
<xs:element name="ADL_ID" type="xs:decimal" minOccurs="0" />
<xs:element name="SV_SV_ID" type="xs:decimal" minOccurs="0" />
<xs:element name="AD_AD_ID_FROM" type="xs:decimal" minOccurs="0" />
/>
<xs:element name="AD_AD_ID_TO" type="xs:decimal" minOccurs="0" />
<xs:element name="ADL_LINK_TYPE" type="xs:decimal" minOccurs="0" />
/>
<xs:element name="ADL_CRDATE" type="xs:dateTime" minOccurs="0" />
<xs:element name="ADL_COMMENT" type="xs:string" minOccurs="0" />
<xs:element name="ACD_ACD_ID" type="xs:decimal" minOccurs="0" />
</xs:sequence>
</xs:complexType>

```

Figura 2-5 Fragmento del archivo XML exportado del SPECTRAPLUS

Cada etiqueta tiene identificadores que permiten la correspondencia con las demás etiquetas dentro del archivo XML, por ejemplo la etiqueta `<SITE_DATA>` que corresponde a las coordenadas geográficas, contiene el identificador `<SID_ID>` que permite la asociación con la etiqueta `<ADDRESS>` a través del identificador `<SID_SID_ID>`.

Cada identificador es del tipo *decimal* ya que son números que constan de entre dos a ocho cifras, existen además otros tipos de datos en cada etiqueta como *string* para texto y el *dateTime* para las fechas.

Cabe mencionar que algunos datos tienen la propiedad `NOT-NULL` es decir que obligatoriamente deberán contener un valor.

2.5. HERRAMIENTAS DE DESARROLLO

Existen dos tecnologías que son las más empleadas para el desarrollo de servicios Web, estas son J2EE²⁹ y .NET, cada una presenta ventajas y desventajas respecto a la otra.

²⁹ **JEE (Java Platform Enterprise Edition):** es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java.

2.5.1. VENTAJAS DE .NET FRENTE A J2EE

- .NET ofrece la posibilidad de usar múltiples lenguajes mientras que con J2EE solo se puede trabajar con Java.
- Las herramientas de desarrollo de .NET (Microsoft Visual Studio) son relativamente más sencillas de utilizar que las disponibles para J2EE.
- .NET posee un mayor soporte para el desarrollo de clientes (como por ejemplo ASP.NET), con el cual se puede obtener un mayor rendimiento que los SERVLETS³⁰ y JSP³¹ de Java.
- .NET se ha desarrollado pensando en los servicios Web además que Microsoft ofrece mejoras continuas de ASP y ASP.NET.

2.5.2. VENTAJAS DE J2EE FRENTE A .NET

- J2EE es una tecnología que generalmente usa código abierto mientras que .NET es propiedad de Microsoft en el cual no todo su código es público.
- Las aplicaciones desarrolladas en J2EE pueden correr sobre cualquier sistema operativo mientras que en .NET generalmente corren sobre sistemas Microsoft.

Para el desarrollo del servicio Web del presente proyecto se optó por usar el framework de .NET, debido a las ventajas que ofrece sobre J2EE.

2.5.3. WCF (WINDOWS COMMUNICATION FOUNDATION)

Es una herramienta para servicios Web que propone una tecnología flexible que permite crear aplicaciones SOA, ofrece varias ventajas como el envío de mensajes (simples o complejos) de un extremo de servicio a otro, permite mantener niveles de seguridad (varios transportes y codificaciones), mensajes confiables y en cola, compatibilidad con AJAX³² y REST y extensibilidad. Por todas estas características

³⁰ **SERVLET** es una clase Java, comúnmente para extender las aplicaciones alojadas por servidores web.

³¹ **JSP (JavaServer Pages)**: es una tecnología Java que ayuda a la creación de páginas web dinámicas basadas en HTML, XML entre otros tipos de documentos.

³² **AJAX (Asynchronous JavaScript And XML)**: es una técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan en el cliente, es decir, en el navegador.

se escogió a WCF como la herramienta para el desarrollo de los servicios Web REST.

Existen otras herramientas como ASP.NET MVC que permite el desarrollo de aplicaciones web fáciles de probar y mantener que sigue el Modelo Vista Controlador (MVC), lo cual permite mantener una separación entre: vistas para la interfaz de usuarios, controladores para el control de los datos especificados por el usuario y modelos para la lógica del negocio.

Generalmente las aplicaciones ASP.NET MVC son usadas para pruebas unitarias y para aplicaciones para sitios web móviles.

2.5.4. WPF (WINDOWS PRESENTATION FOUNDATION)

Para la aplicación de escritorio se optó por WPF ya que es un entorno de trabajo del framework de .NET para crear aplicaciones cliente interactivas permitiendo el uso de una amplia gama de recursos, controles, gráficos, diseño, enlace de datos, documentos y seguridad. WPF utiliza el lenguaje XAML³³ para proporcionar un modelo declarativo para la programación de aplicaciones.

2.5.5. ASP.NET

En un principio solo se requería de aplicaciones con páginas web estáticas que contenían documentos HTML almacenados en el servidor, pero los nuevos requerimientos de que se genere el contenido de las páginas en forma dinámica ha generado que el software se ejecute en el servidor cada vez que el cliente se conecte a la página web deseada, para lo cual actualmente se usan varias herramientas como ASP.NET que está incluida en la plataforma de .NET

ASP.NET es una tecnología que permite desarrollar aplicaciones web dinámicas fácilmente, además de ofrecer compatibilidad con AJAX, LINQ (*Language-*

³³ **XAML** (eXtensible Application Markup Language): es el lenguaje de formato para la interfaz de usuario para WPF y Silverlight, el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET. XAML es un lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuarios visuales ricas desde el punto de vista gráfico.

Integrated Query), XML y servicios Web, variedad en controles web, permite el uso de varios lenguajes para el desarrollo de la aplicación (C#, VB.NET o J#), adaptación de código (el mismo código que se emplea para el uso en Internet Explorer puede emplearse para *Pocket Internet Explorer*) y finalmente da mayor velocidad, potencia y seguridad en comparación con PHP o JSP.

2.5.6. SERVIDOR WEB

Un servidor web es un sistema informático concurrente, es decir, capaz de alojar varios recursos como archivos, imágenes, videos, ejecutables, entre otros, permitiendo acceder a estos de manera simultánea, y además es orientado a la conexión porque usa HTTP que corre sobre TCP.

Los servidores web más conocidos son IIS³⁴, Apache³⁵, Glassfish³⁶ y Apache Tomcat³⁷.

Se prefirió usar IIS 7, ya que a diferencia de las versiones anteriores se cuenta con nuevas características como:

- **Rendimiento:** Nueva interfaz de administración vía web, mayores capacidades en cuanto a la configuración, activación de funciones, se pueden activar o desactivar módulos según las necesidades.
- **Compatibilidad:** Ofrece compatibilidad mediante el protocolo abierto FastCGI³⁸.
- **Carga:** IIS7 puede soportar hasta 100.000 sitios web en un solo servidor, se ha probado el aumento de densidad en el número de aplicaciones en un solo servidor y en número de aplicaciones simultáneas activas.

³⁴ **IIS** (Internet Information Services) es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Este servicio convierte a un computador en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

³⁵ **Apache** es un servidor web de código abierto para plataformas Unix, Microsoft Windows, Macintosh entre otras.

³⁶ **GlassFish** es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems

³⁷ **Tomcat** es un servidor web con soporte de servlets y JSP.

³⁸ **FastCGI** es un protocolo para interconectar programas interactivos con un servidor web.

- **Extensibilidad:** se han desarrollado dos nuevas API para crear módulos; MWA (*Microsoft Web Administration*) y MWM (*Microsoft Web Management*) [22].

2.5.7. TEAM FOUNDATION SERVICE

Es una plataforma de desarrollo en la nube creada por Microsoft con el fin de permitir la colaboración entre los desarrolladores y un seguimiento del estado del trabajo, por lo cual se utilizó esta herramienta para el desarrollo del proyecto con la metodología Scrum. En el capítulo tres se presenta información detallada de su funcionamiento.

CAPÍTULO III

3. DISEÑO Y DESARROLLO DEL SISTEMA

3.1. INTRODUCCIÓN

En este capítulo se realiza el diseño del Sistema basándose en la metodología Scrum la cual ayuda a describir, entender mejor el proceso de trabajo, conocer los problemas que puedan surgir y tomar decisiones respecto a los posibles cambios que se puedan proponer. Además se utiliza la herramienta TFS *on cloud* principalmente para la administración de las tareas asignadas y el uso apropiado de la metodología.

Se describe el desarrollo de cada uno de los componentes del Sistema Distribuido sobre un único Entorno de Desarrollo Integrado (IDE).

Finalmente se describen las capas arquitectónicas del software basado en la arquitectura SOA, además se presentan los respectivos diagramas que ayudan al entendimiento del software desarrollado.

3.2. DESCRIPCIÓN DE LOS ARTEFACTOS UTILIZADOS EN EL SISTEMA DISTRIBUIDO

3.2.1. HISTORIAS DE USUARIO

En Scrum las historias de usuario o *Product Backlogs* capturan los requerimientos de los usuarios, son la base para el desarrollo del software ya que deben presentar las funcionalidades del sistema e indicar como debería reaccionar el sistema ante una petición, por lo tanto es necesario ser muy particular y específico en cada una de ellas. A diferencia de los casos de uso las historias de usuario son flexibles, es decir se puede realizar modificaciones continuas durante el desarrollo del proyecto, además no se requiere de lenguaje técnico para describirlas porque es el cliente el cual las debe realizar.

El análisis que se realizó en el Capítulo II fue empleado para la generación de las historias de usuario en las cuales se empleó un formato de tabla que incluye: el usuario, el número, el nombre, la prioridad, el tiempo estimado, la iteración asignada, la descripción de la funcionalidad requerida, y las observaciones de la historia de usuario.

En el Anexo B se muestran todas las historias de usuario del presente Proyecto. En la Tabla 3-1 se muestra como ejemplo una de ellas.

Usuario 1	Historia 1	
Nombre historia: Formulario RC-1B, Solicitud		
Prioridad en negocio: 1	Iteración asignada: 1	
Tiempo estimado: 1 día		
Descripción: En el formulario de información legal RC-1B el usuario deberá especificar el objeto de la solicitud y el tipo de uso del sistema de una lista que se le presentará por cada uno.		
Observaciones: Los dos campos son obligatorios.		

Tabla 3-1 Ejemplo de historia de usuario

Desde el punto de vista del desarrollo en sí, el Servicio Web debería tener la prioridad más alta, es decir se debería comenzar con la implementación del servicio. Sin embargo en la metodología prima la interacción entre el *Product Owner*, el *Team* y el *Scrum Manager*, para el *Product Owner* la importancia no radica en la tecnología y características del Servicio Web, sino más bien en la implementación de las aplicaciones, principalmente en la aplicación de escritorio por lo cual se decidió establecer la prioridad más alta a la aplicación de escritorio, luego a la aplicación web y finalmente al Servicio Web.

3.2.2. TEAM FOUNDATION SERVICE (TFS)

TFS permite trabajar con la metodología Scrum para estructurar y administrar todas las etapas del desarrollo, desde el análisis de los requerimientos hasta la fase de pruebas o *testing*, permitiendo la interacción de las personas involucradas en el proyecto.

Cabe señalar que al ser Scrum una metodología ágil, permite ser modificable a la necesidad del proyecto, para el presente Proyecto se ha adaptado debido a que solo se cuenta con dos programadores que forman parte del *team*.

La Figura 3-1 muestra los roles de las personas implicadas en el desarrollo.

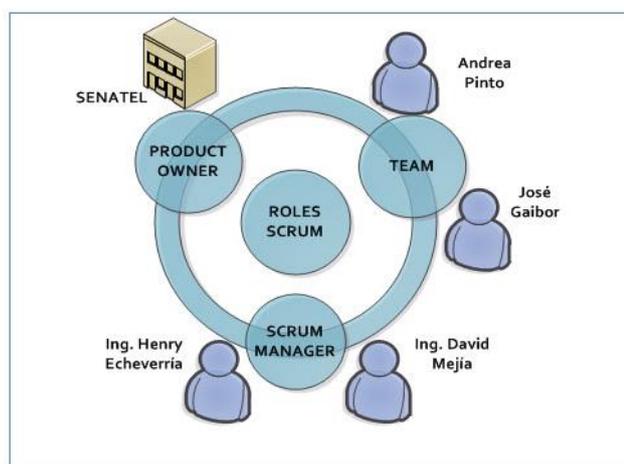


Figura 3-1 Roles de Scrum

TFS cuenta con características como:

- Entorno dinámico de trabajo
- Accesibilidad
- Disponibilidad
- No necesita de infraestructura previa
- No necesita de instalar o configurar un servidor
- Permite el uso de cualquier herramienta de desarrollo como Visual Studio, Eclipse o Xcode³⁹.

Cabe mencionarse que TFS cuenta con una extensa documentación de uso de su plataforma, disponible en [23].

Para tener acceso a TFS se debe ingresar el enlace <http://tfs.visualstudio.com/> y registrarse con una cuenta de Microsoft, se permite hasta un máximo de cinco miembros por proyecto sin costo alguno. A continuación se asigna un nombre al proyecto, para el desarrollo del prototipo se

³⁹ **Xcode** es el entorno de desarrollo integrado (IDE) de Apple Inc.

usó el nombre `Proyecto Senatel` al cual los miembros registrados podrán acceder a través del enlace `https://proyectosenatel.visualstudio.com`.

3.2.2.1. Historias de usuario en TFS

Para comenzar con el desarrollo de cualquier proyecto es necesario ingresar todas las historias de usuario al TFS, por lo cual las historias de usuario para el desarrollo del prototipo fueron ingresadas dentro de tres componentes o *features*:

- **Servicios Web** basados en SOA para validar la información
- **Aplicación de escritorio** que permita el ingreso de los datos offline al concesionario
- **Aplicación web** para presentar los datos al analista de la SENATEL

A cada historia de usuario o *Product Backlog* se la debe descomponer en tareas o *tasks* más específicas para asignar un tiempo y un responsable para la elaboración de cada una de ellas.

En la Figura 3-2 se puede visualizar la jerarquización de las actividades a desarrollarse, además del estado en el cual se encuentran.

Work Item Type	Title	State
Feature	▶ Servicios Web	In Progress
Feature	▶ Aplicación de escritorio	In Progress
Feature	◀ Aplicación Web	In Progress
Product Backlog Item	▶ Reconstruir formularios	Committed
Product Backlog Item	◀ Aplicacion Web para MDBA	New
Task	▶ Crear una interfaz gráfica con los datos del trámite	To Do

Figura 3-2 Tareas TFS

TFS permite la visualización de las tareas en un tablero *board* para verificar el avance en cada una de ellas y gestionar el proyecto de manera iterativa, flexible e incremental, en la Figura 3-3 se puede observar la persona responsable, el tiempo y el estado de la tarea.

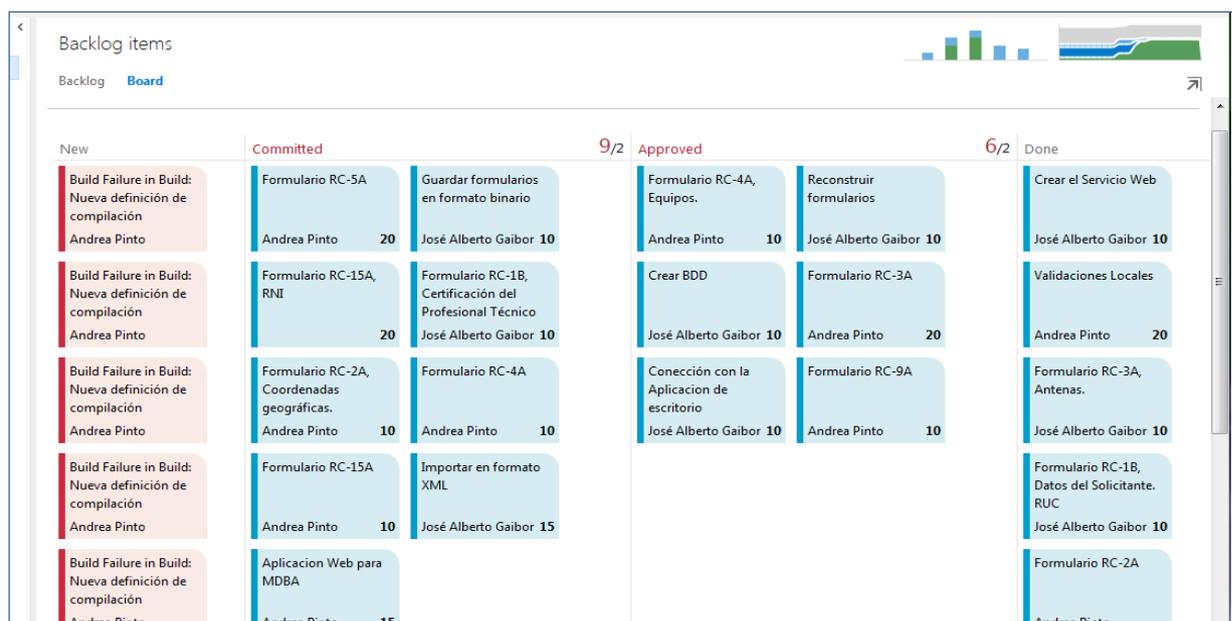


Figura 3-3 Tablero de TFS

3.2.2.2. Sprints en TFS

El TFS es una herramienta que permite gestionar los *sprints* y las reuniones de trabajo, asignando fechas para cada una de ellas como se indica en la Figura 3-4.

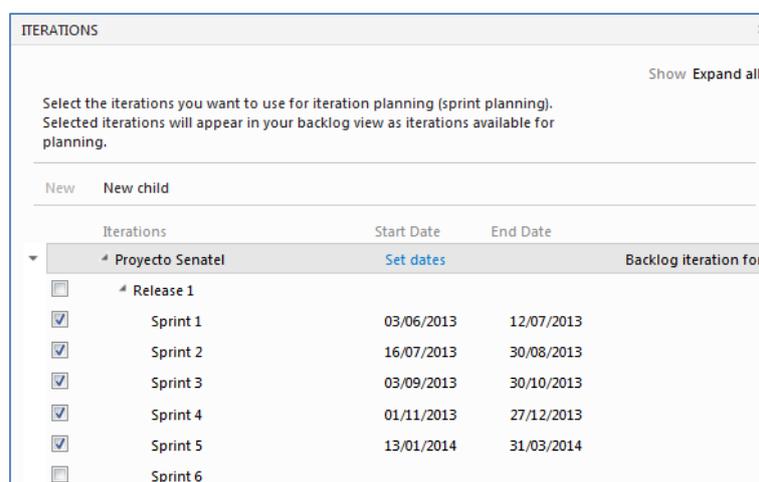


Figura 3-4 Administración de Sprints

En el presente Proyecto se registraron 5 *sprints*, en los cuales se repartieron 63 *Product Backlogs* (33 de MDBA y 30 de MT-RDV) y 103 tareas ya que en cada *Product Backlog* al menos existe una tarea.

La Tabla 3-2 indica la cantidad de *Product Backlogs* y tareas asignadas a cada uno de los *sprints* llevados a cabo.

N° <i>Sprint</i>	Tiempo	N° <i>Product Backlog</i>	N° Tareas
1	3 de junio al 12 de julio de 2013 (30 días)	16	24
2	16 de julio al 30 de agosto de 2013 (34 días)	14	20
3	3 de septiembre al 30 de octubre de 2013 (42 días)	11	21
4	1 de noviembre de 2013 al 27 de diciembre de 2014 (41 días)	10	14
5	13 de enero al 31 de marzo de 2014 (55 días)	12	24

Tabla 3-2 Estadísticas TFS

En la Figura 3-5 se muestra los *Product Backlogs* y tareas asignados al *Sprint 5*.

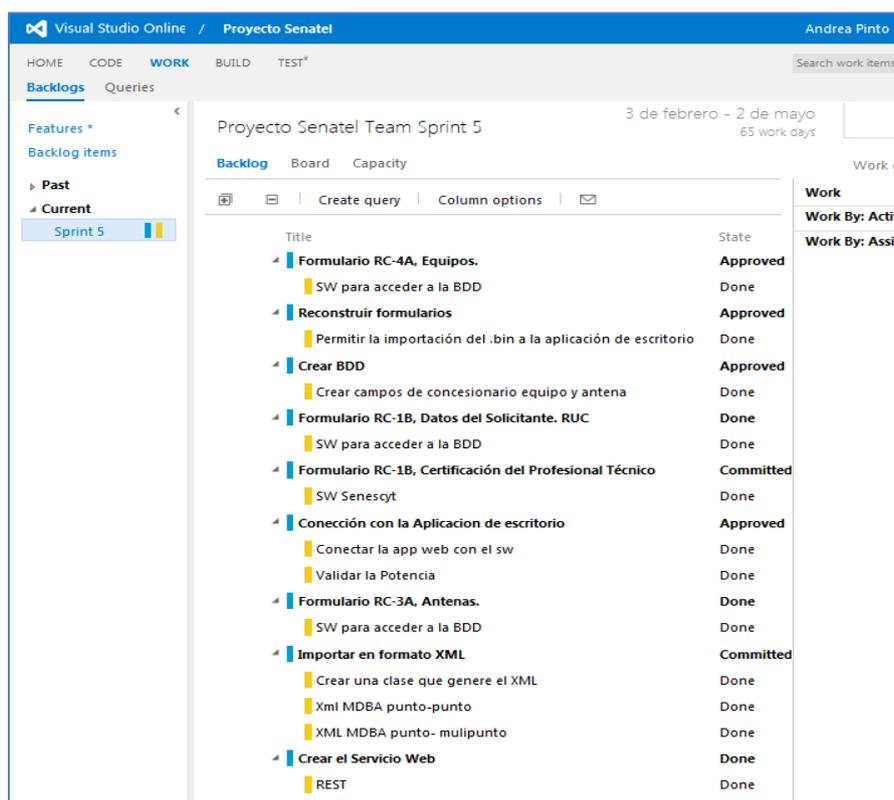


Figura 3-5 *Sprint 5* en TFS

3.2.2.2.1. *Sprint 1*

Tiempo establecido: 3 de junio al 12 de julio de 2013 (30 días)

Observaciones: En el *Sprint 1* se llevó a cabo la implementación de las interfaces de la aplicación de escritorio (Formularios: RC-1A, RC-1B, RC-2A) como lo muestra la Tabla 3-3.

Revisión del Sprint 1: Las 16 Historias de usuario planificadas para este *sprint* se finalizaron en el tiempo establecido y al no existir realimentación del *Product Backlog* se continúa el desarrollo con las tareas destinadas al *Sprint 2*.

Servicio	N° HU	N°	Nombre del <i>Product Backlog</i>	Tareas asignadas
MDBA	1	1	Crear ventana Principal MDBA	Crear ventana Principal MDBA
	2	2	Crear Formulario RC-1B	Crear formulario RC-1B
	3	3	Formulario RC-1B, Solicitud	Crear los campos de solicitud en el formulario RC-1B
	4	4	Formulario RC-1B, Datos del Solicitante	Crear los campos de los datos del solicitante en el formulario RC-1B Realizar validación local
	5	5	Formulario RC-1B, Datos del Profesional Técnico	Crear los campos de los datos del Profesional Técnico en el formulario RC-1B Realizar validación local
	6	6	Crear Formulario RC-2A	Crear formulario RC-2A y control personalizado RC-2A
	7	7	Formulario RC-2A, Información de la Estructura	Crear los campos de la información de la estructura en el formulario RC-2A Realizar validación local
	8	8	Formulario RC-2A, Coordenadas geográficas	Crear los campos de las coordenadas geográficas en el formulario RC-2A Realizar validación local
MT-RDV	1	9	Crear ventana Principal MT-RDV	Crear ventana Principal MT-RDV
	2	10	Crear Formulario RC-1A	Crear formulario RC-1A
	3	11	Formulario RC-1A, Solicitud	Crear los campos de solicitud en el formulario RC-1A
	4	12	Formulario RC-1A, Datos del Solicitante	Crear los campos de los datos del solicitante en el formulario RC-1A Realizar validación local
	5	13	Formulario RC-1A, Datos del Profesional Técnico	Crear los campos de los datos del Profesional Técnico en el formulario RC-1A Realizar validación local
	6	14	Crear Formulario RC-2A	Crear formulario RC-2A y control personalizado RC-2A
	7	15	Formulario RC-2A, Información de la Estructura	Crear los campos de la información de la estructura en el formulario RC-2A Realizar validación local
	8	16	Formulario RC-2A, Coordenadas geográficas	Crear los campos de las coordenadas geográficas en el formulario RC-2A Realizar validación local

Tabla 3-3 *Sprint 1*

3.2.2.2.2. *Sprint 2*

Tiempo establecido: 16 de julio al 30 de agosto de 2013 (34 días)

Observaciones: En el *Sprint 2* se continuó con la implementación de las interfaces de la aplicación de escritorio (Formularios: RC-3A, RC-4A, RC-5A, RC-9A) como lo muestra la Tabla 3-4.

Servicio	N° HU	N°	Nombre del <i>Product Backlog</i>	Tareas asignadas
MDBA	9	1	Crear Formulario RC-3A	Crear formulario RC-3A y control personalizado RC-3A
	10	2	Formulario RC-3A, Antenas	Crear los campos de los datos de las antenas en el formulario RC-3A
				Realizar validación local
	11	3	Crear Formulario RC-4A	Crear formulario RC-3A y control personalizado RC-4A
	12	4	Formulario RC-4A, Equipos	Crear los campos de los datos de los equipos en el formulario RC-4A
				Realizar validación local
	13	5	Crear Formulario RC-9A	Crear formulario RC-9A y control personalizado RC-9A
	14	6	Formulario RC-9A, Sistemas Punto-punto	Crear los campos de la información de los Sistemas Punto-punto
Realizar validación local				
15	7	Formulario RC-9A, Estaciones Fijas	Crear los campos de la información de las Estaciones Fijas en el formulario RC-9A	
16	8	Formulario RC-9A, Perfil topográfico	Crear los campos de la información del Perfil topográfico en el formulario RC-9A	
MT-RDV	9	9	Crear Formulario RC-3A	Crear formulario RC-3A y control personalizado RC-3A
	10	10	Formulario RC-3A, Antenas	Crear los campos de los datos de las antenas en el formulario RC-3A
				Realizar validación local
	11	11	Crear Formulario RC-4A	Crear formulario RC-3A y control personalizado RC-4A
	12	12	Formulario RC-4A, Equipos	Crear los campos de los datos de los equipos en el formulario RC-4A
				Realizar validación local
	13	13	Crear Formulario RC-5A	Crear formulario RC-5A y control personalizado RC-5A
14	14	Formulario RC-5A	Crear los campos del formulario RC-5A	
			Realizar validación local	

Tabla 3-4 *Sprint 2*

Revisión del Sprint 2: Las 14 Historias de usuario se finalizaron en el tiempo establecido, el *Product Owner* decide que se deben implementar nuevas fórmulas (azimut, ángulo de elevación y distancia) en los formularios RC-9A ya implementados, y en los formularios RC-9B por implementarse, por lo cual se planifico realizarlos en el siguiente *Sprint*, aumentando el tiempo del *Sprint 3*.

3.2.2.2.3. *Sprint 3*

Tiempo establecido: 3 de septiembre al 30 de octubre de 2013 (42 días)

Observaciones: En el *Sprint 3* se llevó a cabo la implementación de las interfaces de la aplicación de escritorio (Formularios: RC-5A, RC-9B, RC-15A) como lo indica la Tabla 3-5.

Revisión del Sprint 3: Las 11 Historias de usuario se finalizaron en el tiempo establecido y al no existir realimentación del *Product Backlog* se continúa el desarrollo con las tareas destinadas al *Sprint 4*.

Servicio	N° HU	N°	Nombre del <i>Product Backlog</i>	Tareas asignadas
MDBA	17	1	Crear Formulario RC-9B	Crear formulario RC-9B y control personalizado RC-9B
	18	2	Formulario RC-9B, Sistemas Punto-multipunto	Crear los campos de los datos de Sistemas Punto-multipunto
				Realizar validación local
	19	3	Formulario RC-9B, Estación Fija Central	Crear los campos de los datos de la estación Fija Central en el formulario RC-9B
	20	4	Formulario RC-9B, Estaciones Fijas	Crear control personalizado de las Estaciones Fijas para el formulario RC-9B
				Crear los campos de los datos de la estaciones Fijas en el formulario RC-9B
				Realizar fórmulas de azimut, ángulo de elevación y distancia, Formularios RC-9B y RC-9A
21	5	Formulario RC-9B, Perfil topográfico	Crear control personalizado de las Estaciones Fijas para el formulario RC-9B	
			Crear los campos de la información del Perfil topográfico en el formulario RC-9B	
22	6	Formulario RC-15A, RNI	Crear el formulario RC-15A	
			Realizar los cálculos matemáticos	

MT-RDV	15	7	Formulario RC-5A, estaciones repetidoras	Crear control personalizado de las Estaciones repetidoras para el formulario RC-5A
				Crear los campos de los datos de la estaciones repetidoras en el formulario RC-5A
	16	8	Formulario RC-5A, estaciones fijas	Crear control personalizado de las Estaciones Fijas para el formulario RC-5A
				Crear los campos de los datos de la estaciones Fijas en el formulario RC-5A
	17	9	Formulario RC-5A, estaciones móviles	Crear control personalizado de las Estaciones móviles para el formulario RC-5A
				Crear los campos de los datos de la estaciones móviles y portátiles en el formulario RC-5A
	18	10	Formulario RC-5A, estaciones portátiles	Crear control personalizado de las Estaciones móviles y portátiles para el formulario RC-5A
				Crear los campos de los datos de la estaciones móviles y portátiles en el formulario RC-5A
	19	11	Formulario RC-15A, RNI	Crear el formulario RC-15A
				Realizar los cálculos matemáticos

Tabla 3-5 *Sprint 3*3.2.2.2.4. *Sprint 4*

Tiempo establecido: 1 de noviembre de 2013 al 27 de diciembre de 2013 (41 días)

Observaciones: En el *Sprint 4* se llevó a cabo la implementación de varias funcionalidades de la aplicación de escritorio (Imprimir, Guardar y Reconstruir los formularios) y la creación de la aplicación web, como lo indica la Tabla 3-6.

Servicio	N° HU	N°	Nombre del <i>Product Backlog</i>	Tareas asignadas
MDBA	23	1	Imprimir Formularios	Imprimir en formato PDF cada formulario
				Imprimir en formato PDF todo el trámite
	24	2	Guardar formularios	Guardar en formato binario el trámite
	25	3	Reconstruir formularios	Permitir que el archivo binario se cargue nuevamente a la aplicación de escritorio
	26	4	Aplicación web, visualización	Crear una página de autenticación
				Crear una página para visualizar el trámite
27	5	Aplicación Web, memo	Permitir generar un memo al concesionario	

MT-RDV	20	6	Imprimir Formularios	Imprimir en formato PDF cada formulario
				Imprimir en formato PDF todo el trámite
	21	7	Guardar formularios	Guardar en formato binario el trámite
	22	8	Reconstruir formularios	Permitir que el archivo binario se cargue nuevamente a la aplicación de escritorio
	23	9	Aplicación web, visualización	Crear una página de autenticación
				Crear una página para visualizar el trámite
24	10	Aplicación Web, memo	Permitir generar un memo al concesionario	

Tabla 3-6 *Sprint 4*

Revisión del Sprint: Las 10 Historias de usuario se finalizaron en el tiempo establecido y al no existir realimentación del *Product Backlog* se continúa el desarrollo con las tareas destinadas al *Sprint 5*.

3.2.2.2.5. *Sprint 5*

Tiempo establecido: 13 de enero al 31 de marzo de 2014 (55 días)

Observaciones: En el *Sprint 5* se llevó a cabo la implementación de las funcionalidades del Servicio Web, como lo indica la Tabla 3-7.

Servicio	N° HU	N°	Nombre del <i>Product Backlog</i>	Tareas asignadas
MDBA	28	1	Ingresar formularios	Generar el trámite de MDBA en formato XML, importable al SPECTRAplus.
				Ingresar el trámite de MDBA en formato XML a un repositorio en la SENATEL
	29	2	Servicio Web, RUC	Crear Base de datos del RUC
				Crear SW para el RUC
	30	3	Servicio Web, Profesional Técnico	Realizar la consulta a la SENESCYT
				Crear SW para el Profesional Técnico
	31	4	Servicio Web, Coordenadas geográficas	Realizar la consulta a Google Maps
				Crear SW para las Coordenadas geográficas
	32	5	Servicio Web, Antenas	Crear Base de datos de antenas
				Crear SW para antenas
	33	6	Servicio Web, Equipos	Crear Base de datos de equipos
				Crear SW para equipos

MT-RDV	25	7	Ingresar formularios	Generar el trámite de MT-RDV en formato XML, importable al SPECTRAplus.
				Ingresar el trámite de MT-RDV en formato XML a un repositorio en la SENATEL
	26	8	Servicio Web, RUC	Crear Base de datos del RUC
				Crear SW para el RUC
	27	9	Servicio Web, Profesional Técnico	Realizar la consulta a la SENESCYT
				Crear SW para el Profesional Técnico
	28	10	Servicio Web, Coordenadas geográficas	Realizar la consulta a Google Maps
				Crear SW para las Coordenadas geográficas
	29	11	Servicio Web, Antenas	Crear Base de datos de antenas
				Crear SW para antenas
	30	12	Servicio Web, Equipos	Crear Base de datos de equipos
				Crear SW para equipos

Tabla 3-7 *Sprint 5*

Revisión del Sprint: Las 12 Historias de usuario se finalizaron en el tiempo establecido y al no existir realimentación del *Product Backlog* se finaliza el proceso de desarrollo del prototipo de Sistema distribuido.

A continuación se procede a realizar el despliegue de las aplicaciones en el servidor Windows Server 2008, además de las respectivas seguridades y tolerancia a fallos (detalladas en el Capítulo IV), lo cual tomó un tiempo de 30 días. Posteriormente se realizó la prueba final en las instalaciones de la SENATEL el día 02 de Julio de 2014.

Se concluye por lo tanto el desarrollo del Proyecto al recibir la carta de conformidad por parte del *Product Owner* en este caso la SENATEL.

3.2.2.3. Diagramas de TFS

El diagrama de flujo acumulado o CFD (Cumulative Flow Diagram), indica estadísticamente el número de tareas: nuevas, comprometidas, aprobadas y realizadas en función del tiempo, se puede visualizar información hasta 30 semanas posteriores. La Figura 3-6 muestra el Diagrama de flujo acumulativo desde el 2 de Enero de 2014 hasta el 03 de Julio de 2014, mostrando el número de tareas asignadas en ese lapso de tiempo.

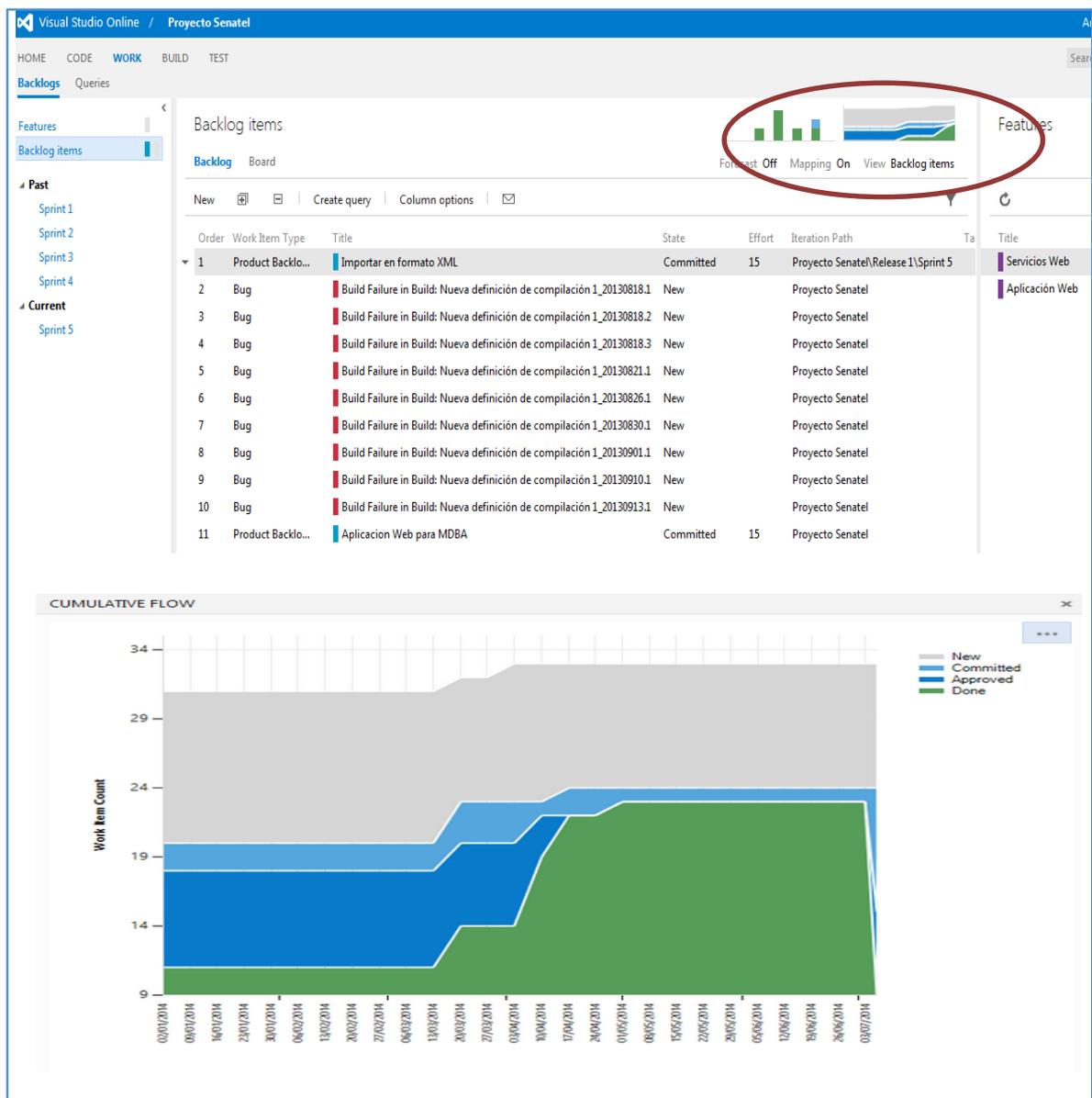


Figura 3-6 Diagrama de flujo acumulativo del TFS

3.2.2.4. Control de código en TFS

Para conectar el código al TFS se debe acceder al proyecto desarrollado en Visual Studio 2012 y en *inicio* seleccionar la opción *conectar a proyecto de equipo*, así se podrá visualizar la pantalla que se indica en la Figura 3-7 en la cual se debe elegir el nombre del proyecto creado en el TFS.

De esta manera se puede guardar el código en TFS en la nube, teniendo acceso a versiones anteriores del código y con la opción de visualizar los cambios efectuados.

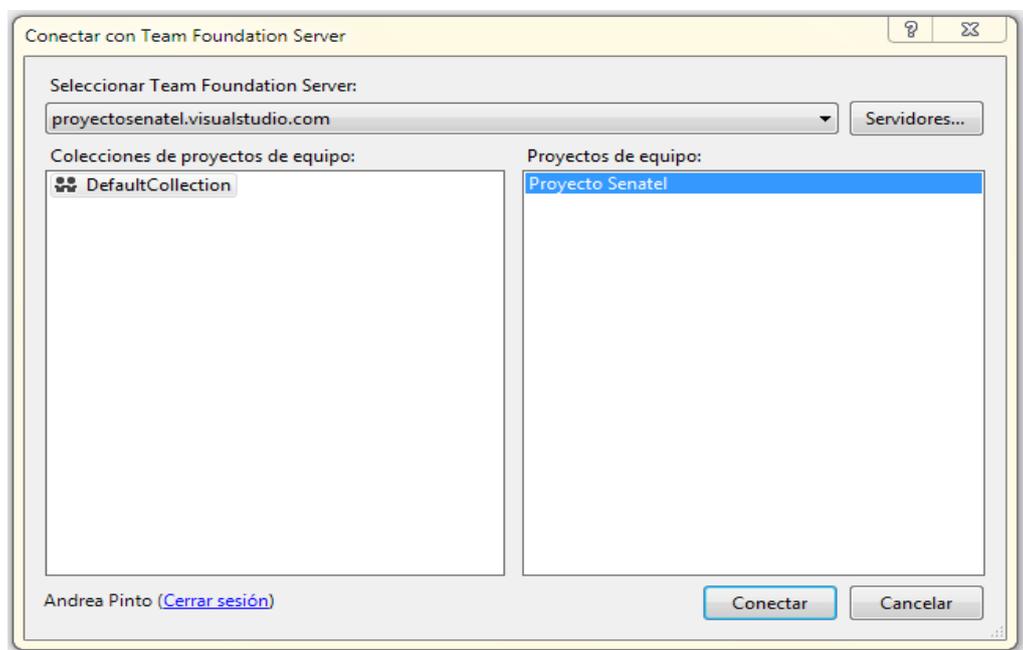


Figura 3-7 Conexión Visual Studio a TFS

3.3.COMONENTES DEL SISTEMA DISTRIBUIDO

Para comprender como está compuesto un sistema, su arquitectura y las relaciones de los elementos que lo componen, es necesario conocer los procesos que realizará y los usuarios que atenderá.

Describir un sistema a través de un diagrama de casos de uso permite conocer las necesidades del usuario, las solicitudes que se debe resolver, por ende las características de desarrollo y funcionamiento del sistema.

En la Figura 3-8 se muestra el siguiente diagrama de casos de uso que describe de manera global y sencilla el sistema, las funciones y mecanismos que debe proporcionar para cada usuario en el presente Proyecto.

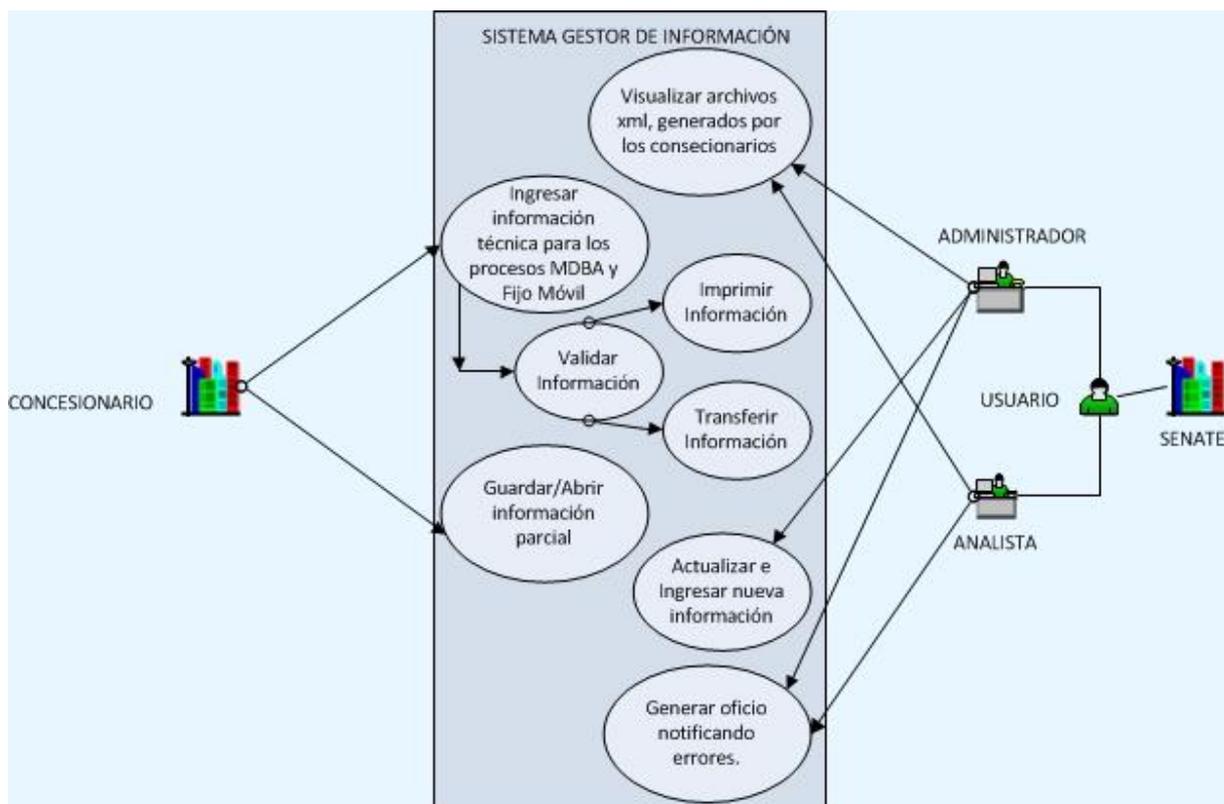


Figura 3-8 Diagrama de casos de uso

A continuación se describen los tres componentes de Sistema Distribuido: la aplicación de escritorio, el servicio Web RESTful y la aplicación web, especificando cada una de las funciones que estos realizan.

3.3.1. APLICACIÓN DE ESCRITORIO

Se planteó el desarrollo de una aplicación de escritorio ya que permite trabajar de modo offline, es decir sin la necesidad de tener conexión a Internet, además al desarrollarla en el entorno de Visual Studio 2012 utilizando WPF se pudo contar con variedad de controles y controles personalizados.

Para comenzar con el desarrollo se crearon los formularios (interfaces gráficas) tanto para Sistemas MDBA como para Servicio MT-RDV los cuales se detallan en la Tabla 3-8.

SERVICIO	FORMULARIO
MDBA/ MT-RDV	<ul style="list-style-type: none"> • Formulario de selección de servicio
MDBA	<ul style="list-style-type: none"> • Formulario principal • Formulario RC-1B: Información Legal • Formulario RC-2A: Infraestructura del Sistema de radiocomunicaciones • Formulario RC-3A: Información de antenas • Formulario RC-4A: Información de equipamiento • Formulario RC-9A: Enlaces punto-punto • Formulario RC-9B: Sistemas punto-multipunto • Formulario RC-15A: Estudio técnico de Emisiones RNI • Formulario Acerca de
MT-RDV	<ul style="list-style-type: none"> • Formulario principal • Formulario RC-1A: Información Legal • Formulario RC-2A: Infraestructura del Sistema de radiocomunicaciones • Formulario RC-3A: Información de antenas • Formulario RC-4A: Información de equipamiento • Formulario RC-5A: Tipo de operación y número de estaciones para Servicio MT-RDV • Formulario RC-15A: Estudio técnico de Emisiones RNI • Formulario Acerca de

Tabla 3-8 Formularios de la Aplicación de Escritorio

Las funcionalidades de la aplicación de escritorio son:

- Permitir ser descargable de la página web de la SENATEL; para el caso del prototipo una página web de prueba.
- Ingresar la información de forma dinámica y amigable.
- Permitir el ingreso de información *offline*.
- Realizar validaciones locales *offline*.
- Realizar cálculos matemáticos para determinar valores como el azimut, distancia entre estructuras, PIRE, etc.
- Conectarse al servicio Web para realizar validaciones *online*.
- Generar automáticamente el Formulario RC-15A.
- Guardar la información ingresada en cada formulario en formato binario
- Imprimir el formulario en formato PDF.

- Generar el archivo XML que posteriormente será importado al software SPECTRAplus.
- Cargar nuevamente un trámite guardado en formato binario en caso que el concesionario requiera de alguna modificación o eliminación.

A continuación se detallan los procesos para las validaciones, controles personalizados, generación de PDF y creación de XML.

3.3.1.1. Validaciones

Para las validaciones locales *offline* de los campos en los que el concesionario ingresará la información, se hizo uso de controles como: *combobox* para seleccionar datos establecidos y *textbox* para el ingreso de información. Para los *textbox* se usó el evento `PreviewTextInput` en el cual se realizarán las validaciones restringiendo al usuario el ingreso de números, letras o símbolos.

En la Figura 3-9 se puede apreciar una validación del Formulario RC-1B, en el *textbox* del RUC (Registro Único de Contribuyentes) nombrado `txtRuc`.

Se llama a la función `EvaluarNumeros` en el evento `PreviewTextInput`, la función realiza la validación restringiendo el ingreso de letras y símbolos, por ejemplo en el caso del RUC se requiere del uso exclusivo de números, además se controla la longitud máxima de caracteres ya que un RUC consta de trece números.

```
private void EvaluarNumeros(object sender, TextCompositionEventArgs e)
{
    int ascci = Convert.ToInt32(Convert.ToChar(e.Text));
    if (ascci >= 48 && ascci <= 57)
    {
        e.Handled = false;
    }
    else
        e.Handled = true;

    txtRuc.ToolTip = "Sólo ingrese números Ejm. 1718011578001";
    txtRuc.MaxLength = 13;
}
```

Figura 3-9 Validación del control *textbox*

Para las validaciones *online* la aplicación se deberá conectar al servicio Web lo cual se explica más adelante.

Para realizar las validaciones locales se consultaron las referencias: [24] y [25].

Todo el código referente a las validaciones se encuentra disponible en el Anexo K adjunto a este documento.

3.3.1.2. Cálculos Matemáticos

La aplicación de escritorio realiza los siguientes cálculos basándose en el Instructivo Formularios de Concesión de Frecuencias [21] y en la RESOLUCIÓN NO. 417-15-CONATEL-2005 [6].

Formulario 4A

- Azimut
- Ángulo de elevación

Formulario RC-5A para Servicio MT-.RDV, RC-9A y RC-9B para Sistemas MDBA

- Validación de la Potencia máxima
- Cálculo de la distancia entre las estructuras

Formulario RC-15A RNI

- Cálculo de R^2 , donde $R = \sqrt{x^2 + (h - d)^2}$, siendo x una constante, h la altura y d la distancia
- Cálculo del PIRE
- Cálculo del *Slim* teórico

En el Anexo C se presentan las funciones realizadas para efectuar los cálculos matemáticos de las distancias, azimut y ángulo de elevación.

Particularmente para el cálculo del azimut se consultaron las referencias: [26] y [27].

3.3.1.3. Controles Personalizados

Se crearon controles personalizados debido a la necesidad de crear y duplicar un conjunto de controles (*textbox*, *combobox*, *label*, etc.) en tiempo de ejecución.

De esta manera, tanto para Sistemas MDBA como para Servicio Móvil Terrestre se crearon los controles que se indica en la Figura 3-10.

Todos los formularios contienen un conjunto de controles personalizados con excepción de los Formularios: RC-1A y RC-1B que son los concernientes a información legal y el Formulario RC-15A RNI que se genera automáticamente con los datos de los demás formularios.

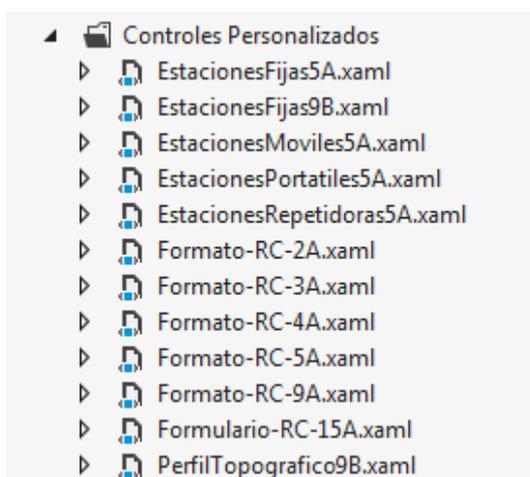


Figura 3-10 Controles personalizados

Para la creación de un control personalizado se debe agregar un nuevo elemento al proyecto y elegir `Control de Usuario WPF`, de esta manera se presentará una interfaz a la que se deberá agregar todos los controles que se necesiten y validaciones en el caso que se requieran.

Conjuntamente se puede crear o editar el código XAML para crear un control de usuario personalizado, todo depende de lo que se requiera. En la Figura 3-11 se indica el código XAML utilizado para la creación del control personalizado que forma parte del Formulario RC-2A. Todo el código de los controles personalizados se encuentra disponible en el Anexo K.

Finalmente se debe agregar el nuevo control al `Cuadro de Herramientas` del proyecto actual y utilizarlo las veces que sean necesarias.

```

UserControl x:Class="App_Escritorio.Formato2A"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
mc:Ignorable="d" Height="478.5" Width="807" Loaded="UserControlLoaded1"
<Grid Margin="0,0,0,0">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="141*"/>
    <ColumnDefinition Width="217*"/>
    <ColumnDefinition Width="446*"/>
  </Grid.ColumnDefinitions>
  <Canvas HorizontalAlignment="Left" Height="478" VerticalAlignment="Top" Width="806" Background="{DynamicResource {x:Static SystemColors.MenuB
  <Label Content=" Estructura del Sistema de Radiocomunicaciones" Height="30" Width="808" FontWeight="Bold" Background="{DynamicResource
  <Label Content=" Estructura" Canvas.Top="30" Height="25" Width="806" Background="{DynamicResource {x:Static SystemColors.ControllightB
  <Label Content="Tipo de Estructura" Canvas.Top="66" RenderTransformOrigin="0.289,-0.192" Width="115" Canvas.Left="331"/>
  <Label Content="Altura de la Estructura s.n.m. (m)" Canvas.Top="118" RenderTransformOrigin="0.289,-0.192" Width="197" Canvas.Left="10"/>
  <TextBox x:Name="txtAlturaEstructura" Height="22" Canvas.Left="232" TextWrapping="Wrap" Canvas.Top="123" Width="77" PreviewTextInput="Eva
  <Label Content="Altura de la Estructura (BASE CIMA) (m)" Canvas.Top="92" RenderTransformOrigin="0.289,-0.192" Width="222" Canvas.Left="16
  <TextBox x:Name="txtAlturaBaseCima" Height="22" Canvas.Left="232" TextWrapping="Wrap" Canvas.Top="95" Width="77" RenderTransformOrigin="0
  <Label Content="Fuente de Energía" Canvas.Top="90" Width="113" Canvas.Left="333"/>
  <Label Content="Tipo de Respaldo" Canvas.Top="117" Width="113" Canvas.Left="333" RenderTransformOrigin="0.575,0.115"/>
  <ComboBox x:Name="cmbTipoEstructura" Height="21" Width="150" Canvas.Left="451" Canvas.Top="69" FontSize="12" IsEditable="True" TabIndex=
  <ComboBox.Background>
    <LinearGradientBrush EndPoint="0,1" StartPoint="0,0">
      <GradientStop Color="#FFF3F3F3" Offset="0"/>
      <GradientStop Color="#FFEBE8E8" Offset="0.5"/>
      <GradientStop Color="#FFDDDDDD" Offset="0.5"/>
      <GradientStop Color="#FF8F8888" Offset="1"/>
    </LinearGradientBrush>
  </ComboBox.Background>
  <ListBoxItem x:Name="lstbxTorreAutosoportada" Content="Torre Autosoportada" />
  <ListBoxItem x:Name="lstbxItmTorreNoAutosoportada" Content="Torre no Autosoportada"/>
  <ListBoxItem x:Name="lstbxItmMastilEmpotrado" Content="Mástil"/>
  <ListBoxItem x:Name="lstbxItmEmpotramiento" Content="Empotramiento en una Edificación"/>
  </ComboBox>
  <ComboBox x:Name="cmbFuenteEnergia" Height="21" Width="150" Canvas.Left="451" Canvas.Top="94" FontSize="12" IsEditable="True" TabIndex=
  <ComboBox.Background>
    <LinearGradientBrush EndPoint="0,1" StartPoint="0,0">
      <GradientStop Color="#FFF3F3F3" Offset="0"/>

```

Figura 3-11 Código XAML del control personalizado que forma parte del Formulario RC-2A

En la Figura 3-12 se muestra como ejemplo el control personalizado que forma parte del Formulario RC-2A.

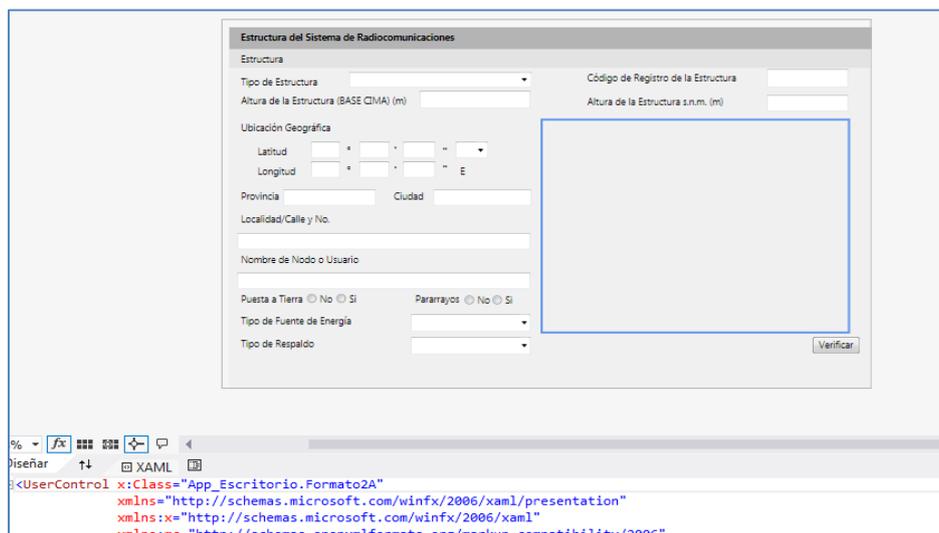


Figura 3-12 Control Personalizado para el Formulario RC-2A

Para realizar los controles personalizados se consultaron las referencias: [28] y [29]. Todo el código referente a la creación de controles personalizados se encuentra disponible en el Anexo K.

3.3.1.4. PDF

Para imprimir los formularos en formato PDF establecido por la SENATEL, se hizo uso de la librería *iText*, la cual permite crear, editar, manipular y generar archivos en formatos PDF, RTF⁴⁰ y HTML en varios lenguajes de programación. Se puede descargar la versión de la librería para C# desde la referencia [30].

Con *iText Sharp* y el conjunto de clases que dispone se puede crear documentos PDF a partir de cualquier origen de datos, en este caso a partir de los campos ingresados en los formularios, es necesario aludir que el documento en PDF debe ser un PDF editable para que los campos puedan ser ingresados. En la Figura 3-13 se muestra un fragmento de código que muestra el uso de *iText Sharp* en el presente Proyecto.

```

PdFs.cs*  X  Formato-RC-2A.xaml.cs
App_Escritorio.PdFs
using System.IO;
using Microsoft.Win32;
using iTextSharp.Text;
using iTextSharp.text.pdf;
using System.Windows;

namespace App_Escritorio
{
    internal class PdFs
    {
        public void PdfFormB()
        {
            if (Persistencia.ListaB.Count != 0)
            {
                const string pdfTemplate = @"..\..\Templates\Forma_Legal_IB.pdf";
                var dlg = new SaveFileDialog
                {
                    FileName = "Forma_Legal_IB", DefaultExt = ".pdf", Filter = "Text documents (*.pdf)*.pdf";
                };
                // Muestra el save file dialog
                var result = dlg.ShowDialog();
                // Guardar el archivo
                if (result != true) return;
                string newFile = dlg.FileName;
                var pdfReader = new PdfReader(pdfTemplate); //se crea una instancia del lector de PDF
                var pdfStamper = new PdfStamper(pdfReader, new FileStream(newFile, FileMode.Create));
                AcroFields pdfFormFields = pdfStamper.AcroFields;
                BaseFont bfDescBold = BaseFont.CreateFont(BaseFont.HELVETICA, BaseFont.WINANSI, BaseFont.CACHED);
                //winansi para las Ñ, y letra con apostrofe o acentos
                pdfFormFields.AddSubstitutionFont(bfDescBold);
                string valor = string.Empty;
                Clase_B cargarDatos = Persistencia.ListaB[0];

                var fecha = DateTime.Today.ToString("D"); //se puede cambiar de formato con "d" "g"
                string[] nombresR1 = cargarDatos.NombreProfesional.Split(' ');
                pdfFormFields.SetField("OBJETO DE LA SOLICITUD", valor);
                pdfFormFields.SetField("RUC", cargarDatos.Ruc);
                pdfFormFields.SetField("NOMBRE DE LA EMPRESA", cargarDatos.NombreEmpresa);
                pdfFormFields.SetField("ACTIVIDAD DE LA EMPRESA", cargarDatos.ActividadEmpresa);
                pdfFormFields.SetField("PROVINCIA", cargarDatos.ProvinciaEmpresa);
                pdfFormFields.SetField("CIUDAD", cargarDatos.CiudadEmpresa);
                pdfFormFields.SetField("DIRECCION", cargarDatos.DireccionEmpresa);
                pdfFormFields.SetField("TELEFONO FAX", cargarDatos.TelefonoEmpresa);
                pdfFormFields.SetField("NOMBRES", cargarDatos.NombresRepresentanteLegal);
                pdfFormFields.SetField("APELLIDO MATERNO", cargarDatos.ApellidoMaternoRepresentanteLegal);
                pdfFormFields.SetField("APELLIDO PATERNO", cargarDatos.ApellidoPaternoRepresentanteLegal);
                pdfFormFields.SetField("CI", cargarDatos.CedulaRepresentanteLegal);
                pdfFormFields.SetField("NOMBRES",
                    cargarDatos.NombresRepresentanteLegal + " " +
                    cargarDatos.ApellidoPaternoRepresentanteLegal + " " +
                    cargarDatos.ApellidoMaternoRepresentanteLegal);
            }
        }
    }
}

```

Figura 3-13 Fragmento de código que muestra el uso de *iText Sharp*

Para imprimir los formularos en formato PDF se consultaron las referencias: [31] y [32]. Todo el código referente a la creación de los formularios en formato PDF establecido por la SENATEL se encuentra disponible en el Anexo K.

⁴⁰ **RTF (Rich Text Format)** es un formato de archivo informático para el intercambio de documentos, describe un tipo de archivo que puede manejar el formato de texto básico.

3.3.1.5. XML

Para generar el archivo en formato XML a ser importado por el software SPECTRAplus, se hizo uso de la librería `System.Xml`, la cual permite la creación del archivo XML. Ubicando los campos necesarios con sus respectivos valores en las posiciones adecuadas se obtiene el archivo XML con la estructura correcta. En la Figura 3-14 se muestra un fragmento de código en el cual se puede visualizar el uso del objeto instanciado de la clase `XmlTextWriter`.

```

XmlTextWriter bookWriter = null;
m_strFileName = path;
bookWriter = new XmlTextWriter(m_strFileName, null);
try
{
    bookWriter.Formatting = Formatting.Indented;
    bookWriter.Indentation = 6;
    bookWriter.Namespaces = false;
    bookWriter.WriteStartDocument();
    bookWriter.WriteStartElement("", "SPECTRAPLUS", "");

    //Address, datos correspondientes al fabricante de Equipo
    bookWriter.WriteStartElement("", "ADDRESS", "");
    bookWriter.WriteStartElement("", "AD_ID", "");
    bookWriter.WriteString("222");
    bookWriter.WriteEndElement();
    bookWriter.WriteStartElement("", "AD_CRDATE", "");
    bookWriter.WriteString("2010-04-06T11:48:38-05:00");
    bookWriter.WriteEndElement();
    bookWriter.WriteStartElement("", "AD_LWRDATE", "");
    bookWriter.WriteString("2012-07-27T11:11:56-05:00");
    bookWriter.WriteEndElement();
}

```

Figura 3-14 Ejemplo de uso de `XmlTextWriter`

Como se mencionó en el Capítulo II, las etiquetas del esquema XML tienen identificadores que permiten la correspondencia o relación entre los datos del esquema.

Para que la aplicación de escritorio genere el archivo XML con los respectivos identificadores se tuvo que seguir la relación que se indica en los diagramas tanto para Sistemas MDBA (Ver Figura 3-15) como para Servicio MT-RDV (Ver Figura 3-16), respectivamente.

Los diagramas no corresponden a una base de datos, ni a una estructura en particular, sólo es el resultado del análisis de las etiquetas en el esquema de XML.

Para imprimir los formularios en formato XML se consultaron las referencias: [33] y [34]. Todo el código referente a la generación del XML se encuentra disponible en el Anexo K.

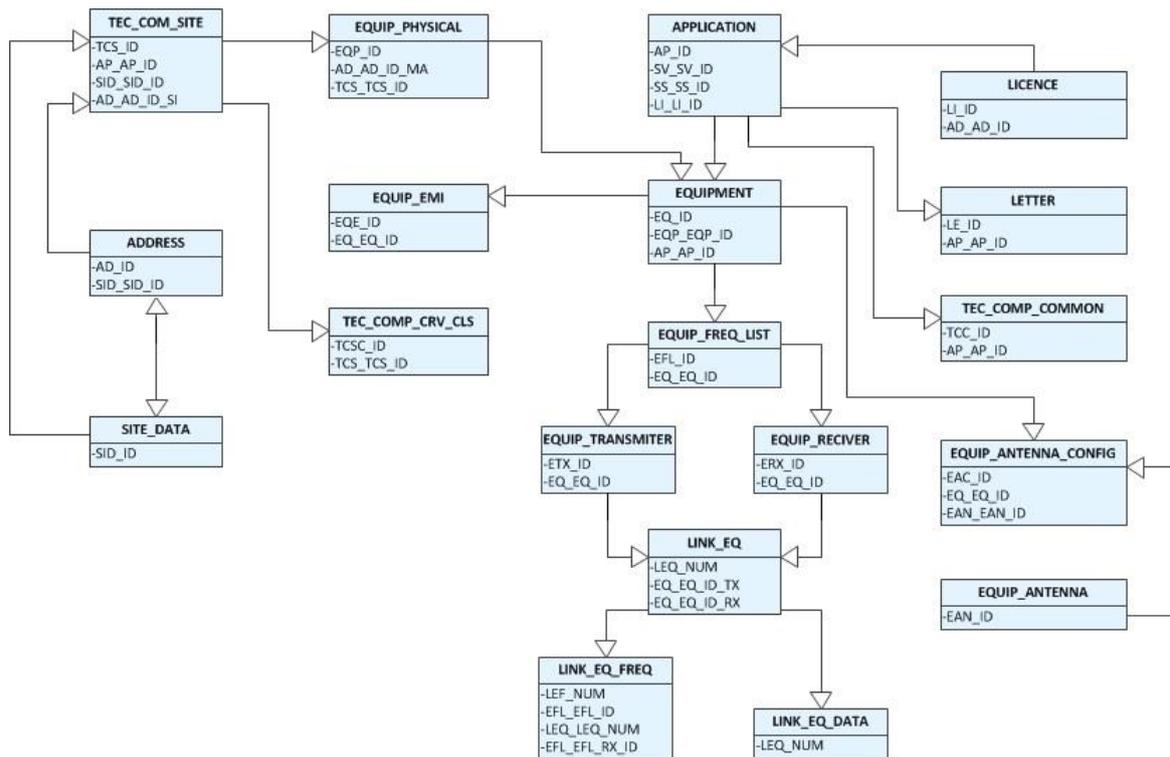


Figura 3-15 Diagrama del esquema XML para Sistemas MDBA

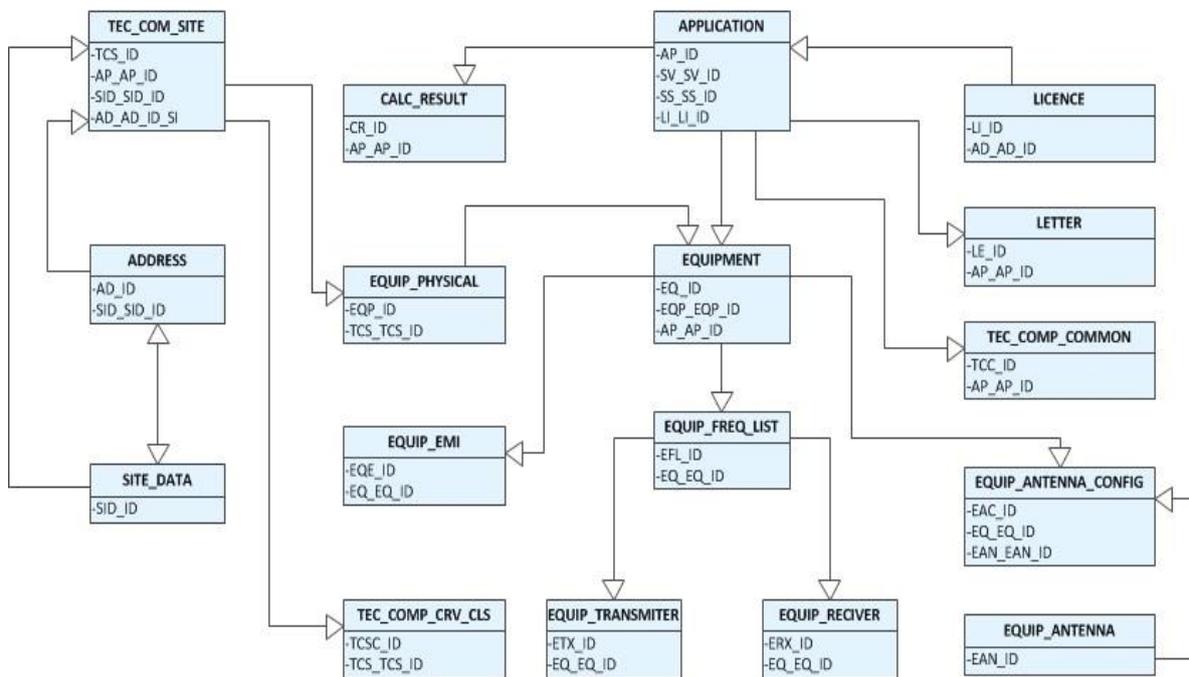


Figura 3-16 Diagrama del esquema para Servicio MT-RDV

3.3.2. SERVICIO WEB REST

Se planteó el desarrollo de servicios Web en WCF, porque está orientado a la creación y diseño de servicios Web de manera sencilla y concisa. Un servicio Web WCF es un componente de software que no posee interfaz gráfica, pero permite la interoperabilidad o comunicación con otras aplicaciones, específicamente para el prototipo el Servicio Web realiza:

- Validación de información *online*; es decir el Servicio Web le permite al concesionario recibir información (RUC, antenas y equipos) de la base de datos.
- Validación de Títulos Profesionales, el servicio realiza peticiones a la página de la SENESCYT.
- Validación de Coordenadas Geográficas, el servicio realiza peticiones a servicio de Google Maps.
- Transferencia de archivos XML a un repositorio de pruebas.
- Autenticación de usuarios.

Los servicios REST exponen sus operaciones u otros recursos a través de un URI, en la mayoría de los casos es una cadena URL HTTP, de esta forma el cliente puede consumir este tipo de servicios de una forma sencilla a través de métodos estándar como GET y POST del protocolo HTTP. Esto realmente es una ventaja para el cliente ya no se encuentra atado a una determinada tecnología o lenguaje para usar el servicio.

El entorno de desarrollo de .NET, desde la versión del framework 3.5 proporciona una opción para el desarrollo de servicios WCF, la cual se ha mejorado para la versión 4.0.

La opción que se debe elegir para desarrollar un Servicio Web es nuevo proyecto WCF como lo indica la Figura 3-17.

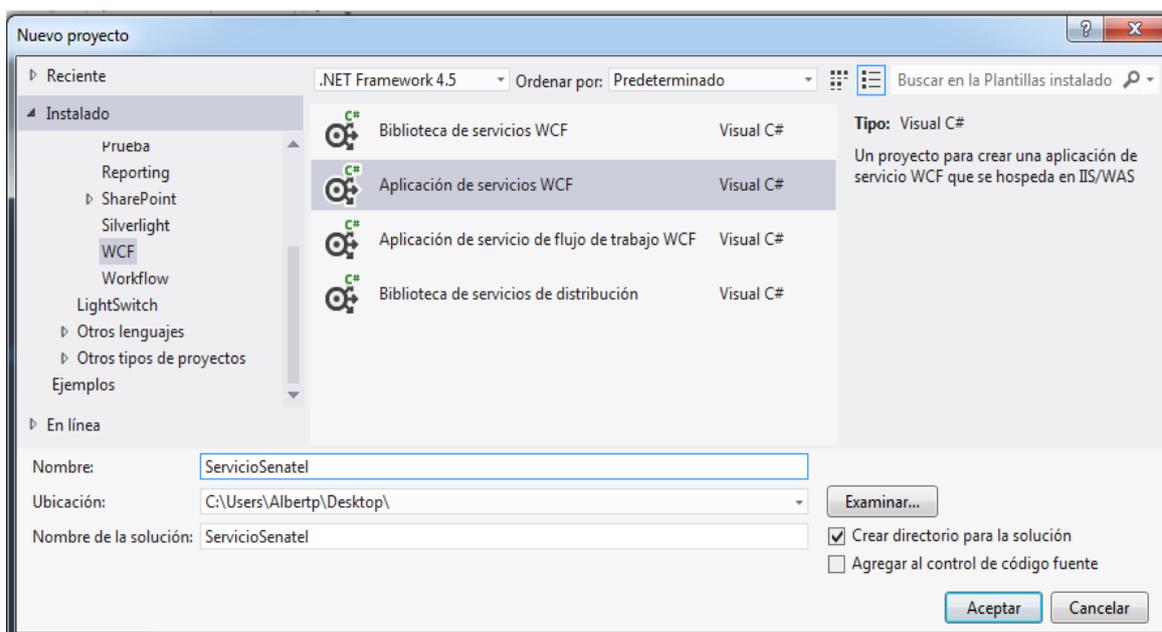


Figura 3-17 Creación de un nuevo proyecto WCF

Se creará una carpeta con elementos que se incluyen por defecto como lo indica la Figura 3-18, entre los cuales se encuentran:

- **IService.cs** es la Interfaz que permite declarar los métodos que el servicio hará públicos, es decir representa el contrato.
- **Service.svc** es la clase en la que se implementarán los métodos declarados en la interfaz.
- **Web.config** este archivo permite configurar los `endpoint` necesarios, en este archivo se pueden especificar características del servicio en formato XML, entre las que están el tipo de servicio, el protocolo que se usará, etc.

Los `endpoints` están compuestos por tres valores:

- ✓ `Address` (¿Dónde?): representa la ubicación del servicio.
- ✓ `Binding` (¿Cómo?): cómo se comunica el cliente con el servicio.
- ✓ `Contract` (¿Qué?): especifica los métodos y los tipos de datos que estos retornan.

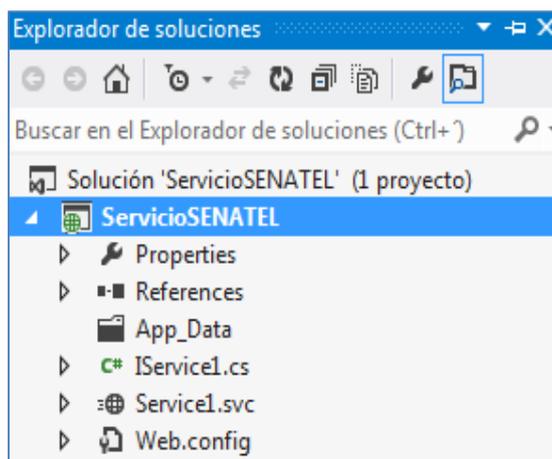


Figura 3-18 Elementos por defecto de un proyecto WCF

Particularmente para el desarrollo del presente Proyecto se trabajó con los tres elementos mencionados, sobre los que se realizó los cambios necesarios que a continuación se describen. Cabe mencionar que se consultaron las siguientes referencias: [35], [36] y [37].

3.3.2.1. Archivo de Configuración

El archivo `web.config` está constituido por varias secciones, entre ellas el `<system.serviceModel>` que agrupa los elementos de configuración WCF. Algunos de los elementos principales de `<system.serviceModel>` son:

- `<services>`, puede contener una colección de elementos `<service>`, con especificaciones para todos los servicios de los anfitriones de la aplicación.
- `<bindings>`, puede contener una colección de elementos `<binding>` proporcionados por el sistema o personalizados por el desarrollador. Los elementos de estas colecciones describen cómo el cliente se comunica con el servicio.
- `<behaviors>`, puede contener una colección de elementos `<behavior>`, provistos por el sistema o personalizados, permiten modificar o establecer:

- ✓ El comportamiento de los servicios `serviceBehaviors` el cual afecta a los aspectos del servicio.
- ✓ El comportamiento de extremo `endpointBehaviors` que afecta solo a las propiedades relacionadas con un `endpoint`.

Permitiendo así, diferenciar el campo de acción, consiguiendo ser mucho más detallados y lograr objetivos más específicos.

Algunos de los elementos secundarios de `<system.serviceModel>` son:

- `<service>`, contiene el elemento `<endpoint>` que posee atributos que permiten a la aplicación cliente conocer dónde ubicar al servicio y qué métodos puede usar el servicio. Pueden existir varios elementos `<service>` debido a que cada uno se identifica de manera única mediante su atributo `Name` que consta del espacio de nombres seguido de un punto y el nombre del servicio, para el presente Servicio este atributo es: `name="ServicioSenatel.ServicioSenatel"`.
`<BehaviorConfiguration>`, es un atributo que posee tanto `<service>` como `<endpoint>`, mediante el nombre que se configure en la sección `<behavior>` se le puede otorgar un determinado comportamiento al servicio al extremo de conexión. Por ejemplo en `<endpoint>` se puede usar un `<behavior>` para indicar el comportamiento Rest.
- `<binding>`, permite especificar parámetros y características que están detrás de la conexión de punto final, por ejemplo permite especificar el protocolo de transporte, la codificación y demás detalles requeridos por otro servicio o cliente para comunicarse.
- `<behavior>`, como se mencionó define dos colecciones secundarias:
 - ✓ `Endpoint Behaviors`, su influencia es limitada al comportamiento de un `endpoint`, el `endpoint` puede conectar con el `<behavior>` usando el nombre de este último. Por ejemplo credenciales para autenticar a un cliente
 - ✓ `Service Behaviors`, posee mayor influencia, afecta al comportamiento del servicio. Por ejemplo establecer que `endpoint`

de los que pertenece a `<service>` pueden ser o no afectados dependiendo de su tipo *kind*.

Una vez que se ha creado el nuevo proyecto WCF y conociendo algunos aspectos del archivo de configuración, se lo debe modificar para otorgarle un comportamiento específico al servicio, en este caso particular especificar que la tecnología del servicio será REST. Describir la tecnología del servicio es importante ya que de esto depende la forma de implementar los métodos o solicitudes.

En base a la explicación anterior de algunos de los elementos que forman parte de un archivo de configuración y que se busca modificar o establecer un comportamiento para el servicio, las etiquetas a modificar o agregar en este caso son `<binding>` y `<behaviors>` las mismas que contienen los valores necesarios para determinar el comportamiento de un extremo.

En la Tabla 3-9 se muestran por ejemplo algunos subelementos de `endpoint Behaviors`.

ELEMENTO	DESCRIPCIÓN
<code><callbackTimeouts></code>	Especifica cuanto tiempo se debe esperar por una respuesta.
<code><soapProcessing></code>	Define el comportamiento del extremo de cliente para distintos tipos de mensaje.
<code><clientVia></code>	Especifica la ruta que debe seguir un mensaje.
<code><webHttp></code>	Al habilitar esta configuración en un extremo junto con <code><webHttpBinding></code> se habilita en el servicio WCF el modo web.

Tabla 3-9 Algunos subelementos de `endpoint Behaviors`

La intención es que el servicio sea de tipo REST, es decir que se pueda consumir de forma sencilla por cualquier tipo de cliente o incluso por otro servicio, únicamente mediante los métodos del protocolo HTTP, la opción a configurar es un `<endpointBehavior>` que contenga un elemento `<webHttp>`, para relacionarlo luego con el respectivo `<endpoint>`. Se puede evidenciar de forma sencilla que la configuración funciona, solamente realizando una llamada al servicio desde un navegador como lo indica la Figura 3-19.

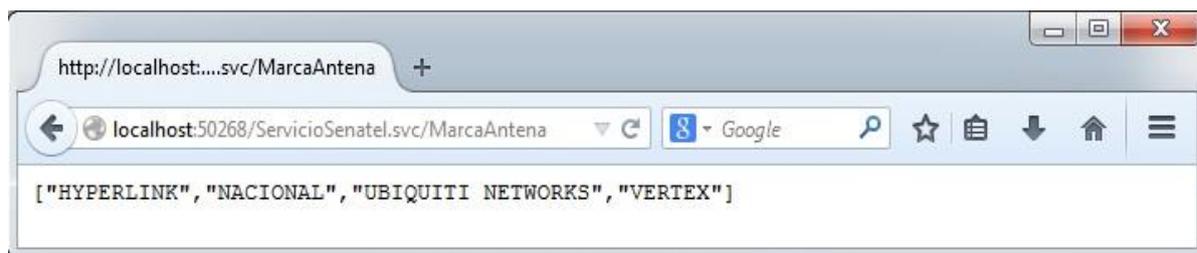


Figura 3-19 Consulta a un Servicio Web RESTful

El siguiente paso es trabajar con las clases que por defecto representan el servicio, estas clases traen código sobre el que se puede trabajar, lo recomendable es eliminar estas clases y crear nuevas con nombres acorde al proyecto que se desarrolle, en caso de decidir conservar las clases por defecto es recomendable usar la opción refactorizar que permite cambiar el nombre de los archivos en todos los elementos del proyecto donde se encuentren instanciados. (Ver Figura 3-20).

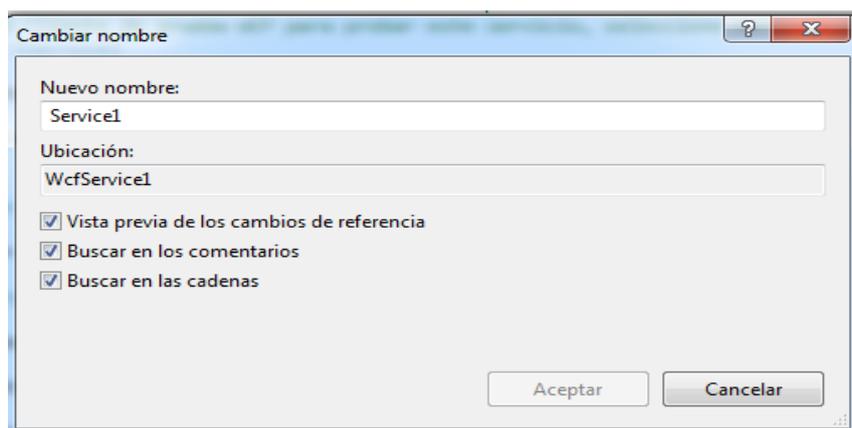


Figura 3-20 Opción para refactorizar nombres

3.3.2.1.1. *Transferencia de Archivos*

El servicio debe ser capaz de recibir los archivos XML que el concesionario genera en la aplicación de escritorio, con este fin se debe realizar modificaciones al archivo `web.config`.

Buscando modificar o establecer un comportamiento para la recepción de archivos, se debe trabajar con otro tipo de etiquetas correspondientes a la colección `<ServiceBehaviors>`.

Algunas de las opciones se muestran en la Tabla 3-10.

ELEMENTO	DESCRIPCIÓN
<dataContractSerializer>	Contiene algunos parámetros de configuración para un objeto de la clase <code>DataContractSerializer</code> , que permite serializar y de serializar una instancia de un tipo de secuencia de datos.
<serviceCredentials>	Especifica la credencial y los ajustes que se utilizarán para autenticar en el servicio las credenciales del cliente.
<serviceDiscovery>	Especifica la detectabilidad de extremos de servicio.

Tabla 3-10 Algunos atributos de un elemento `<ServiceBehaviors>`

Específicamente en este caso se modificó el atributo `MaxItemsInObjectGraph`, que determina el máximo número de objetos a serializar, pertenece al elemento `<dataContractSerializer>`.

Los atributos de los elementos `<binding>` permiten definir características de un enlace, algunos de ellos se muestran en la Tabla 3-11.

PROPIEDAD	DESCRIPCIÓN
<code>ReaderQuotas</code>	Obtiene o establece restricciones en la complejidad de mensajes SOAP que pueden ser procesados por los extremos configurados con este enlace.
<code>MaxReceivedMessageSize</code>	Obtiene o establece el tamaño máximo, en bytes, de un mensaje que puede ser procesado por el enlace.
<code>MaxBufferSize</code>	Obtiene o establece la cantidad máxima de memoria, en bytes, que se asigna para su uso por el administrador de los búferes de mensajes que reciben los mensajes del canal.
<code>TransferMode</code>	Obtiene o establece un valor que indica si el servicio configurado con el enlace utiliza <i>streaming</i> o amortiguada (o ambos) los modos de transferencia de mensajes.

Tabla 3-11 Algunas propiedades de un elemento `webHttpBinding`

En particular para el presente Proyecto se modificarán los valores `TransferMode`, `MaxBufferSize` y `MaxReceivedMessageSize`, con valores acorde al tipo de información que será transportada como se muestra en la Figura 3-21.

El parámetro `ReaderQuotas` no será modificado al no tratarse de un servicio SOAP.

```
<bindings>
  <webHttpBinding>
    <binding name="RestBinding" transferMode="Streamed" maxBufferSize="65536" maxReceivedMessageSize="67108864">
      <!--<readerQuotas maxStringLength="5242880" maxArrayLength="16384" maxBytesPerRead="4096" />-->
      <security mode="None">
        <transport clientCredentialType="None"/>
      </security>
    </binding>
  </webHttpBinding>
</bindings>
```

Figura 3-21 Valores `TransferMode`, `MaxBufferSize` y `MaxReceivedMessageSize`

Haciendo uso de los atributos `name`, correspondientes de los elementos `behavior` y `binding` anteriormente configurados se deben establecer los atributos `behaviorConfiguration` y `bindingConfiguration` pertenecientes al `endpoint` que define el servicio (Ver Figura 3-22).

```
<system.serviceModel>
  <services>
    <service name="MdbaService.ServiceMdba" behaviorConfiguration="default">
      <endpoint address="" behaviorConfiguration="web" binding="webHttpBinding" bindingConfiguration="RestBinding"
        name="MdbaService" contract="MdbaService.IServiceMdba" />
    </service>
  </services>
  <behaviors>
    <endpointBehaviors>
      <behavior name="web">
        <webHttp />
      </behavior>
    </endpointBehaviors>
    <serviceBehaviors>
      <behavior name="default">
```

Figura 3-22 Atributos `behaviorConfiguration` y `bindingConfiguration`

Para realizar la implementación de la transferencia de archivos se consultaron las referencias bibliográficas: [38] y [39]. Todo el código de la transferencia de archivos y archivos de configuración del servicio RESTful se encuentra disponible en el Anexo K.

3.3.2.2. Interfaz del Servicio Web

El nombre de la interfaz por defecto viene precedida de la letra I mayúscula, la importancia de esta interfaz radica en la definición del contrato, es decir en los métodos que el servicio tendrá disponibles para el usuario.

El espacio de nombres de mayor prioridad es `System.ServiceModel`, el cual a su vez proporciona un conjunto de clases para determinar un modelo de servicio, las cuales se necesitan para especificar el contrato de servicio y las operaciones WCF (Ver Tabla 3-12).

CLASE	DESCRIPCIÓN	SINTAXIS
<code>OperationContractAttribute</code>	Se antepone a un método que define una operación que forma parte de un contrato de servicio en una aplicación de WCF.	<code>[ServiceContract]</code>
<code>ServiceContractAttribute</code>	Se antepone a la clase que define un contrato de servicio en una aplicación de WCF.	<code>[OperationContract]</code>
<code>DataContractFormatAttribute</code>	Específica a la infraestructura de WCF un tipo de dato particular definido por el usuario para utilizar <code>DataContractSerializer</code> .	<code>[DataContract]</code> <code>[DataMember]</code>

Tabla 3-12 Espacios de nombres para definir un contrato de servicio

La interfaz del proyecto deberá llevar entre el espacio de nombres y el nombre de la interfaz el código `[ServiceContract]` para marcarla como al contrato y dentro de esta se podrá definir todos los métodos que sean necesarios, para satisfacer las necesidades de los usuarios, cada uno deberá estar antecedido de `[OperaciónContract]`.

En la Figura 3-23 se muestra la sintaxis usada para descubrir un contrato.

Los métodos que se definen en la interfaz del servicio, pueden trabajar con los tipos primitivos de datos ya conocidos ya sea como parámetro de entrada o como valor a retornar, además de ello pueden usar tipos de datos particulares que defina el desarrollador utilizando `[DataContract]`.

```

[OperationContract]
[WebGet(UriTemplate = "/DatosModeloEquipo/{marca}/{modelo}", ResponseFormat = WebMessageFormat.Json)]
List<String> DatosModeloEquipo(string marca, string modelo);

[OperationContract]
[WebGet(UriTemplate = "/ObtenerRuc/{ruc}", ResponseFormat = WebMessageFormat.Json)]
List<String> ObtenerRuc(string ruc);

[OperationContract]
[WebGet(UriTemplate = "/ObtenerProfesional/{cedula}", ResponseFormat = WebMessageFormat.Json)]
List<String> ObtenerProfesional(string cedula);

[OperationContract]
[WebGet(UriTemplate = "/ObtenerLugar/{coordenadas}", ResponseFormat = WebMessageFormat.Json)]
String ObtenerLugar(string coordenadas);

[OperationContract]
[WebGet(UriTemplate = "/ObtenerCoordenadas/{coordenadas}", ResponseFormat = WebMessageFormat.Json)]
Stream ObtenerCoordenadas(string coordenadas);

[OperationContract]
[WebGet(UriTemplate = "/ValidarUsrInt/{usuario}/{clave}", ResponseFormat = WebMessageFormat.Json)]
String ValidarUsrInt(string usuario, string clave);

```

Figura 3-23 Fragmento de código que describe un contrato

Un contrato de datos es una colección compleja de tipos de datos primitivos que permiten un acuerdo formal entre un servicio y un cliente, abstractamente se describe todos los datos que involucra un determinado proceso del servicio. Para la comunicación, el cliente y el servicio no necesitan compartir los mismos tipos, solo el mismo contrato. Al anteponer `[DataMember]` a un tipo de dato primitivo pasa a formar parte del contrato de datos.

En la Figura 3-24 se indica el contrato de datos referente a la información que se requiere del concesionario.

```

[DataContract]
public class EmpresaData
{
    [DataMember]
    public string AD_NAME;
    [DataMember]
    public string AD_FIRST_NAME;
    [DataMember]
    public string AD_COMPANY;
    [DataMember]
    public string AD_STREET;
    [DataMember]
    public string AD_CITY;
    [DataMember]
    public string AD_COUNTY;
    [DataMember]
    public string AD_PHONE;
    [DataMember]
    public string AD_E_MAIL;
    [DataMember]
    public string AD_TAX_ADMIN_NUM;
}

```

Figura 3-24 Contrato de datos del Concesionario

Definir el comportamiento de los métodos que forman parte del contrato de servicio es importante, pues influye en cómo serán consumidos por el usuario, en gran parte depende de la función que desempeñaran.

El *namespace* `System.ServiceModel.Web` proporciona clases para determinar el comportamiento de métodos pertenecientes a servicios RESTful, como también a otros servicios.

En la Tabla 3-13 se indica una breve descripción de tres de estas clases.

CLASE	DESCRIPCIÓN	SINTAXIS
<code>WebGetAttribute</code>	Este atributo permite especificar que se trata de una operación de obtención de información convocada por el WCF REST.	<code>[WebGet]</code>
<code>WebInvokeAttribute</code>	Especifica una operación de invocación y que se puede solicitar por el modelo de programación web de WCF.	<code>[WebInvoke]</code>
<code>WebServiceHost</code>	Es una clase derivada que se complementa con el de WCF tipo REST.	

Tabla 3-13 Algunas clases del *Namespace* `System.ServiceModel.Web`

Para `[WebGet]` y `[WebInvoke]` se definen propiedades similares que permiten especificar parámetros para el comportamiento de este tipo de métodos, salvo la excepción de la propiedad `Method` con que cuenta `WebInvoke` mas no así `WebGet` (VerTabla 3-14)

La propiedad `UriTemplate` perteneciente a las clases representadas por `WebGet` y `WebInvoke` permite asociar la información contenida en los parámetros incluidos en las solicitudes del protocolo HTTP, con los nombres de los parámetros del `ServiceContract`. Es decir permite asociar la información incluida en las solicitudes del cliente con los respectivos parámetros de entrada que necesita un determinado método del `ServiceContract` para responder la solicitud (Ver Figura 3-25)

PROPIEDAD	DESCRIPCIÓN
RequestFormat	Mediante esta propiedad se puede especificar el formato de las solicitudes hechas a una operación de servicio. Las opciones disponibles son XML y JSON ⁴¹ .
ResponseFormat	El formato de las respuestas enviadas desde una operación de servicio hacia un cliente es especificado por esta propiedad. Las opciones disponibles son XML y JSON.
Metodo	Permite especificar operación de un determinado protocolo que llevará a cabo, como por ejemplo el método POST del protocolo HTTP.
UriTemplate	Permite establecer y obtener el URI al que responderá el método del servicio.

Tabla 3-14 Propiedades de WebGet y WebInvoke

```
[OperationContract]
[WebGet(UriTemplate = "/MarcaEquipo", ResponseFormat = WebMessageFormat.Json)]
List<String> MarcaEquipo();

[OperationContract]
[WebGet(UriTemplate = "/ModeloAntena/{marca}", ResponseFormat = WebMessageFormat.Json)]
List<String> ModeloAntena(string marca);
```

Figura 3-25 Métodos y su comportamiento

La mayor parte de los métodos del `ServiceContract` pertenecientes a este proyecto son `WebGet`, ya que gran parte son operaciones sólo de lectura.

La respuesta a una petición `WebGet` puede requerir por parte del servicio más de un dato para realizar una operación o comparación, los que deben ser proporcionados por el usuario, en este tipo de circunstancias son útiles las formas de obtener y especificar los parámetros necesarios que se muestran en la Tabla 3-15.

⁴¹ **JSON**, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. Usado como alternativa a XML.

URI	METODO	DESCRIPCIÓN
/MarcaAntena	GET	No requiere de parámetros y devuelve todo el resultado de la operación.
/ModeloAntena/{marca}	GET, PUT	Requiere un parámetro de comparación para devolver un mejor resultado.
/DatosModeloAntena/{marca}/{modelo}	GET, PUT	Requiere más de un parámetro para reducir el tiempo de ejecución y aumentar la eficiencia.
/DatosModeloAntena/{marca}/{modelo}/{ganancia}	GET, POST, PUT, DELETE	Requiere aún más parámetros debido a que son operaciones precisas.

Tabla 3-15 Parámetros URI

En cuanto al formato de la respuesta se refiere, en el presente Proyecto se usa el formato JSON para determinadas solicitudes (RUC, antenas y equipos), su elección no sigue reglas inflexibles, sino más bien está sujeto a las necesidades del usuario y al volumen de información que se maneje, por ejemplo al tratarse de una operación que arroja un resultado muy puntual o con pocos datos, la opción más adecuada sería JSON. [40]

En la Figura 3-26 se muestra la respuesta a la solicitud `ObtenerRUC` en formato JSON, claramente se puede visualizar la diferencia en la Figura 3-27 en la cual se muestra la misma respuesta a la solicitud pero en formato XML. JSON maneja un formato de intercambio de datos ligero, retornando información legible y completamente independiente del lenguaje.

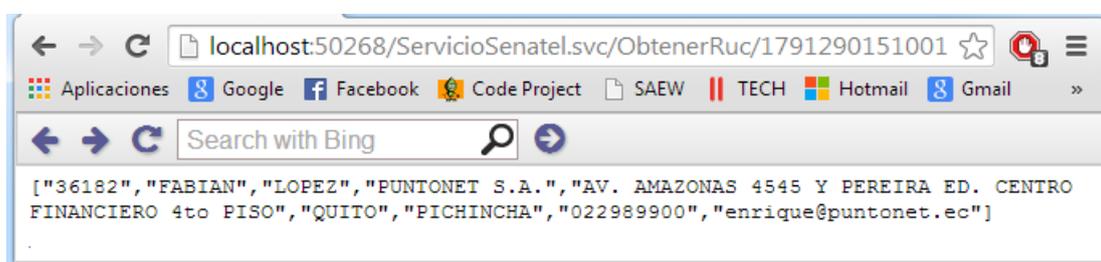


Figura 3-26 Solicitud de datos del concesionario respuesta formato JSON

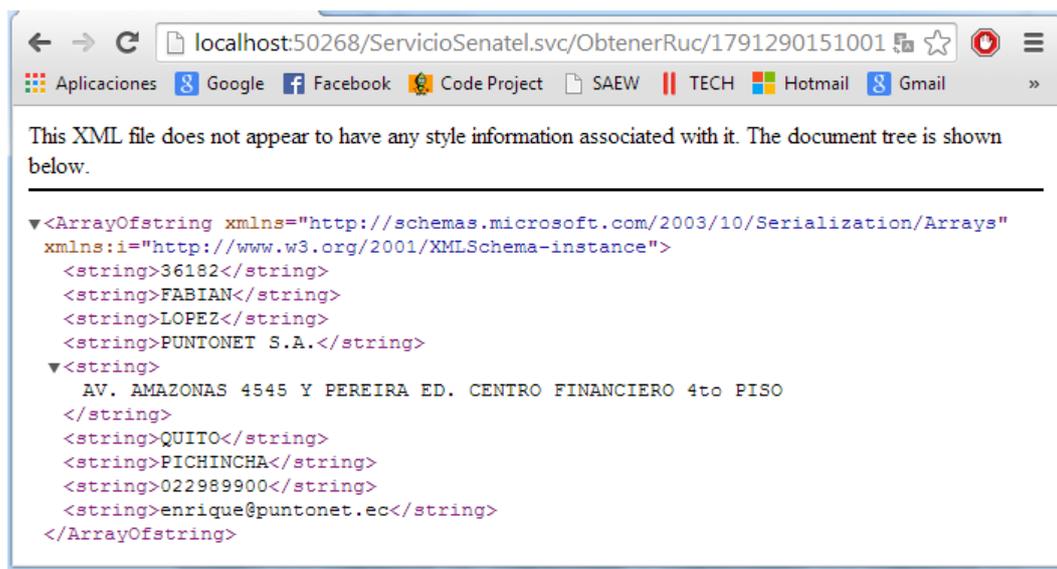


Figura 3-27 Solicitud de datos del concesionario respuesta formato XML

3.3.2.3. Implementación de la Interfaz de Servicio

La clase que implementa los métodos especificados en la interfaz de servicio, hereda las definiciones de los métodos y debe implementarlos respetando los tipos de datos, tanto para el retorno como para los parámetros de entrada.

En este punto se puede hacer uso de todas las herramientas disponibles para brindar una solución a las necesidades del usuario, así, puede implementarse consultas a bases de datos, programación orientada a objetos o inclusive realizar consultas a otros servicios, algunas de estas alternativas fueron implementadas y se las describe a continuación.

3.3.2.3.1. Consulta a Base de Datos

El software SPECTRAplus califica los procesos para el Otorgamiento de Títulos Habilitantes, Registros de MDBA, entre otros, apoyándose en información técnica, parte de la cual mantiene en registros de su base de datos.

Para el desarrollo del prototipo la SENATEL proporcionó los nombres de algunas tablas y campos a los cuales se van a realizar las consultas (Ver Figura 3-28).

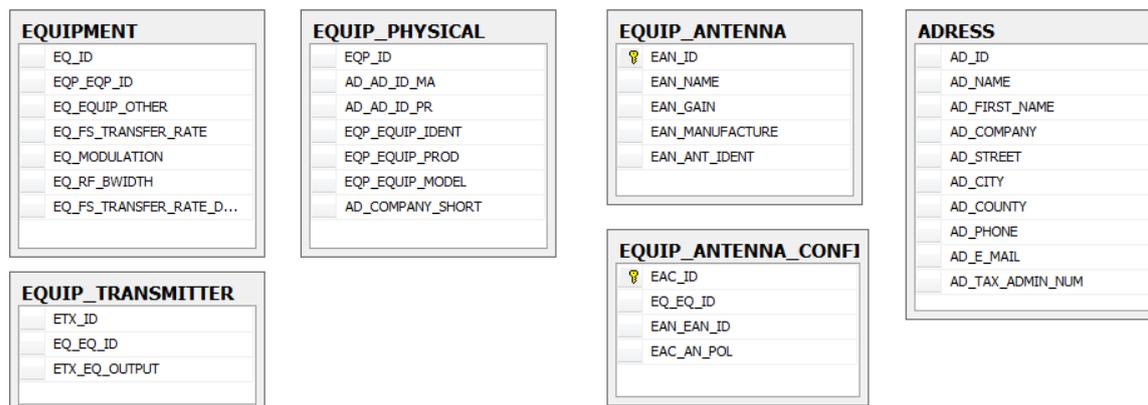


Figura 3-28 Algunas de las tablas que forman parte de la Base de Datos de SPECTRAplus

La información proporcionada no corresponde a un tipo de base de datos relacional ya que las tablas no tienen relación de dependencia entre sí.

Un sistema de clases permite al servicio conectarse y manipular los datos de las tablas como si fuera una base de datos, permitiendo realizar las consultas necesarias para responder a las solicitudes de los usuarios. (Ver Figura 3-29).

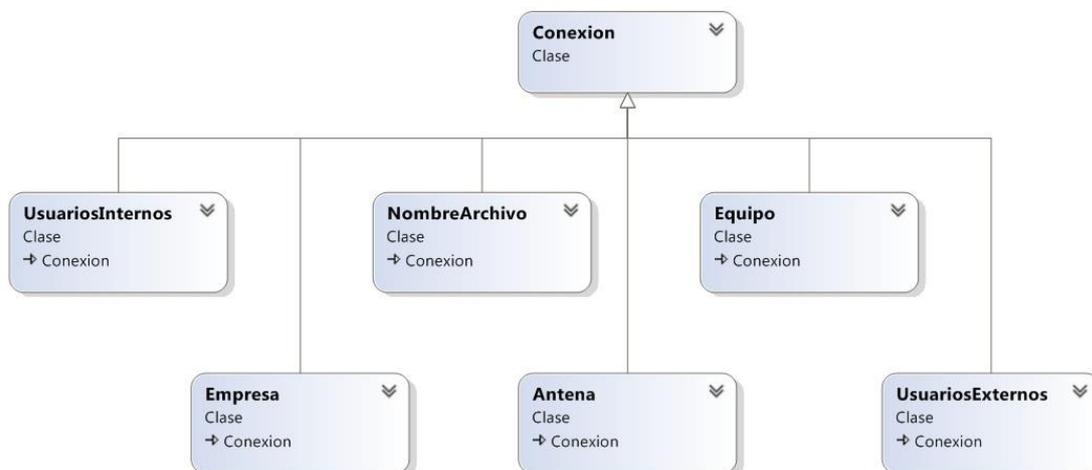


Figura 3-29 Diagrama de clases para manipular la Base de Datos

La clase **Conexion** como lo indica su nombre permite la conexión con la base de datos, acceder, modificar o eliminar los datos, contiene una variable tipo **String** con la cadena de conexión a la base SQL y un objeto **SqlConnection**, juntos permiten establecer la conexión con la base (Ver Figura 3-31).

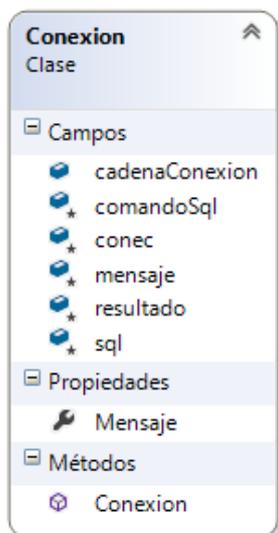


Figura 3-30 Clase `Conexion`

El beneficio de tener una clase independiente para la conexión es que se logra organizar de mejor manera el código, se evita repetir código con el mismo propósito en una misma clase (o varias clases), se realiza las consultas sin tener que volver a digitar el código de la conexión en cada consulta a realizar, solamente es necesario heredarlo de la clase `Conexion` (Ver Figura 3-31).

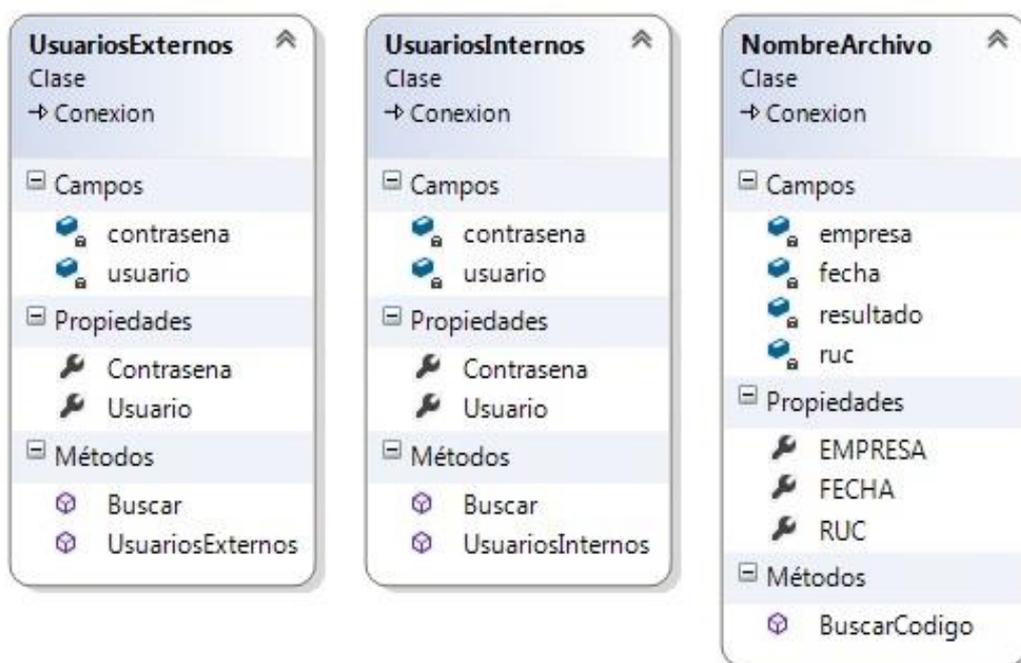


Figura 3-31 Clases que heredan de `Conexion`

Las clases `Empresa`, `Antena` y `Equipo` cumplen tareas muy similares para diferentes grupos de datos. La clase `Empresa` se ocupa de manipular los datos del concesionario e implementa los métodos necesarios para consultar información de la tabla `ADRESS`.

La clase `Antena` maneja la tabla `EQUIP_ANTENNA` que guarda información técnica de las antenas disponibles para realizar los trámites.

La clase `Equipo` trabaja con la tabla `EQUIP_PHYSYCAL` en la que se almacena la información de los equipos homologados que pueden utilizarse en los trámites.

En la Figura 3-32 se muestran estas clases con sus respectivos campos, propiedades y métodos.

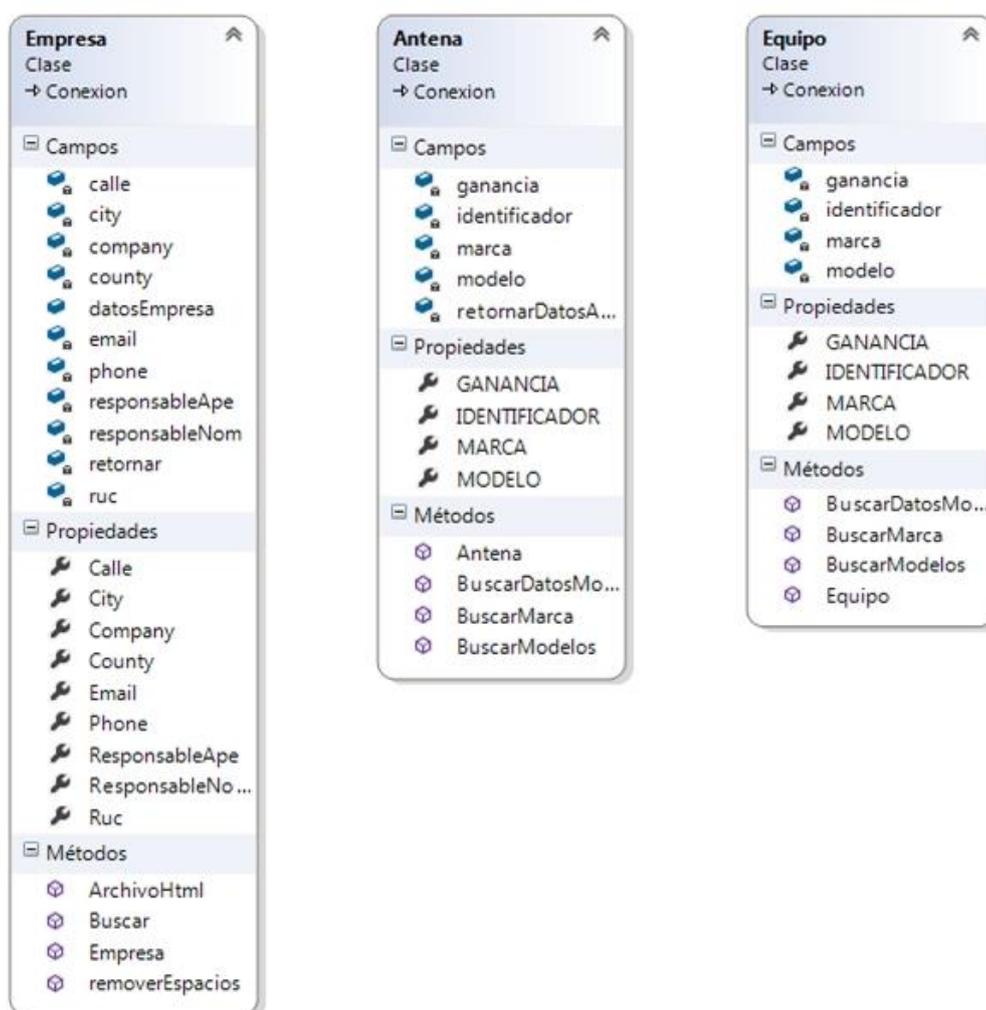


Figura 3-32 Diagramas de clases para manipular datos

En la Figura 3-33 se muestra un fragmento de código para consultar información del concesionario.

```

public bool Buscar()
{
    bool resultado = false;
    this.sql = string.Format(@"SELECT AD_ID, AD_NAME , AD_FIRST_NAME , AD_COMPANY ,
                             AD_STREET , AD_CITY , AD_COUNTY , AD_PHONE , AD_E_MAIL,
                             AD_MAN_NUMBER FROM ADDRESS WHERE AD_TAX_ADMIN_NUM ='{0}'", this.ruc);
    this.comandoSql = new SqlCommand(this.sql, this.conec);
    this.conec.Open();
    SqlDataReader Reg = null;
    Reg = this.comandoSql.ExecuteReader();
    if (Reg.Read())
    {
        resultado = true;
        datosEmpresa.Insert(0,String.Format("{0}", Reg[0]));
        datosEmpresa.Insert(1,String.Format("{0}", Reg[1]));
        datosEmpresa.Insert(2,String.Format("{0}", Reg[2]));
        datosEmpresa.Insert(3,String.Format("{0}", Reg[3]));
        datosEmpresa.Insert(4,String.Format("{0}", Reg[4]));
        datosEmpresa.Insert(5,String.Format("{0}", Reg[5]));
        datosEmpresa.Insert(6,String.Format("{0}", Reg[6]));
        datosEmpresa.Insert(7,String.Format("{0}", Reg[7]));
        datosEmpresa.Insert(8,String.Format("{0}", Reg[8]));
        datosEmpresa.Insert(9, String.Format("{0}", Reg[9]));
    }
    else
    {
        this.mensaje = "ERROR";
        resultado = false;
    }
    this.conec.Close();
    return resultado;
}

```

Figura 3-33 Fragmento de código para consultar información del concesionario

3.3.2.3.2. Consulta a SENESCYT

Debido a la necesidad de validar títulos de tercer nivel se optó por realizar la consulta a la entidad que se encarga de la administración dicha información, la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT).

La SENESCYT cuenta con un sistema que no es precisamente un servicio web pero del que se puede hacer uso para validar información del personal técnico del concesionario (Ver Figura 3-34).

Este sistema permite verificar si un profesional cuenta con un título debidamente reconocido y registrado en la SENESCYT, por medio de la cédula de una persona o su nombre.



Figura 3-34 Consulta de títulos registrados SENESCYT

Haciendo uso del software Wireshark se pudo analizar la solicitud HTTP cuando se realizan las consultas y de esta forma poder construir una solicitud idéntica, usarla con el método POST y poder descargar el contenido de la página de consulta de títulos. (Ver Figura 3-35).

Se envía un *request* con los datos del profesional técnico que se ingresan en el Formulario RC-1A o RC-1B, estos serán procesados y luego se obtiene un *response* que contiene una página en formato HTML con los datos requeridos, el contenido de esta página será procesado haciendo uso de la librería HtmlAgilityPack.

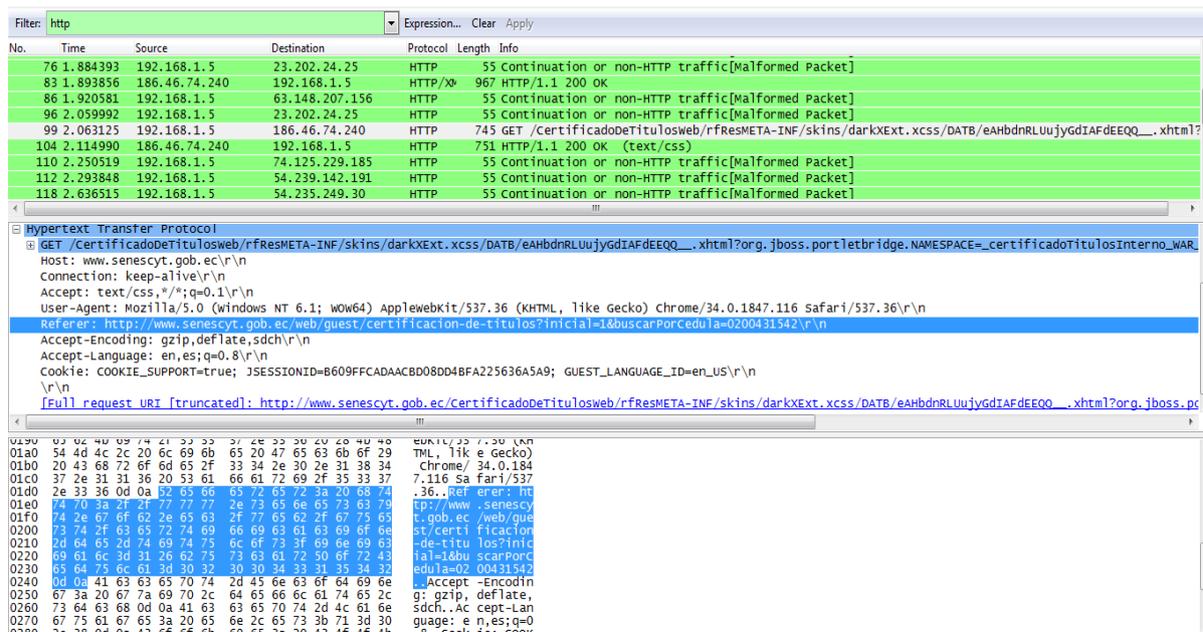


Figura 3-35 Captura de solicitud con Wireshark

A continuación se presentan los pasos a seguir para realizar una consulta a un servicio externo, que en este caso se realiza a la página de la SENESCYT.

- a) Almacenar los parámetros necesarios para la solicitud, en este caso el número de cédula.
- b) Instanciar una URI con la dirección de la SENESCYT para hacer la petición y los parámetros adecuados.
- c) Crear el objeto `Request` del tipo `WebRequest`, con el URI definido.
- d) Configurar `Request`, indicando que se va a usar el método POST.
- e) `Content-type`.
- f) La longitud y la versión del protocolo HTTP.
- g) Determinar un objeto para poder obtener un *stream* de la petición. Se va a añadir los parámetros como si fueran bytes, que al fin de cuentas es el contenido de la petición.
- h) Finalmente se procesa los datos que devuelve una llamada a `GetResponse`, mediante la librería `HtmlAgilityPack`, se verifica la respuesta a la solicitud, ya que puede ser que no existen datos para el número de cédula requerido.

En la Figura 3-36, el fragmento de código evidencia la ejecución de los puntos anteriormente detallados.

```
//Obtener y devolver datos sobre el Profesional Tecnico
public List<String> ObtenerProfesional(string cedula)
{
    List<String> datosprofe = new List<string>();
    //Crear los datos POST.
    string datosPost = "certificacion-de-titulos?inicial=1&buscarPorCedula=" + cedula;
    byte[] bytePost = Encoding.UTF8.GetBytes(datosPost);
    // Crear el requerimiento con el URL que puede recibir el POST.
    string uri = "http://www.senescyt.gob.ec/web/guest/" + datosPost;
    WebRequest requerimiento = WebRequest.Create("http://www.senescyt.gob.ec/web/guest/" + datosPost);
    requerimiento.Method = "POST";
    requerimiento.ContentType = "application/x-www-form-urlencoded";
    requerimiento.ContentLength = bytePost.Length;
    Stream dataStream = requerimiento.GetRequestStream();
    dataStream.Write(bytePost, 0, bytePost.Length);
    dataStream.Close();
    WebResponse response = requerimiento.GetResponse();
    dataStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    string responseFromServer = reader.ReadToEnd();
    string decodedXml = HttpUtility.UrlDecode(responseFromServer);
    List<String> html2 = new List<string>();
    html2 = contenidoEmpresa.ArchivoHtml(responseFromServer);
    datosprofe = html2;
    reader.Close();
    dataStream.Close();
    response.Close();
    return datosprofe;
}
```

Figura 3-36 Consulta a la SENESCYT mediante el método POST de HTTP

Se utiliza el método POST de HTTP, mediante un objeto `WebRequest`, para descargar la información de la página de la SENESCYT, procesarla y obtener la información que es de interés para el proyecto.

3.3.2.3.3. Consulta de Geocodificación a Google Maps

Como se requería de una consulta a un servicio que brinde información estructurada con respecto a la ubicación geográfica se optó por usar el servicio de Google Maps, el cual ofrece el servicio de cartografía digital más usado en el mundo, razón por la cual se optó por usar este servicio para la validación de las coordenadas geográficas de las estructuras que el concesionario deberá ingresar en el Formulario RC-2A.

Google Maps proporciona servicios web para utilizarlos en aplicaciones que requieran el uso de mapas. Los servicios web de Google Maps envían solicitudes HTTP a determinadas URL que devuelven datos geográficos en formato JSON o XML para ser procesados por cualquier tipo de aplicación. [41]

Para el caso del prototipo se requería del uso de la geocodificación de Google Maps, la geocodificación o codificación inversa es el proceso de convertir coordenadas geográficas en direcciones interpretables por las personas, por ejemplo: *0° 11' 17" S 78° 29'40" O a Ecuador, Pichincha, Quito, Mariana de Jesús y Antonio de Ulloa.*

De esta manera cuando el concesionario ingrese las coordenadas geográficas de las estructuras la aplicación de escritorio cambiará la latitud y longitud a formato decimal y las enviará al servicio web, el cual a su vez realizará las peticiones al servicio de *Google Maps* a través del siguiente URL:

<http://maps.googleapis.com/maps/api/geocode/xml?latlng=40.714224,-73.961452&sensor=false>

El servicio web envía como parámetros la latitud y la longitud en la URL además del parámetro sensor de ubicación *sensor*, el cual indica si se dispone de un sensor (por ejemplo, un localizador GPS) que en este caso es falso ya que generalmente se usa este parámetro en aplicaciones para dispositivos móviles.

La Figura 3-37 muestra un fragmento de código que específicamente permite descargar la información desde el servicio de Google Maps, envía las coordenadas previamente transformadas en formato decimal y descarga la información en un objeto de tipo XDocument, después llama a una función de nombre ProcesarDatos(), que será la que analice el archivo en busca únicamente de la información que el usuario requiere.

```

public void ObtenerCoordenadas(string coordenadas)
{
    string geocodeURL = "http://maps.googleapis.com/maps/api/" + "geocode/xml?latlng=" + coordenadas + "&sensor=false";
    try
    {
        geoDoc2 = XDocument.Load(geocodeURL);
        geoDoc2.Save(@"F:\coordenadas.xml");
        ProcesarDatos();
    }
    catch (WebException)
    {
        mensaje = "Verifique su conexión de Internet";
        return;
    }
}

```

Figura 3-37 Descargar información desde Google Maps

En la Figura 3-38 se muestra la interfaz gráfica del Formulario RC-2A, en la cual el concesionario además visualiza en el mapa la ubicación de las coordenadas ingresadas.

Figura 3-38 Interfaz gráfica del Formulario RC-2A

Para procesar la respuesta se escogió el formato XML ya que a través de las etiquetas del archivo XML se puede tener un mejor procesamiento de la información requerida.

Para realizar la geocodificación se consultaron las referencias: [42], [43], [44] y [45]. Todo el código referente a la consulta a Google Maps se encuentra disponible en el Anexo K.

3.3.3. APLICACIÓN WEB

Se planteó el desarrollo de una aplicación web para que los analistas de la DGGER puedan revisar la información tanto técnica como legal presentada en los trámites.

Para el desarrollo de la aplicación web se usó ASP.NET en el entorno de Visual Studio 2012. Esta aplicación cumple con las siguientes funciones:

- Permitir la autenticación del analista y del concesionario
- Permitir la descarga de la aplicación de escritorio
- Convertir el archivo XML enviado por el concesionario en información legible por el analista.
- Generar un Memorando en caso de existir inconsistencias en la información.
- Permitir el envío de un correo electrónico al concesionario en caso de existir inconsistencias en la información.

Para presentar la información de forma legible se usó un `GridView`, en el cual presenta la información necesaria al analista para realizar la última validación. A través de los identificadores de cada etiqueta se recorre el archivo XML presentando así la información indispensable.

En la Figura 3-39 se muestra un fragmento de código ASP.NET utilizado para la presentación de la información necesaria al analista.

```

Default.aspx.cs  Default.aspx  X
<asp:Label ID="lblNombreEmpresa" runat="server" Font-Bold="True" Font-Size="Medium"></asp:Label>
<asp:Panel ID="pnlXmlMdba" runat="server" HorizontalAlign="Center" ScrollBars="Auto">
<asp:GridView ID="dgvAplicacion" runat="server" AutoGenerateColumns="False" GridLines="None" Height="10px" Width="100%" Font-Size="Smaller" HorizontalAlign="Center" CellPadding="4" ForeColor="#333333" >
  <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
  <columns>
    <asp:boundfield datafield="TCS_NAME" headertext="Estructura" HeaderStyle-BorderColor="White" ItemStyle-BorderColor="White">
      <HeaderStyle BorderColor="White" />
      <ItemStyle BorderColor="White" />
    </asp:boundfield>
    <asp:boundfield datafield="DIRECCION" headertext="Dirección" ItemStyle-Width="5%" HeaderStyle-Width="5%" >
      <HeaderStyle Width="100%" />
      <ItemStyle Width="100%" />
    </asp:boundfield>
    <asp:boundfield datafield="LATITUD" headertext="Latitud"/>
    <asp:boundfield datafield="LONGITUD" headertext="Longitud"/>
    <asp:boundfield datafield="SID_LOC" headertext="Tipo de Soporte"/>
    <asp:boundfield datafield="SID_H_NM" headertext="Altura SMM"/>
    <asp:boundfield datafield="SID_STRUCT_HEIGHT" headertext="Altura Base-Cima "/>
    <asp:boundfield datafield="EAC_AN_H" headertext="Altura Base_Antena "/>
    <asp:boundfield datafield="EAC_AN_AZI" headertext="Azimut">
      <FooterStyle HorizontalAlign="Center" VerticalAlign="Middle" />
      <HeaderStyle HorizontalAlign="Center" VerticalAlign="Middle" Wrap="True" />
    </asp:boundfield>
    <asp:boundfield datafield="EAC_AN_POL" headertext="Polarización "/>
    <asp:boundfield datafield="EAC_AN_ELEV" headertext="Angulo de Elevación"/>
    <asp:boundfield datafield="ANTENA" headertext="Antena Asociada"/>
    <asp:boundfield datafield="EAN_TYPE" headertext="Tipo de antena"/>
    <!--<asp:boundfield datafield="RANGO_FREQ" headertext="Rango de Frecuencias"/-->
    <asp:boundfield datafield="ESTACION" headertext="Equipo"/>
    <asp:boundfield datafield="EQ_MULTIPLEX" headertext="Tipo de Operación (Multiplexación)"/>
    <asp:boundfield datafield="EQ_COMMENT" headertext="Potencia de salida"/>
    <asp:boundfield datafield="TCS_FX_SERVICE_TYPE" headertext="Enlace"/>
  </columns>
  <EditRowStyle BackColor="#999999" />
  <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
  <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
  <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
  <RowStyle HorizontalAlign="Center" VerticalAlign="Middle" BackColor="#F7F6F3" ForeColor="#333333" />
  <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />

```

Figura 3-39 Fragmento de código ASP.NET

3.3.3.1. Adjuntar documentos en ASP.NET

Para generar el Memorando en Microsoft Word se debe agregar la referencia Microsoft.Office.Interop.Word.dll (Ver Figura 3-40).

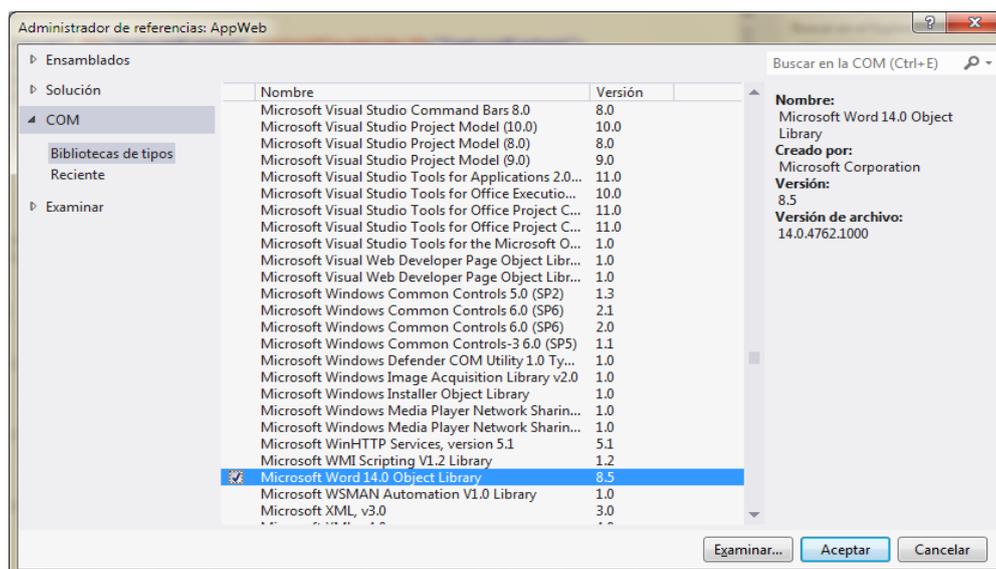


Figura 3-40 Referencia Word

Se debe establecer un método para crear el documento a través de una plantilla, de esta manera cuando el analista de la DGGER desee enviar un Memorando al concesionario la aplicación web permitirá editar un documento de Word con el formato establecido por la SENATEL.

En la Figura 3-41 se muestra un fragmento de código de la generación del Memorando.

```

Default.aspx.cs*  Default.aspx*
AppWeb_Default  MiDocumento()

private void MiDocumento()
{
    Object oMissing = System.Reflection.Missing.Value;
    //Creamos una instancia de una Aplicación Word.
    var objWordApplication = new Word.Application();
    object fileName = (Server.MapPath(ConfigurationManager.
        AppSettings["RutaXmIs"] + "Memo.docx"));
    //A la aplicación Word, le añadimos un documento.
    Word.Document objWordDocument = objWordApplication.Documents.Add(ref fileName, ref oMissing,
        ref oMissing, ref oMissing);
    //Activamos el documento recién creado, de forma que podamos escribir en el
    objWordDocument.Activate();
    //Empezamos a escribir
    //Hace visible la Aplicación para ver lo escrito
    objWordApplication.Visible = true;

    objWordApplication.Selection.Font.Size = 12; //Tamaño de la Fuente
    objWordApplication.Selection.Font.Bold = 1; // Negrita
    objWordApplication.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;
    objWordApplication.Selection.TypeText("MEMORANDO No. DGGER-2014-");
    //Indicamos que el texto anterior es parte de un párrafo.

    objWordApplication.Selection.TypeParagraph();
    objWordApplication.Selection.TypeParagraph();
    objWordApplication.Selection.TypeParagraph();
    objWordApplication.Selection.TypeParagraph();
}

```

Figura 3-41 Fragmento de código de la generación del Memorando

Para realizar el memorando se consultaron las referencias: [46] y [47].

3.3.3.2. Envío de Correos desde ASP.NET

Para enviar correos electrónicos a los concesionarios desde la aplicación web desarrollada en ASP.NET primero se debe importar la librería `System.Net.Mail`, después se debe configurar el protocolo SMTP y la cuenta de correo desde la cual se enviarán los correos, se usó una cuenta de Gmail en el desarrollo del prototipo.

Cabe señalar que para el funcionamiento dentro de la SENATEL se deberá cambiar la configuración SMTP al servidor de correo con el cual se opere internamente.

En la Figura 3-42 se muestra el fragmento de código que permite el envío de correos electrónicos. De la librería `System.Net.Mail`, se usan las clases `MailMessage` para crear el mensaje (asunto, cuerpo del mensaje, etc), `MailAddress` para indicar quien envía el correo.

A continuación se crea una instancia de la clase `SmtpClient` para configurar el servidor de correo. Finalmente se encapsula el envío dentro de un `try` en caso de existir problema al enviar el correo.

```
protected void BtnEnviarClick(object sender, EventArgs e)
{
    // Configuración del mensaje
    var msg = new MailMessage();
    msg.To.Add(txtDestinatario.Text);
    msg.From = new MailAddress("andrepin29@gmail.com", "SENATEL", System.Text.Encoding.UTF8);
    msg.Subject = txtAsunto.Text ;
    msg.SubjectEncoding = System.Text.Encoding.UTF8;
    msg.Body = txtMensaje.Text;
    msg.BodyEncoding = System.Text.Encoding.UTF8;
    msg.IsBodyHtml = false;

    //Configuración del SMTP
    var client = new SmtpClient
    {
        Credentials = new System.Net.NetworkCredential("andrepin29@gmail.com", "xx"),
        Port = 587,
        Host = "smtp.gmail.com",
        EnableSsl = true
    };

    try
    {
        client.Send(msg);
    }
    catch (SmtpException ex)
    {
        Console.WriteLine(ex.Message);
        Console.ReadLine();
    }
    Response.Redirect("/Default.aspx");
}
```

Figura 3-42 Envío de correo en ASP.NET

En la Figura 3-43 se muestra la interfaz para el envío de correo en la aplicación web.

Para realizar el memorando se consultaron la referencia: [48]. Todo el código referente a la aplicación web se encuentra disponible en el Anexo K.

Figura 3-43 Interfaz de envío de correo

3.4.CAPAS ARQUITECTÓNICAS DEL SOFTWARE

En el presente Proyecto se utilizó la arquitectura SOA, como se mencionó en el Capítulo I, SOA promueve la reutilización masiva de recursos, permitiendo que una aplicación pueda consumir un servicio indiferente de cual sea la tecnología o lenguaje en el que está desarrollado.

En la Figura 3-44 se presentan algunas plataformas con las que se podrían desarrollar aplicaciones basándose en la arquitectura SOA.

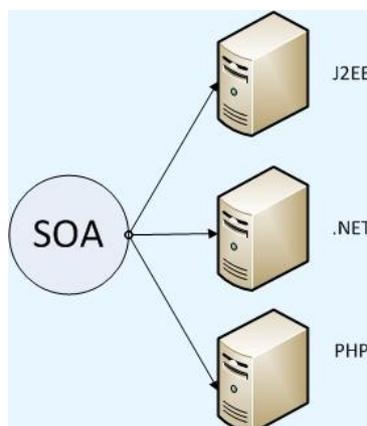


Figura 3-44 Algunas plataformas para el desarrollo de aplicaciones SOA

SOA separa la lógica de negocio, los datos y su presentación de forma independiente, distribuyendo funciones y a la vez mejorando el rendimiento del sistema, las capas de SOA trabajan con estándares que no necesariamente son obligatorios, este aspecto además de marcar la diferencia con cualquier otra arquitectura que pueda parecersele, permite una mayor interoperabilidad y lo hace sencillo de consumir convirtiéndola en una arquitectura idónea para el desarrollo de sistemas distribuidos (Ver Figura 3-45).



Figura 3-45 Capas de la arquitectura SOA

La descripción de los elementos de esta arquitectura y como se encuentran plasmados en el presente Proyecto permite dilucidar las funciones y procesos que se realizan.

A continuación se describen las Capas Aplicación, de Servicios, de Procesos de Negocio y Datos.

3.4.1. CAPA APLICACIÓN

La Capa Aplicación o Presentación del sistema corresponde a la interfaz de usuario que provee una solución a la necesidad del usuario de interactuar con el sistema. El prototipo de sistema Distribuido cuenta con dos interfaces de usuario una desarrollada en WPF que corresponde a la Aplicación de Escritorio y otra desarrollada en ASP.NET correspondiente a la Aplicación Web.

La aplicación WPF desarrollada facilita interfaces (ventanas) para los trámites tanto de Sistemas MDBA, como Servicio MT-RDV y a su vez cada una de estas interfaces cuenta con los controles que el concesionario requiere para el ingreso de la información legal y técnica que se mencionó en el Capítulo II. La aplicación Web facilita la visualización de los trámites ingresados por el concesionario.

3.4.1.1. Interfaz de la Aplicación de Escritorio

El usuario encontrará en la aplicación de escritorio ventanas principales tanto para Sistemas MDBA como para el Servicio MT-RDV, además de un diverso número de ventanas secundarias para cada una de las ventanas principales, proporcionando al usuario una potente interfaz gráfica, que se basa su funcionamiento en la programación orientada a objetos, creando instancias de formularios, clases y controles personalizados que representan de forma organizada los campos necesarios para que el concesionario ingrese la información técnica y legal.

La ventana de Servicio MT-RDV difiere visualmente de la ventana de Sistemas MDBA, debido a los formularios que implementa, por ejemplo, el Servicio MT-RDV maneja el formulario RC-1A y MDBA el formulario RC-1B, además el Servicio MT-RDV no necesita los formularios RC-9A y RC-9B como en MDBA, pero si requiere del formulario 5A.

Para cada sub-ventana que se necesite se realiza una instancia de un objeto de tipo formulario, permitiendo hacer uso de cada control que posee el formulario instanciado, las ventanas principales solo pueden crear una sub-ventana de cada tipo como se puede evidenciar por las relaciones presentes en las Figura 3-46 y

Figura 3-47, y además el proceso se debe llevar a cabo una a la vez ya que mantienen una relación modal.

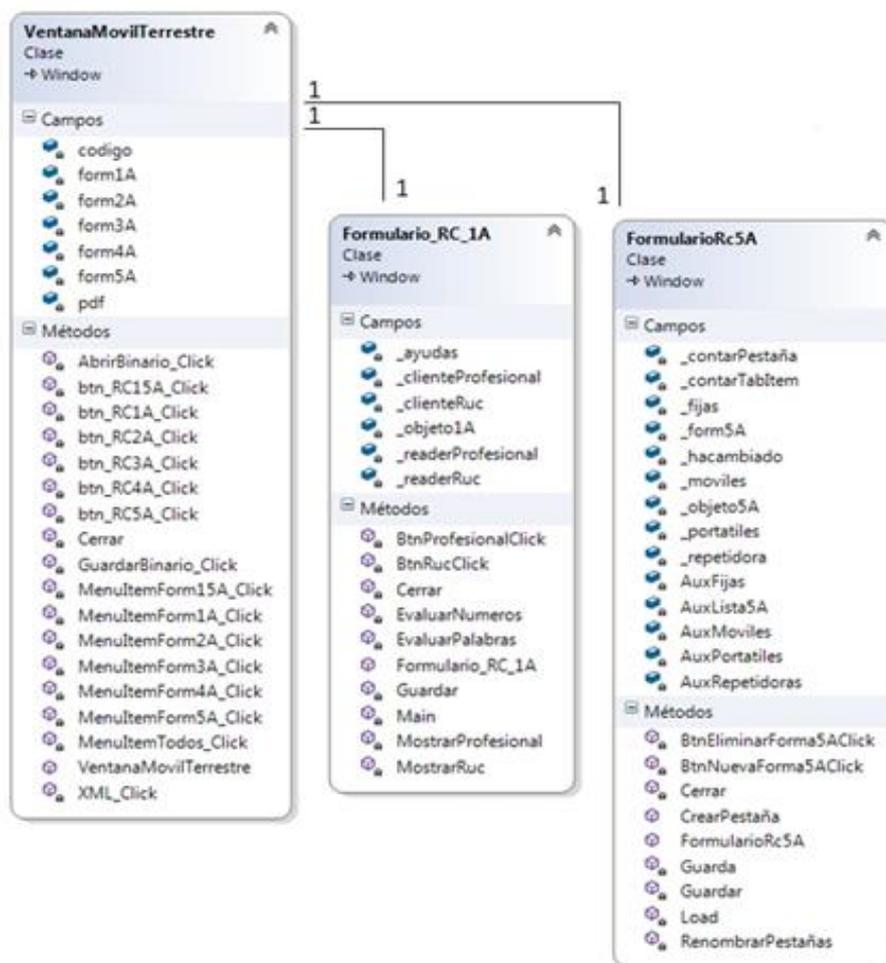


Figura 3-46 Diagrama de clases de los formularios 1A y 5A (Servicio MT-RDV)



Figura 3-47 Diagrama de clases de los formularios MDBA

La información legal que corresponde tanto a Móvil Terrestre como a MDBA se ingresa una vez por cada trámite que se implemente, de tal forma que los formularios RC-1A y RC-1B (Figura 3-48) proporcionan en su entorno gráfico todos los elementos necesarios para que el concesionario pueda registrar la información necesaria acorde a los contenidos expuestos en el Capítulo II. Cabe mencionar que cada formulario crea un objeto de su correspondiente clase, que le permite guardar y reconstruir la información como se verá más adelante.



Figura 3-48 Diagrama de clases de los formularios y clases RC-1A y RC-1B

En un trámite de Sistema MDBA o Servicio MT-RDV, se pueden ingresar diferentes números de estructuras, antenas, equipos, formularios de enlaces, etc.

En primera instancia los formularios encargados de recibir esta información se deberían repetir o crear cuantas veces sea necesario, operativamente no es lo más provechoso ya que no existiría un orden, consumiría muchos recursos y además visualmente no sería lo adecuado.

Los controles personalizados ofrecen una solución, cada formulario no contendrá directamente los objetos (controles) necesarios para que el usuario ingrese la información, sino más bien que estos estarán alojados en un objeto diferente (control personalizado) con un nombre que identifique claramente al formulario que pertenece, estos controles estarán disponibles en el Cuadro de Herramientas agrupados en una categoría identificada con el mismo nombre del SpaceName del proyecto.

De esta forma con tan sólo declararlo correctamente se puede crear en tiempo de ejecución tantos controles como sean necesarios y se aplica para los formularios RC-2A, RC-3A, RC-4A y RC-9A.

Es necesario especificar que para cada formulario existe una clase para guardar la información en ellos ingresada.

En la Figura 3-49 se muestran los diagramas de clases de los Formulario RC-2A y RC-3A, y en la Figura 3-50 de los formularios RC-4A y RC-9B.

En cada uno de los diagramas de clase se aprecia: la clase, el respectivo formulario Window y el control personalizado UserControl. Así, por ejemplo, la Clase_2A se usa para guardar y reconstruir el Formulario_RC_2A, el mismo que contiene al control personalizado Formato2A.

La información que se ingresa en los formularios RC-5A y RC-9B tiene una variante, ya que se requiere de la implementación de controles personalizados anidados, como se evidencia en la Figura 3-51 y Figura 3-52, es decir que se crea un control dentro de otro ya existente.

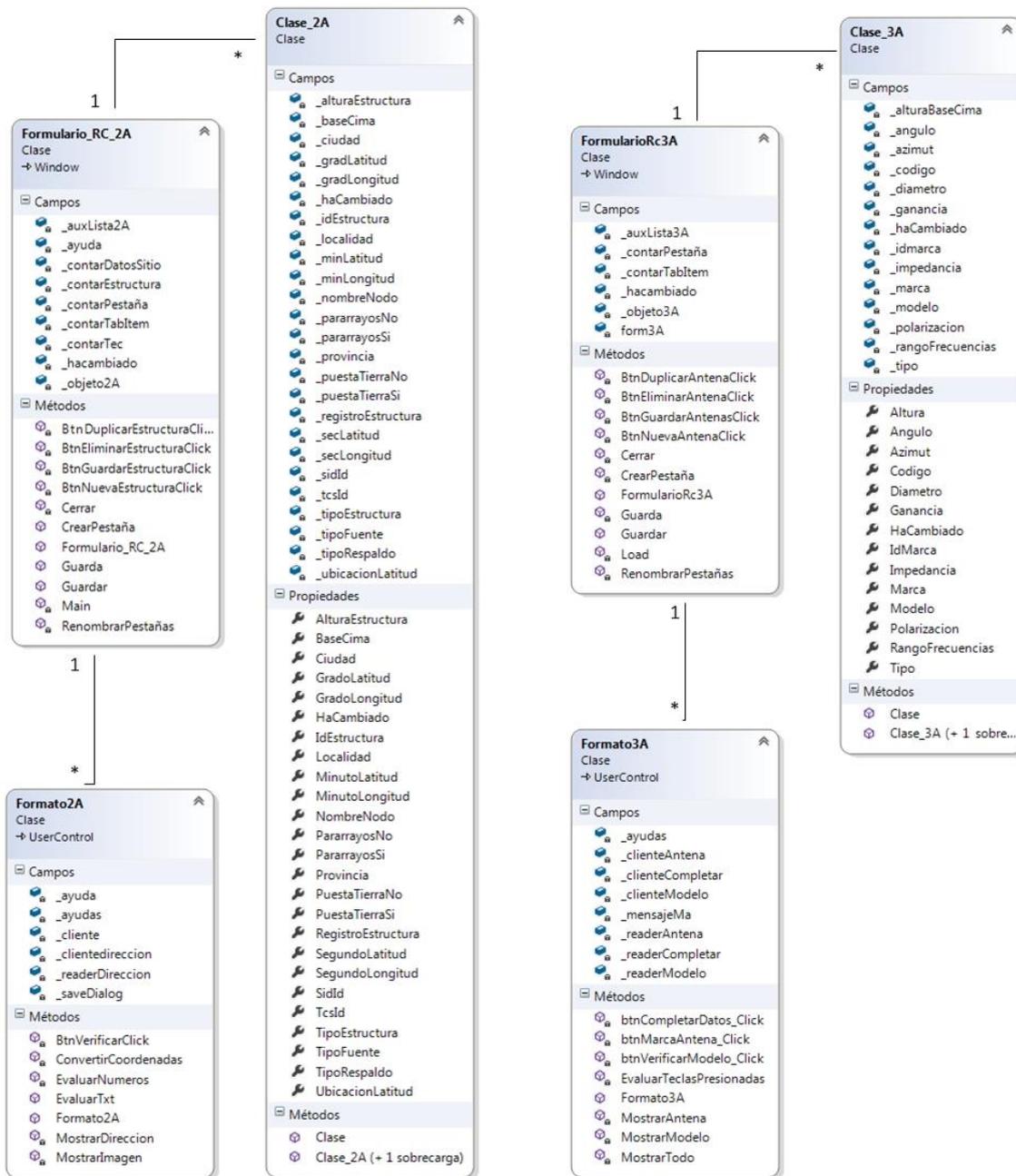


Figura 3-49 Diagrama de clases de los formularios RC-2A y RC-3A

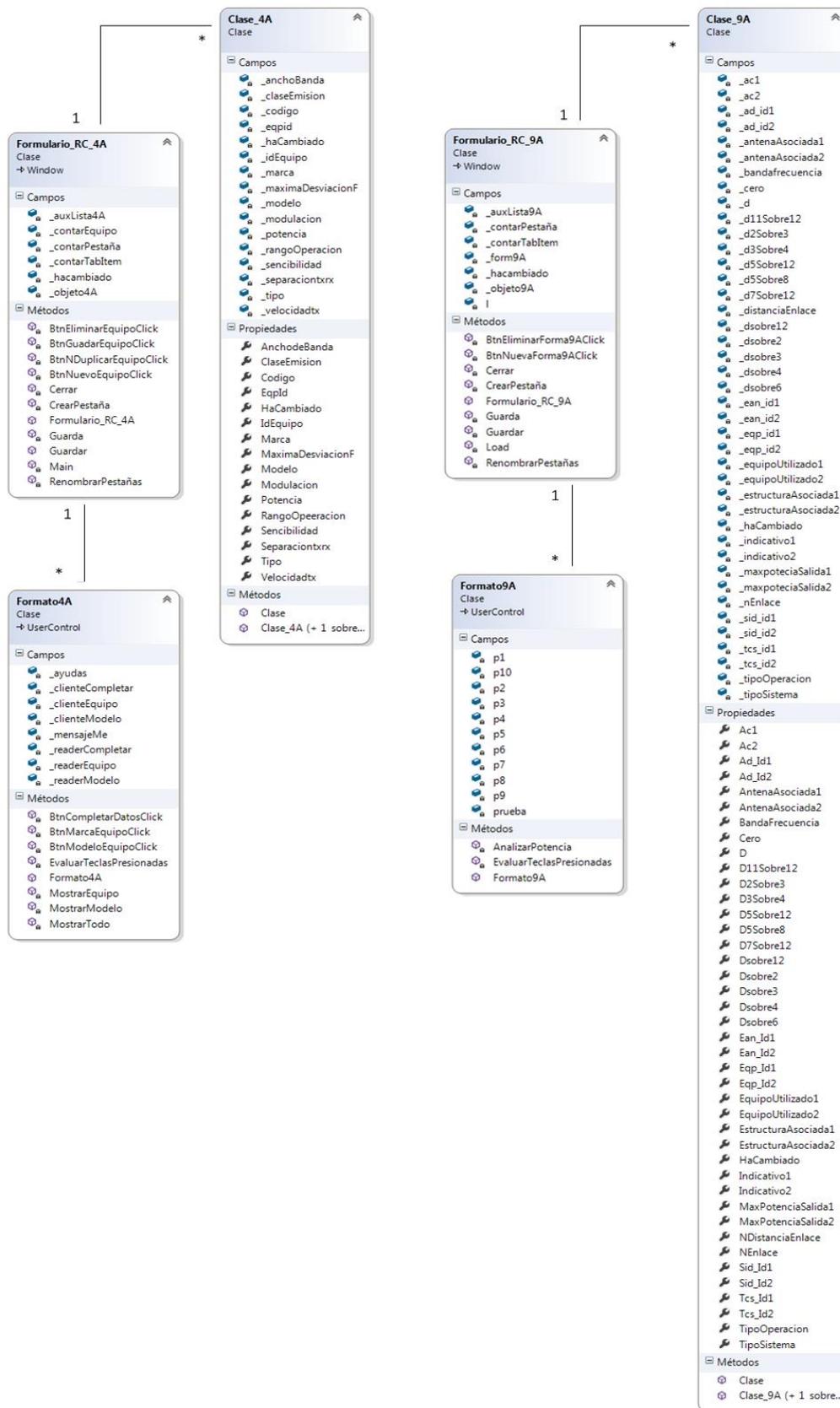


Figura 3-50 Diagrama de clases de los formularios RC-4A y RC-9A

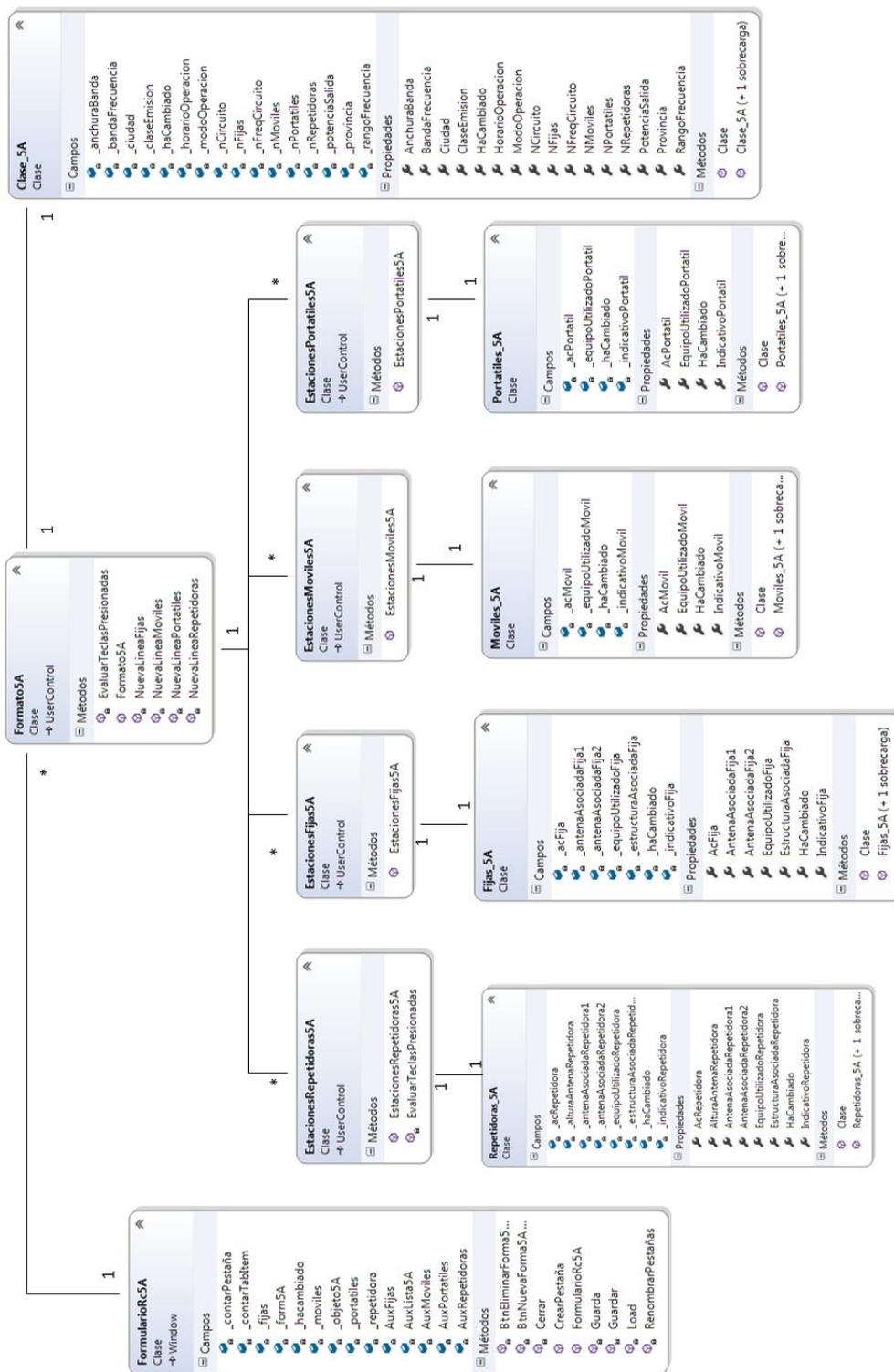


Figura 3-51 Diagrama de clases del formulario RC-5A

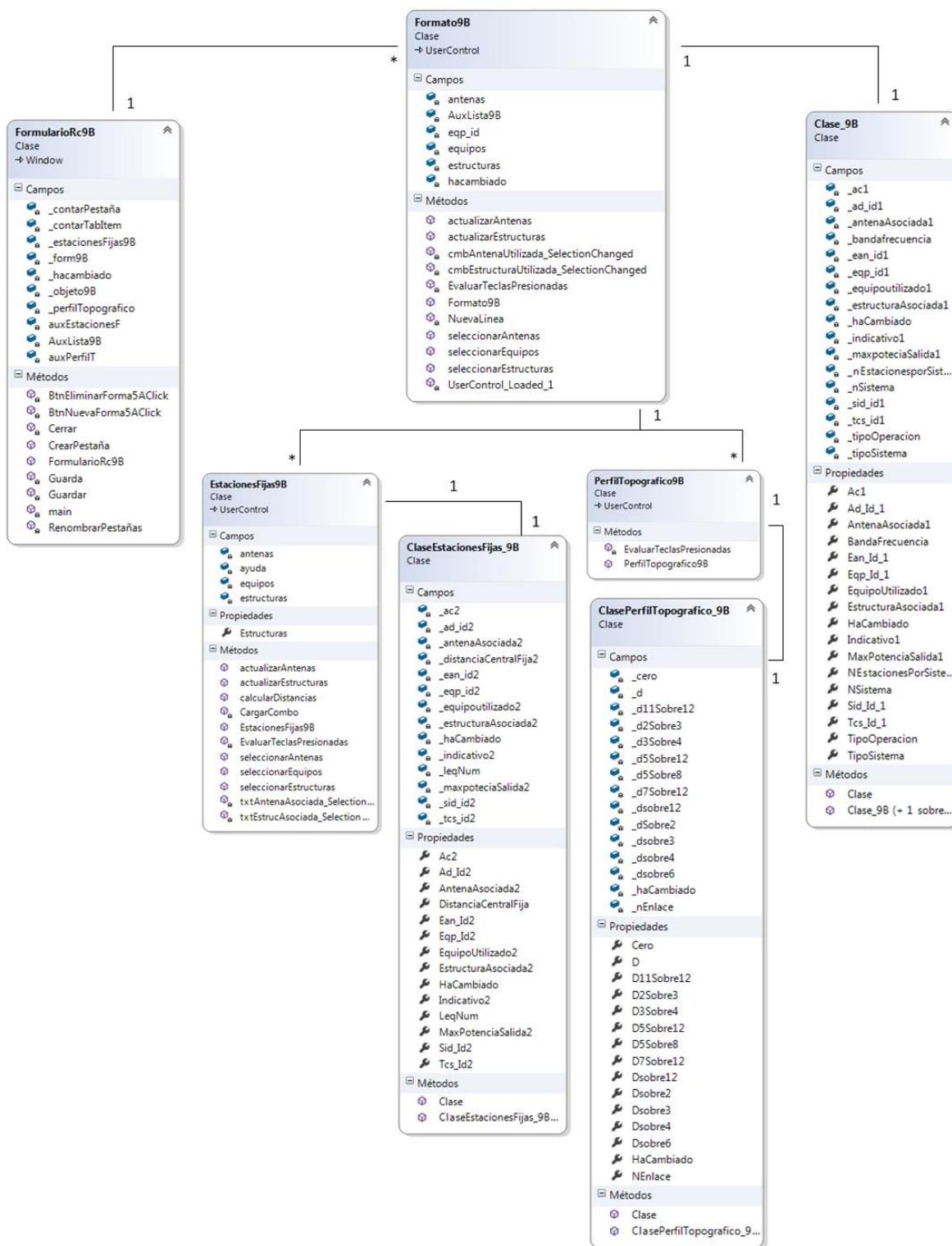


Figura 3-52 Diagrama de clases del formulario RC-9B

3.4.1.1.1. Guardar y reconstruir información en la Aplicación de escritorio

Guardar avances del trabajo es indispensable incluso como medida de seguridad, para brindar solución a este requerimiento, la aplicación de escritorio implementa un sistema que almacena los objetos que el usuario crea al interactuar con los distintos formularios de la aplicación de escritorio. Este sistema consiste en una clase denominada `Persistencia` que declara un conjunto de listas con todos los tipos de objetos presentes en la aplicación de escritorio, de esta forma el usuario puede almacenar toda la información ingresada en cada uno de los formulario (Ver Figura 3-53).

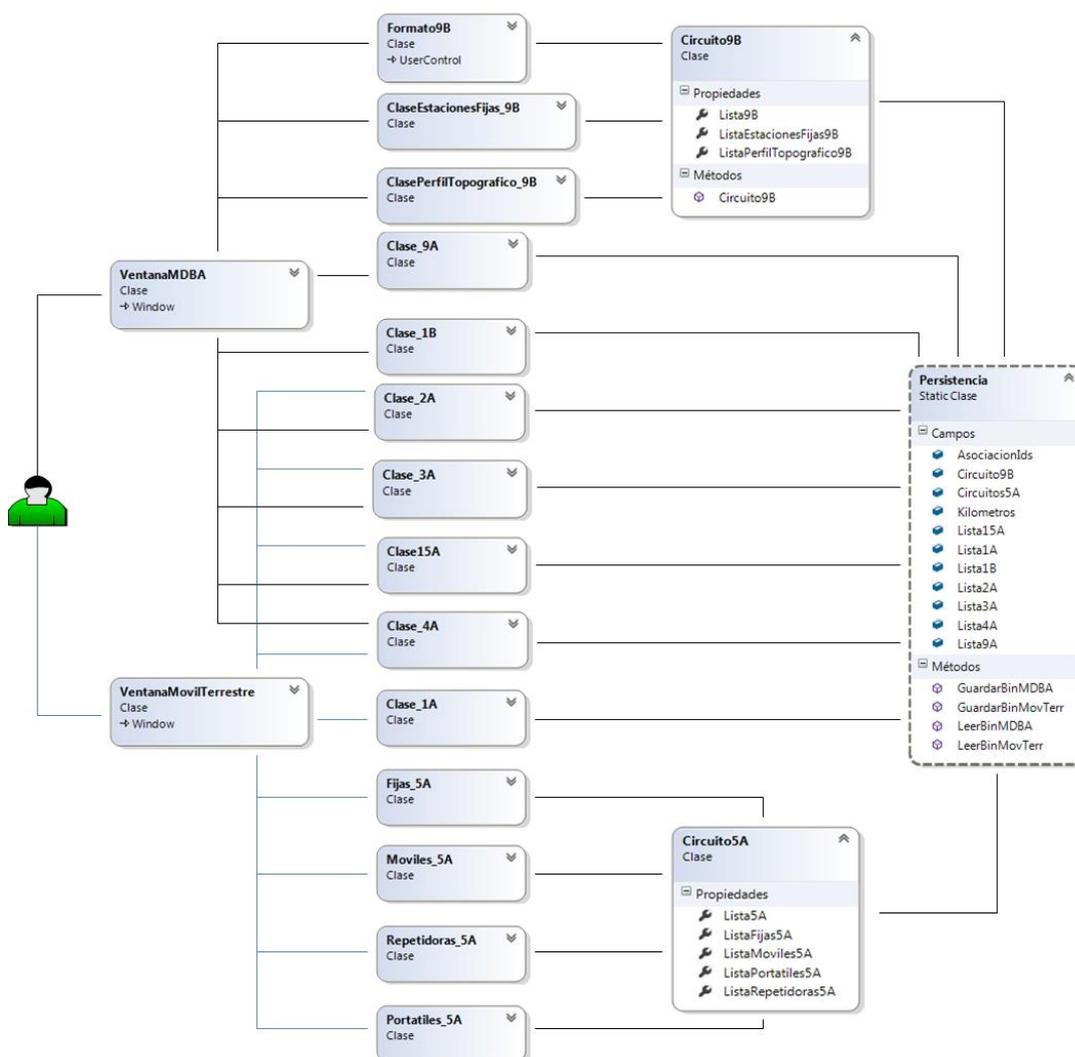


Figura 3-53 Diagrama de clases del almacenamiento de la Información

La clase `Persistencia` proporciona un servicio temporal de almacenamiento, es decir que si la aplicación se cierra se pierden los datos almacenados en esta clase, para brindar una solución más duradera se optó por definir una clase denominada `PersistenciaBinaria`, a la que además se la marco como `[Serializable()]`, todo esto con el fin de crear un archivo binario ligero que permita guardar y volver a cargar la información que almacena el archivo.

En la Figura 3-54 se puede visualizar un Fragmento de la clase `PersistenciaBinaria`.

```
[Serializable()]
public class PersistenciaBinaria
{
    public List<Clase_1A> Lista1A
    {
        get;
        set;
    }
    public List<Clase_1B> Lista1B
    {
        get;
        set;
    }
    public List<Clase_2A> Lista2A
    {
        get;
        set;
    }
    public List<Clase_3A> Lista3A
    {
        get;
        set;
    }
    public List<Clase_4A> Lista4A
    {
        get;
        set;
    }
}
```

Figura 3-54 Fragmento de la clase `PersistenciaBinaria`

La clase `PersistenciaBinaria` permite hacer una copia exacta de la clase `Persistencia`, para ello es necesario que las dos clases cuenten con el mismo número de elementos a almacenar y que estos sean del mismo tipo.

En la Figura 3-55 se muestra los métodos `GuardarBinMdba` y `AbrirBinMdba` que permiten guardar y reconstruir la información de los trámites o procesos correspondientes al servicio MDBA.

Se evidencia que tanto para guardar como para abrir necesito un objeto del tipo `PersistenciaBinaria`.

```
public static void GuardarBinMdba(string pathBin)
{
    PersistenciaBinaria AlmacenBin = new PersistenciaBinaria();
    AlmacenBin.Lista1B = Lista1B;
    AlmacenBin.Lista2A = Lista2A;
    AlmacenBin.Lista3A = Lista3A;
    AlmacenBin.Lista4A = Lista4A;
    AlmacenBin.Lista9A = Lista9A;
    AlmacenBin.Circuito9B = Circuito9B;
    AlmacenBin.Lista15A = Lista15A;

    Stream StreamArchivoBinario = File.Create(pathBin);
    BinaryFormatter serializador = new BinaryFormatter();
    serializador.Serialize(StreamArchivoBinario, AlmacenBin);
    StreamArchivoBinario.Close();
}

public static void LeerBinMdba(string pathBin)
{
    Stream StreamArchivoBinario = File.Open(pathBin, FileMode.Open);
    BinaryFormatter serializador = new BinaryFormatter();
    PersistenciaBinaria AlmacenBin = (PersistenciaBinaria)serializador.Deserialize(StreamArchivoBinario);
    Lista1B = AlmacenBin.Lista1B;
    Lista2A = AlmacenBin.Lista2A;
    Lista3A = AlmacenBin.Lista3A;
    Lista4A = AlmacenBin.Lista4A;
    Lista9A = AlmacenBin.Lista9A;
    Circuito9B = AlmacenBin.Circuito9B;
    Lista15A = AlmacenBin.Lista15A;
    StreamArchivoBinario.Close();
}
```

Figura 3-55 Métodos GuardarBinMdba y AbrirBinMdba

3.4.1.2. Interfaz de la Aplicación Web

La aplicación web provee una interfaz de usuario fácil de utilizar la cual consta de una página maestra `MasterPage` la cual permite crear una plantilla para todas las páginas `Pages` que se hereden de ella, manteniendo así un formato consistente en todo el sitio web.

En la Figura 3-56 se muestra en el diagrama de clases las páginas construidas, una página para la autenticación de los usuarios, otra para la visualización de los trámites y una para el envío de correo electrónico.

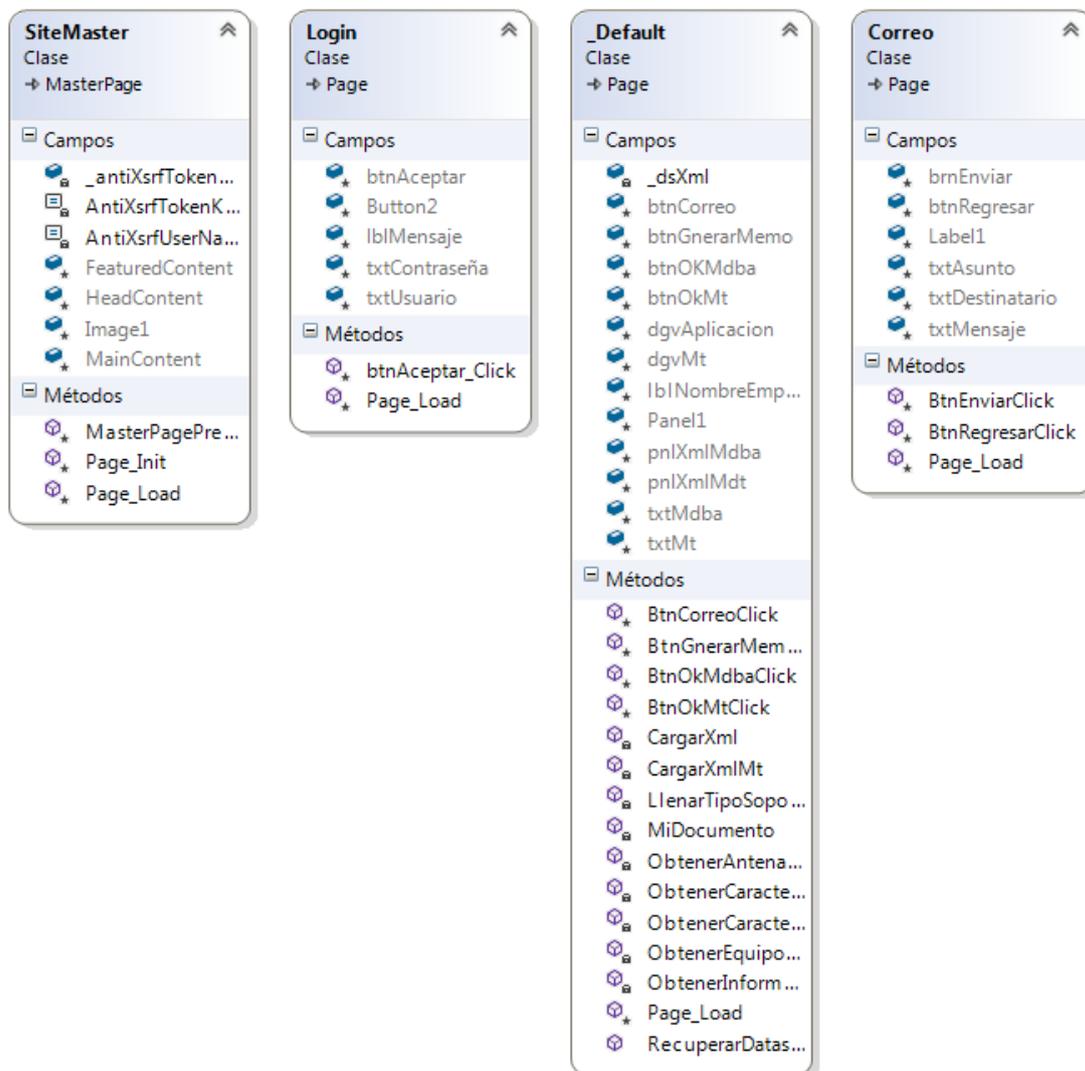


Figura 3-56 Diagrama de clases de la Aplicación Web

3.4.2. CAPA DE SERVICIOS

El desafío en el desarrollo de sistemas SOA se encuentra precisamente en contar con los servicios adecuados, el análisis de los requisitos debe proveer de la visión correcta para determinar qué componentes o funcionalidades son las que se deben publicar, estructurando una correcta abstracción de las necesidades y la lógica de negocio.

Es importante que un servicio exponga de forma clara y concisa las funcionalidades que posee y cómo un usuario puede acceder a ellas, en esta descripción debe constar la información que ciertas operaciones requieren que se entregue como parámetros de entrada, para que puedan completar todos los procesos necesarios y finalmente devolver los resultados al cliente.

Precisamente WSDL permite lograr estos intercambios de información gracias a la estandarización en los formatos de comunicación y mensajes, dejando como responsabilidad del desarrollador la elección del formato que más se ajuste a sus objetivos.

La descripción WSDL es un documento XML con una estructura establecida previamente, en este caso, para el desarrollo de los servicios WCF en .NET, como programadores directamente no existe la responsabilidad de detallar el documento WSDL, este se genera automáticamente, considerando dos aspectos importantes del archivo `Web.config`.

El primero es habilitar que se genere `serviceMetadata` para el extremo correspondiente y se puedan descubrir los métodos del servicio mediante HTTP o HTTPS, en este punto no es obligación habilitar `includeExceptionDetailInFaults`, pero es recomendable para recibir informes con cierto detalle sobre un error suscitado en el servicio.

El segundo es definir el contrato de servicio, este aspecto es muy importante y está ligado con uno de los elementos principales del WSDL como es como lo es `service`, permite indicar cuál es el nombre del punto de montaje del servicio por medio del cual se puede acceder a los métodos definidos en el contrato (Interfaz), para ello en el atributo `contract` del `<endpoint>` se debe indicar el espacio de nombre al que pertenece el contrato (Interfaz) seguido de un punto y el nombre de la interfaz. Para el presente servicio el nombre es: `ServicioSenatel.IServicioSenatel`.

En la Figura 3-57 se muestra un fragmento del archivo `Web.config`.

```

<system.serviceModel>
  <services>
    <service name="ServicioSenatel.ServicioSenatel" behaviorConfiguration="de
      <endpoint address="" behaviorConfiguration="web" binding="webHttpBindin
        bindingConfiguration="RestBinding" name="ServicioSenatel"
        contract="ServicioSenatel.IServicioSenatel" />
    </service>
  </services>
  <behaviors>
    <serviceBehaviors>
      <behavior name="default">
        <serviceMetadata httpGetEnabled="true"/>
        <serviceDebug includeExceptionDetailInFaults="true"/>
      </behavior>
    </serviceBehaviors>
  </behaviors>

```

Figura 3-57 Archivo Web.config contract y serviceMetadata

El archivo WSDL resultado de la configuración del servicio WCF no es típicamente un archivo XML, sino que más bien se trata de un esquema XML en el que se encuentra detallados los siguientes elementos principales [49]:

- **Types**, los elementos `Types` proporcionan las definiciones de tipos de datos que se utilizan para describir los mensajes intercambiados (Ver Figura 3-58).

```

▼<wsdl:types>
  ▼<xsd:schema targetNamespace="http://tempuri.org/Imports">
    <xsd:import
      schemaLocation="http://localhost:50268/ServicioSenatel.svc?xsd=xsd0"
      namespace="http://tempuri.org/" />
    <xsd:import
      schemaLocation="http://localhost:50268/ServicioSenatel.svc?xsd=xsd1"
      namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    <xsd:import
      schemaLocation="http://localhost:50268/ServicioSenatel.svc?xsd=xsd2"
      namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <xsd:import
      schemaLocation="http://localhost:50268/ServicioSenatel.svc?xsd=xsd3"
      namespace="http://schemas.microsoft.com/Message" />
  </xsd:schema>
</wsdl:types>

```

Figura 3-58 Fragmento del archivo WSDL, Types

- **Message**, los elementos `Message` proporcionan la abstracción de los datos que serán transmitidos (Ver Figura 3-59).

```

▼<wsdl:message name="IServicioSenatel_MarcaAntena_InputMessage">
  <wsdl:part name="parameters" element="tns:MarcaAntena" />
</wsdl:message>
▼<wsdl:message name="IServicioSenatel_MarcaAntena_OutputMessage">
  <wsdl:part name="parameters" element="tns:MarcaAntenaResponse" />
</wsdl:message>

```

Figura 3-59 Fragmento del archivo WSDL, Message

- **PortType**, son un conjunto de abstracciones que se refieren a los mensajes que intercambian cada uno de los métodos definidos en la interfaz tanto de entrada como de salida (Ver Figura 3-60).
- **Service**, son un conjunto de puertos relacionados (Ver Figura 3-60).

```
▶ <wsdl:portType name="IServicioSenatel">...</wsdl:portType>
  <wsdl:service name="ServicioSenatel"/>
```

Figura 3-60 Fragmento del archivo WSDL, operaciones en Service

- **Binding**, provee especificaciones de protocolo de formato de datos para las operaciones y los mensajes definidos por un portType particular (Ver Figura 3-61).
- **Port**, que especifica una dirección de una unión, definiendo así un único criterio de valoración de comunicación (Ver Figura 3-61).

```
▼ <wsdl:portType name="IServicioSenatel">
  ▼ <wsdl:operation name="MarcaAntena">
    <wsdl:input
      wsaw:Action="http://tempuri.org/IServicioSenatel/MarcaAntena"
      message="tns:IServicioSenatel_MarcaAntena_InputMessage"/>
    <wsdl:output
      wsaw:Action="http://tempuri.org/IServicioSenatel/MarcaAntenaResponse"
      message="tns:IServicioSenatel_MarcaAntena_OutputMessage"/>
    </wsdl:operation>
  ▼ <wsdl:operation name="MarcaEquipo">
    <wsdl:input
      wsaw:Action="http://tempuri.org/IServicioSenatel/MarcaEquipo"
      message="tns:IServicioSenatel_MarcaEquipo_InputMessage"/>
    <wsdl:output
      wsaw:Action="http://tempuri.org/IServicioSenatel/MarcaEquipoResponse"
      message="tns:IServicioSenatel_MarcaEquipo_OutputMessage"/>
    </wsdl:operation>
  ▼ <wsdl:operation name="ModeloAntena">
    <wsdl:input
      wsaw:Action="http://tempuri.org/IServicioSenatel/ModeloAntena"
      message="tns:IServicioSenatel_ModeloAntena_InputMessage"/>
    <wsdl:output
      wsaw:Action="http://tempuri.org/IServicioSenatel/ModeloAntenaResponse"
      message="tns:IServicioSenatel_ModeloAntena_OutputMessage"/>
    </wsdl:operation>
```

Figura 3-61 Fragmento del archivo WSDL, operaciones en PortType

Incuestionablemente distribuir la lógica de los procesos trae consigo grandes ventajas, como impedir la saturación de un determinado elemento del sistema, la disminución de tiempos de espera y mejorar la eficiencia de las aplicaciones.

De esta manera el servicio debe proporcionar a la aplicación de escritorio mecanismos para validar la información que se sumen a los que ya posee, permitiendo un mejor desempeño.

3.4.2.1. Interfaz

La interfaz alberga o representa el `[ServiceContract]`, define lo que puede hacer el servicio, es decir que representa el acuerdo entre las partes involucradas definiendo, las funciones que se pueden usar, los datos que se deben enviar y los que se recibirán y de qué manera se debe realizar el intercambio de estos datos. En la Figura 3-62 se muestra la interfaz con los métodos a publicar y la clase que los implementa.

Los siguientes son los contratos disponibles:

- **Service Contract**, enumera las operaciones que el servicio publicará. Se utiliza `[ServiceContract]` para definir el contrato y `[OperationContract]` para identificar la función que se publicará (WSDL), se puede hacer de dos maneras, primero y la más común anteponiendo `[ServiceContract]` al nombre de interfaz para luego enlistar las funciones que a posterior deberán ser implementadas por una clase o, aplicando `[ServiceContract]` directamente a la clase que implementa los métodos.
- **Data Contract**, describe una estructura particular de datos que manejará una determinada operación, se lo puede implementar usando la sintaxis `[DataContract]` e identificar cada uno de sus miembros con `[DataMember]`.

- **Message Contract**, especifica el contenido de un mensaje.

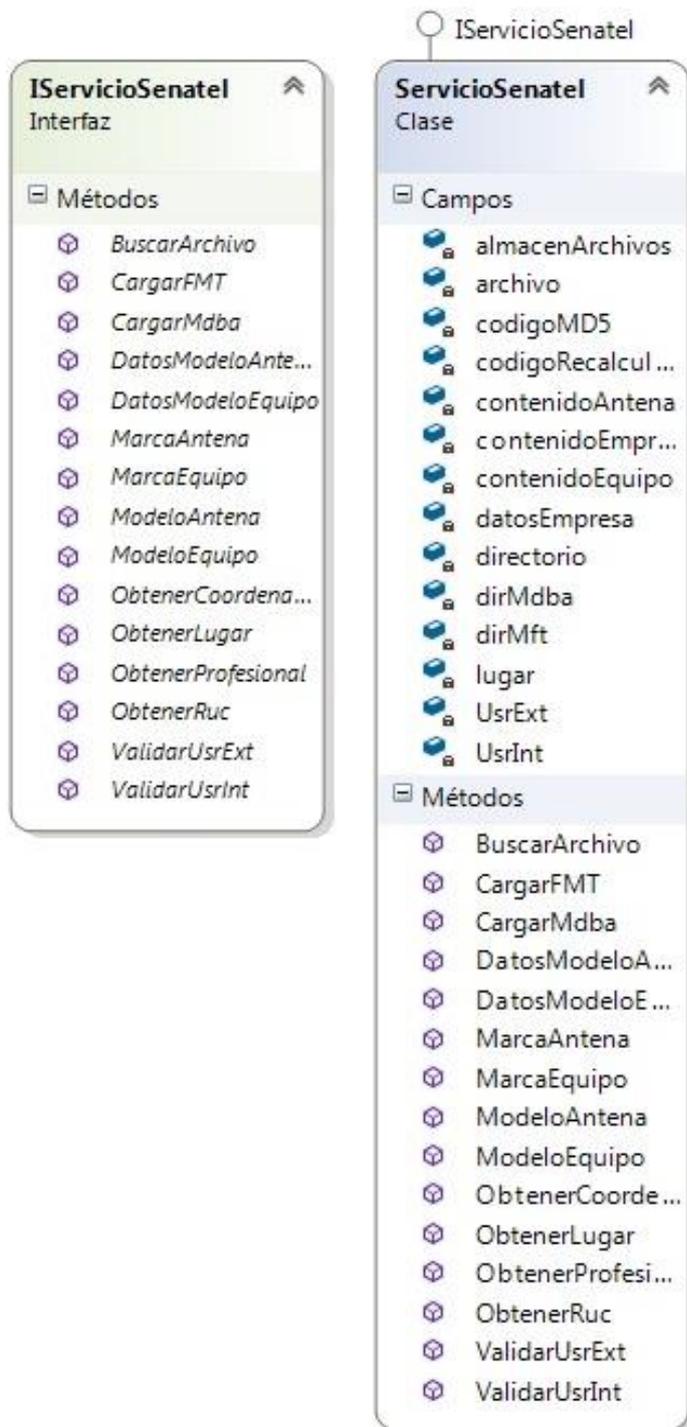


Figura 3-62 Interfaz con los métodos a publicar y la clase que los implementa

3.4.2.2. Data Binding

Se encargan de especificar como se comunican los puntos finales del servicio (EndPoints), es decir que procedimiento tomar en cuanto a transporte, codificación, seguridad, tipo de mensaje, etc. usará un determinado método para comunicarse con el usuario.

Los `bindings` en el `Web.config` se agrupan dentro de la etiqueta `<bindings>`, se permite crear `bindings` personalizados a medida de los requerimientos, un `binding` posee varios atributos entre ellos, `name` y `namespace` identifican al `binding` dentro del metadata del servicio.

A diferencia del `BasicProfileBinding` que permitía una gran interoperabilidad a los servicios Web de primera generación estos nuevos `bindings` ofrecen entre otras novedades, seguridad y confiabilidad, el transporte se lo puede hacer por HTTP o HTTPS, los mensajes cuentan con cifrado de texto, WCF provee cerca de 23 de estos nuevos tipos de `bindings`; algunos de ellos son:

- `BasicHttpBinding`, configura un servicio para trabajar con atributos y servicios que se ajustan al perfil básico.
- `WsHttpBinding`, procura brindar fiabilidad y seguridad, codifica el mensaje con XML y transporte con HTTP.
- `WsDualHttpBinding`, brinda una comunicación dúplex a través de SOAP.
- `WebHttpBinding`, permite configurar extremos de los servicios para responder a HTTP en lugar de SOAP.
- `mexHttpsBinding`, configura un enlace para el enlace para que el intercambio de mensajes se realice sobre HTTPS.

En la Figura 3-63 se muestra un fragmento de código del `WebHttpBinding` usado en el servicio para transferencia de archivos.

```

<bindings>
  <webHttpBinding>
    <binding name="RestBinding" transferMode="Streamed"
      bufferSize="65536" maxReceivedMessageSize="67108864">
      <security mode="None">
        <transport clientCredentialType="None"/>
      </security>
    </binding>
  </webHttpBinding>
</bindings>

```

Figura 3-63 WebHttpBinding usado en el servicio para transferencia de archivos

3.4.3. CAPA PROCESOS DE NEGOCIO

Otra promesa de SOA es poder crear aplicaciones a partir de servicios existentes. Al ser componentes los servicios Web pueden verse como cajas negras con la capacidad de ser utilizadas y reutilizadas sin necesidad de conocer su funcionalidad [50].

En el presente proyecto a más del uso de servicios propios, se utiliza servicios ya existentes; uno de la página web de la SENESCYT y otro que es el servicio de geocodificación de Google Maps.

A continuación se describen todas las validaciones que realiza el Servicio Web a través de diagramas de secuencia desarrollados en función del tiempo, el cual fluye hacia abajo del diagrama indicando instancias y eventos en lugar de clases

3.4.3.1. Validación de RUC

En la Figura 3-64 a través del diagrama de secuencia se describe el proceso de validación del RUC, en el cual el concesionario que en el diagrama representa al usuario, ingresa el RUC en la aplicación de escritorio la cual a través del evento Click consulta al servicio el cual, solo si existe el RUC en la base de datos devuelva el nombre, la dirección y el teléfono de la empresa, además del nombre del representante legal y el correo del representante legal. Como se mencionó anteriormente se retorna la información concerniente al RUC en formato JSON.

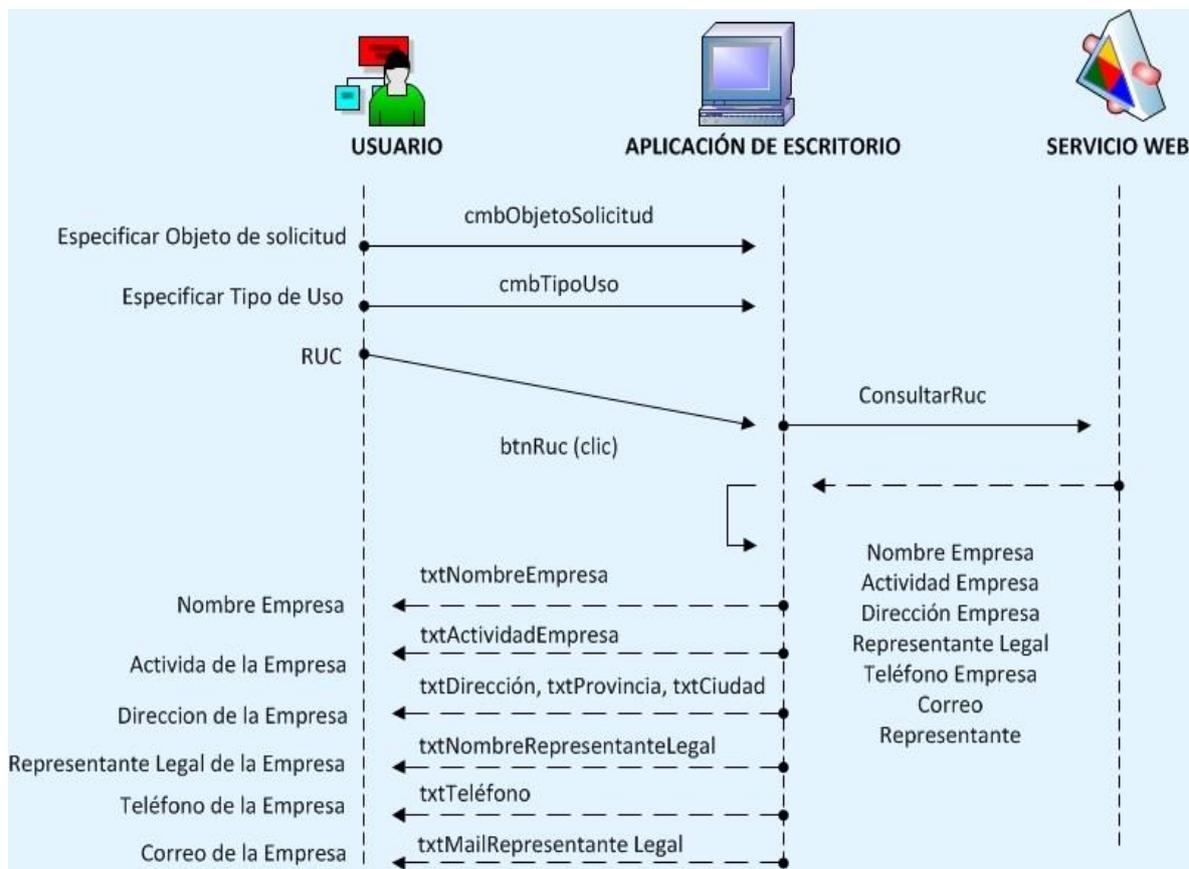


Figura 3-64 Diagrama de secuencia RUC

3.4.3.2. Validación de antenas

En la Figura 3-65 a través del diagrama de secuencia se describe el proceso de validación de antenas, en el cual el concesionario que en el diagrama representa al usuario, ingresa la marca de la antena, a través del evento `Click` consulta al servicio el cual sólo si existe la marca en la base de datos retorna los modelos, y de la misma manera al elegir el modelo devuelve la ganancia de la antena.

Como se mencionó se retorna la información concerniente a la antena en formato JSON.

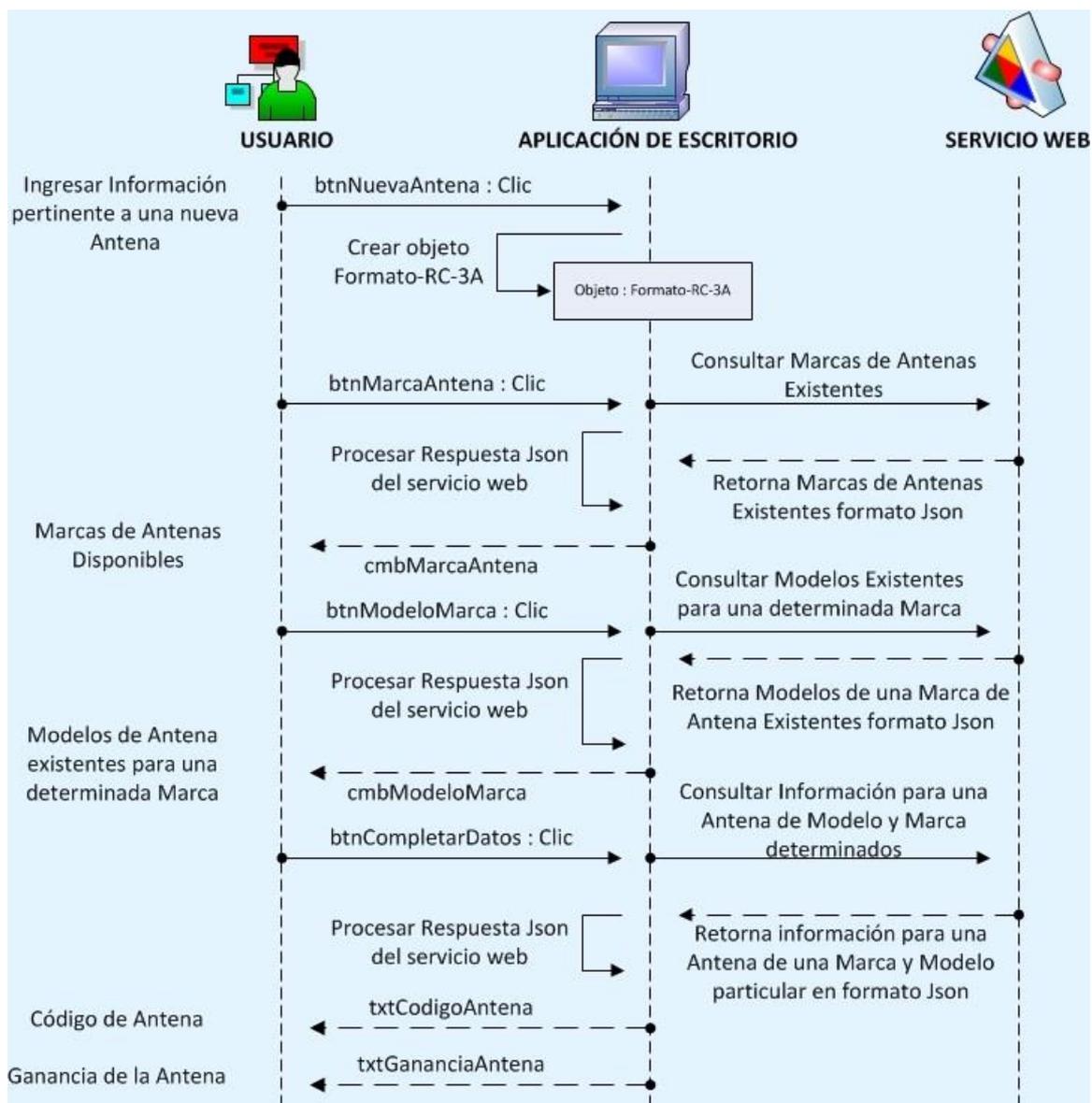


Figura 3-65 Diagrama de secuencia Antenas

3.4.3.3. Validación de equipos

En la Figura 3-66 a través del diagrama de secuencia se describe el proceso de validación de equipos, en el cual el concesionario que en el diagrama representa al usuario, ingresa la marca del equipo en la aplicación de escritorio la cual a través del evento `Click` consulta al servicio, el cual sólo si existe la marca en la base de datos retorna los modelos. Como se mencionó se retorna la información concerniente al equipo en formato JSON.

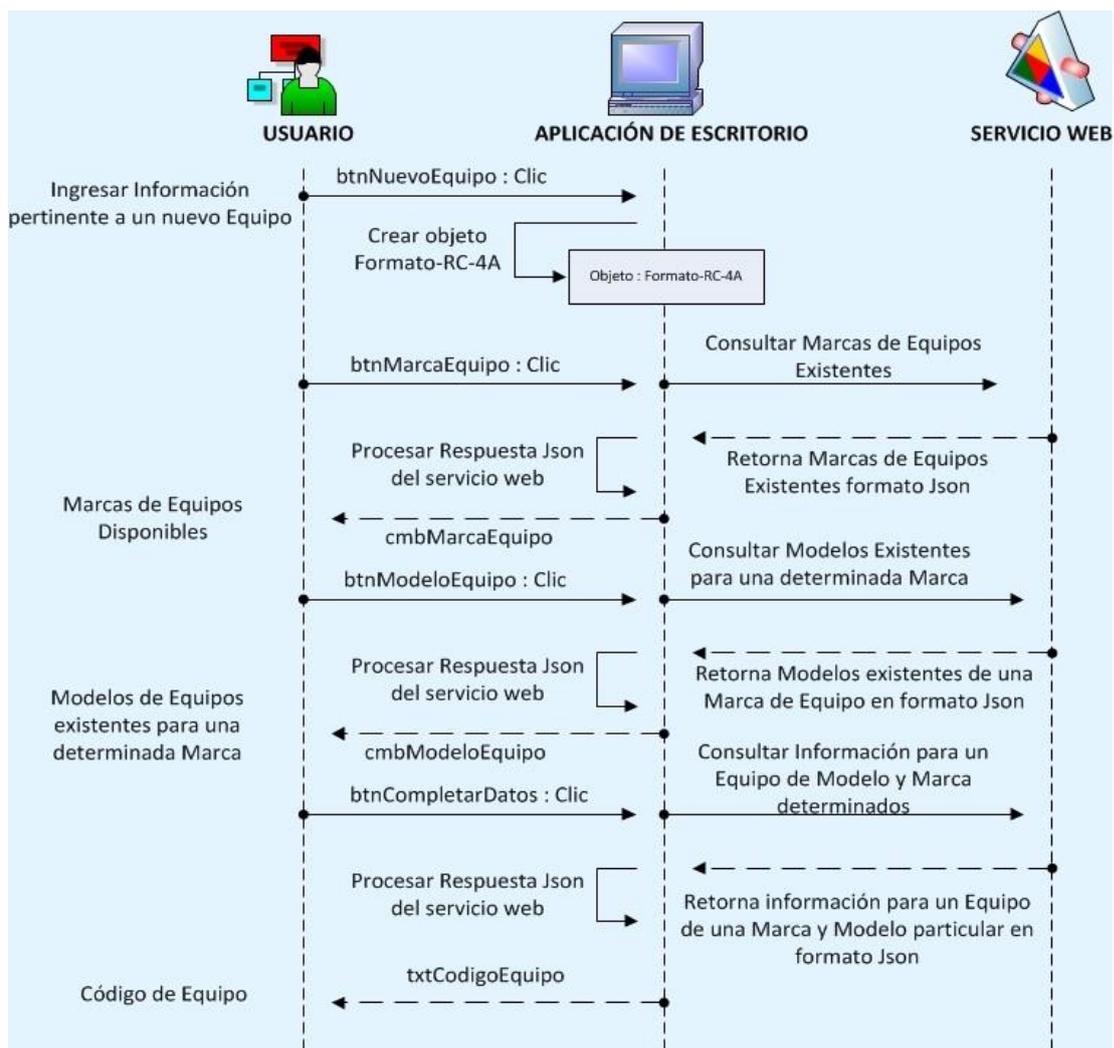


Figura 3-66 Diagrama de secuencia Equipos

3.4.3.4. Validación Profesional Técnico

En la Figura 3-67 el diagrama de secuencia describe el proceso de validación del título del profesional técnico que elabora el trámite, para lo cual se deberá ingresar el número de cédula de identidad en la aplicación de escritorio, la cual a través del evento `Click` del botón realizará la consulta al servicio web, el que a su vez solicitará la información a la página web del SENESCYT la cual responderá con la información en formato HTML al servicio web. El servicio Web procesará la información y la enviará a la aplicación de escritorio.

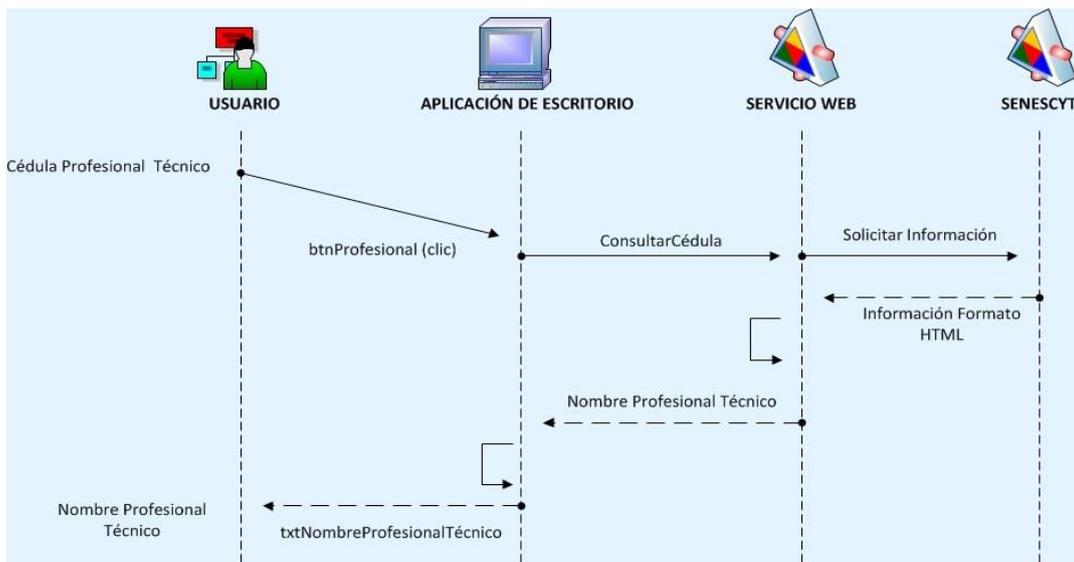


Figura 3-67 Diagrama de secuencia de la validación del Profesional Técnico

3.4.3.5. Validación de Coordenadas Geográficas

En la Figura 3-68 el diagrama de secuencia describe el proceso de validación de coordenadas geográficas para lo cual se deberá ingresar las coordenadas en la aplicación de escritorio, a través del evento Click del botón validar realizará la consulta al servicio Web, este a su vez solicitará la información al servicio de geocodificación de Google Maps.

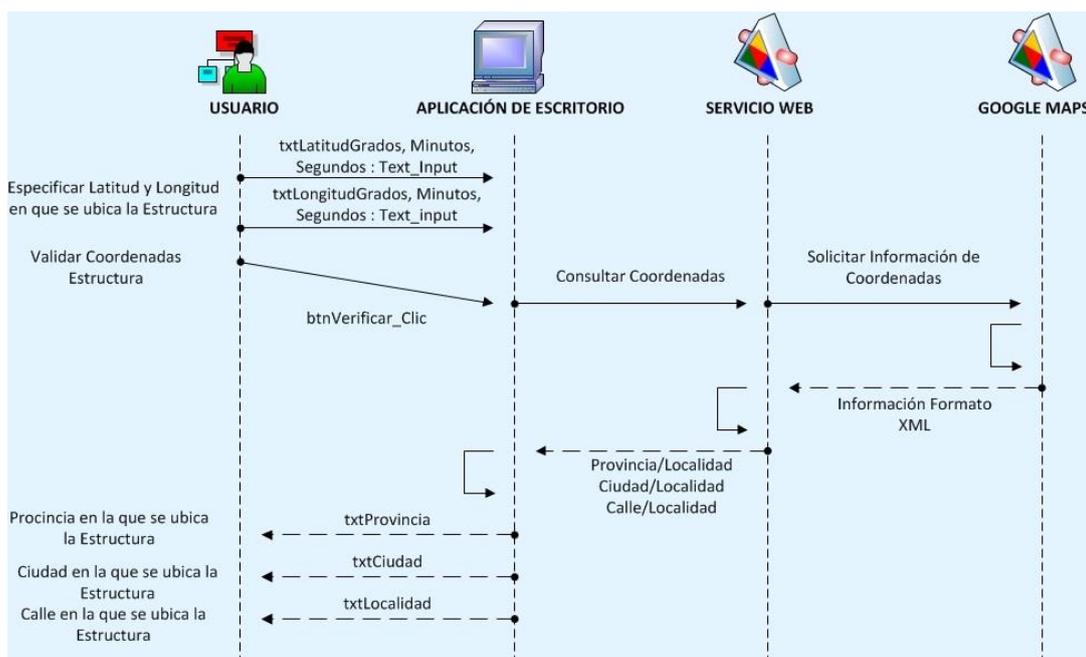


Figura 3-68 Diagrama de secuencia de la validación de Coordenadas Geográficas

El servicio de Google Maps responderá con la información respecto a la ubicación de las coordenadas en formato XML al servicio web, el cual procesará la información y la enviará a la aplicación de escritorio.

Como se mencionó antes el servicio de Google Maps también permite visualizar en el mapa la ubicación de las coordenadas ingresadas.

3.4.4. CAPA DE DATOS

Finalmente se encuentra la Capa de Datos o Persistencia la que contiene los objetos que reciben la información de la base de datos. Como se mencionó para el desarrollo del prototipo no se utilizó una base de datos sino un conjunto de tablas proporcionadas por la SENATEL, las que permitieron realizar las consultas necesarias para la validación del RUC, antenas y equipos.

En la Figura 3-69, la Figura 3-72 y la Figura 3-73 se muestran los diagramas de colaboración de cada una de las consultas que se realizan a las tablas.

En los diagramas de colaboración a diferencia de los diagramas de secuencia no se desarrollan en función del tiempo, sino más bien muestran las instancias específicas de las clases desde el punto de vista del espacio.

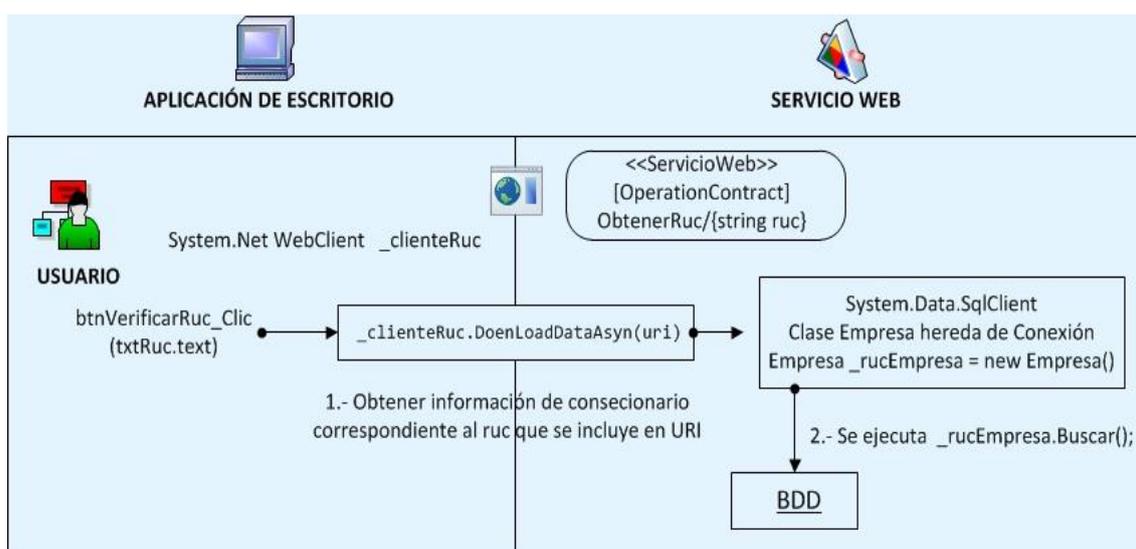


Figura 3-69 Diagrama de colaboración del RUC

En la Figura 3-70 se presenta un fragmento del código que corresponde a las interacciones descritas en el anterior diagrama de colaboración, inicialmente en el evento clic del botón `bntVerificarRuc`, valida que la cantidad de dígitos del `ruc` sea la adecuada es decir trece, si esta condición se cumple puede estructurar una petición o URI, que mediante un objeto `WebClient` y de manera asincrónica obtiene la información del servicio.

```
private void BtnRucClick(object sender, RoutedEventArgs e)
{
    int i;
    i = txtRuc.SelectionStart;
    if (txtRuc.Text == string.Empty)
    {
        MessageBox.Show("El campo de Ruc debe ser llenado", "Advertencia RUC", MessageBoxButton.OK, MessageBoxImage.Error );
    }
    else if (i < 13)
    {
        MessageBox.Show("El campo de Ruc debe contener 13 dígitos", "Advertencia RUC", MessageBoxButton.OK, MessageBoxImage.Error);
    }
    else if (i == 13)
    {
        try
        {
            String URI = "http://" + direccionIP + "/ServicioSenatel.svc/ObtenerRuc/" + txtRuc.Text;
            _clienteRuc.DownloadDataAsync(new Uri(URI));
        }
        catch (Exception ex)
        {
            Console.WriteLine("Exception: {0}", ex.ToString());
        }
    }
}
}
```

Figura 3-70 Consultar información del RUC al servicio

Por su parte el servicio tiene definido un método que recibe como parámetro de entrada el valor del RUC, con este valor hace uso de la clase `Conexion` y `Empresa` para realizar la consulta a la base de datos y retornar la información existente en un objeto `Data Contract`, de nombre `datosEmpresa` (Ver Figura 3-71).

```
EmpresaData datosEmpresa = new EmpresaData();
Empresa contenidoEmpresa = new Empresa();
//Obtener y devolver información asociada al RUC
public List<string> ObtenerRuc(string ruc)
{
    contenidoEmpresa.Ruc = ruc;
    List<string> empresa = new List<string>();
    if (contenidoEmpresa.Buscar() == true)
    {
        empresa = contenidoEmpresa.datosEmpresa;
    }
    else
    {
        datosEmpresa.AD_STREET = contenidoEmpresa.Mensaje.ToString();
    }
    return empresa;
}
```

Figura 3-71 Método en el Servicio que busca la información de un RUC

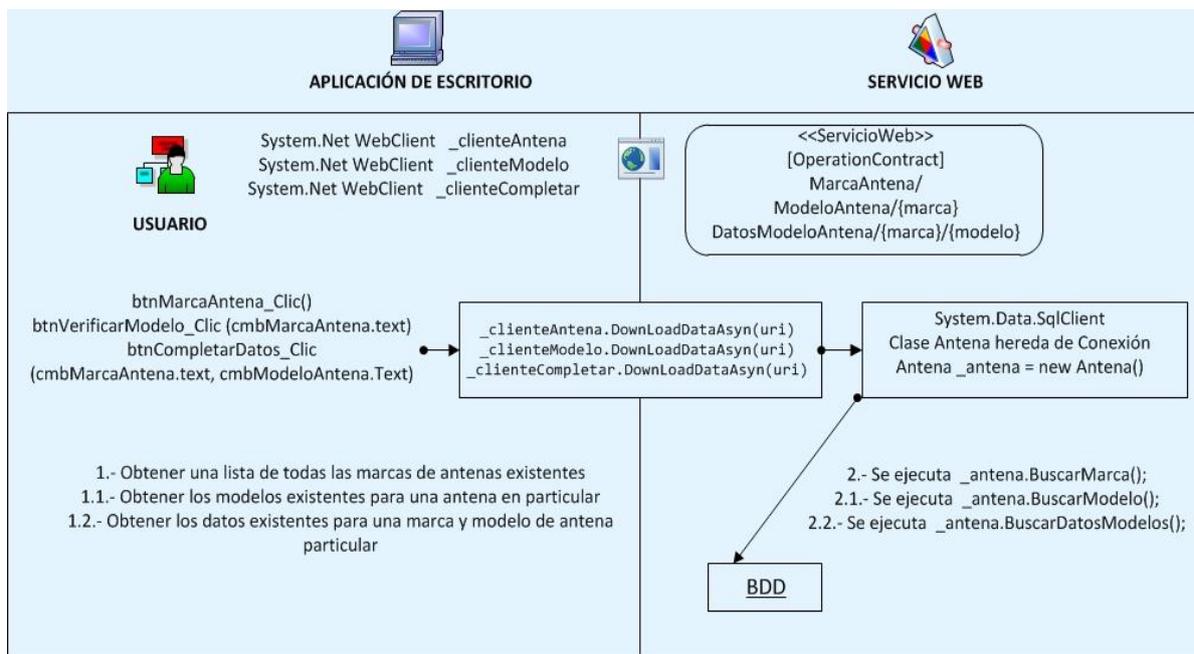


Figura 3-72 Diagrama de colaboración de la antena

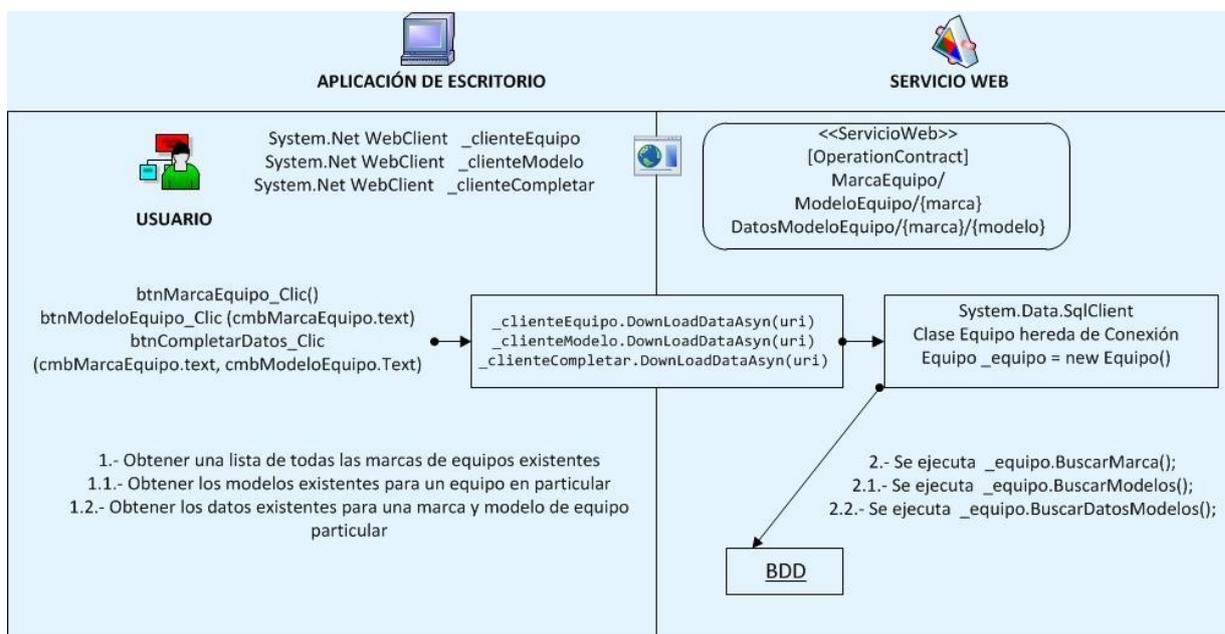


Figura 3-73 Diagrama de colaboración del equipo

El servicio cuenta con dos métodos similares que permiten las operaciones que se describen en los anteriores diagramas de colaboración como se puede apreciar en

la Figura 3-74, estos métodos reciben como parámetro tanto la marca como el modelo ya sea de un equipo o una antena y posteriormente haciendo uso de las clases Conexion, Equipo y Antena respectivamente, y se realizan las consultas necesarias para devolver la información que el usuario requiere.

```
//Obtener y devolver los datos correspondientes a una marca y modelo específico de antena
public List<String> DatosModeloAntena(string marca, string modelo)
{
    List<String> DatosModeloAntena = new List<String>();
    contenidoAntena.MARCA = marca;
    contenidoAntena.MODELO = modelo;
    DatosModeloAntena = contenidoAntena.BuscarDatosModelos();
    return DatosModeloAntena;
}

//Obtener y devolver los datos correspondientes a una marca y modelo específico de equipo
public List<String> DatosModeloEquipo(string marca, string modelo)
{
    List<String> DatosModeloEquipo = new List<String>();
    contenidoEquipo.MARCA = marca;
    contenidoEquipo.MODELO = modelo;
    //if (contenidoAntena.BuscarDatosModelos())
    //{ DatosModelo.GANANCIA = contenidoAntena.GANANCIA; DatosModelo.IDENTIFICADOR = contenidoAntena.IDENTIFICADOR;
    DatosModeloEquipo = contenidoEquipo.BuscarDatosModelos();
    return DatosModeloEquipo;
}
```

Figura 3-74 Métodos para consultar modelos e Equipos y Antenas

CAPÍTULO IV

4. IMPLEMENTACIÓN, PRUEBAS Y SEGURIDAD

4.1. INTRODUCCIÓN

En este capítulo se describe el despliegue de los componentes del Sistema Distribuido, analizando los requerimientos para conseguir un funcionamiento adecuado.

Se describe la seguridad y tolerancia a fallos, a nivel físico y de aplicaciones.

Se proporciona una guía que describe el procedimiento adecuado para la utilización del sistema. Se realizan las pruebas con los tres componentes del Sistema Distribuido en operación, analizando la seguridad y tolerancia a fallos.

Finalmente se presentan los costos referenciales.

4.2. IMPLEMENTACIÓN

4.2.1. DESPLIEGUE DEL PROTOTIPO DE SISTEMA DISTRIBUIDO

Para el despliegue del Sistema Distribuido se utilizó una sola máquina en la cual se encuentran alojados los tres componentes del Sistema Distribuido, razón por la que las configuraciones del servicio Web y la aplicación web se publicaron directamente y sin necesidad de configurar dispositivos intermedios como *routers* y *switches*

Para describir el despliegue de los tres componentes del Sistema Distribuido se utilizaron diagramas de despliegue, este tipo de diagramas permiten visualizar la arquitectura del sistema desde el punto de vista del despliegue, mostrando el hardware sobre el que se ejecutará el sistema y cómo el software se dispone en ese hardware.

Los diagramas de despliegue del prototipo muestran la configuración de los nodos⁴² que participan en la ejecución de los componentes⁴³ (los que residen dentro de los nodos).

La principal diferencia entre los componentes y los nodos es que los componentes son elementos que participan en la ejecución de un sistema, mientras que los nodos son los elementos en los que se ejecutan los componentes.

En la Figura 4-1 se muestran algunos de los componentes del prototipo de Sistema Distribuido

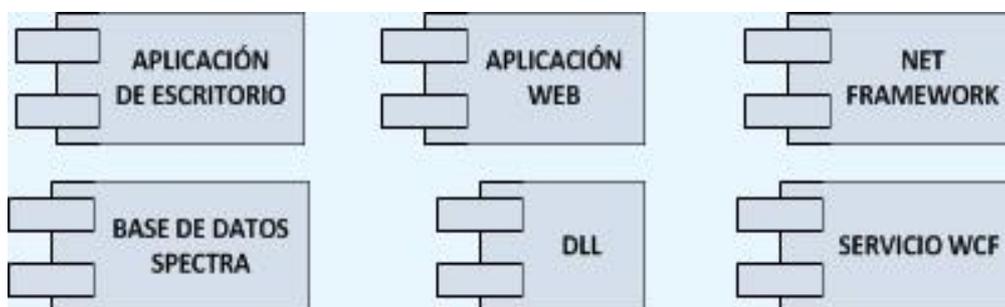


Figura 4-1 Algunos componentes presentes en el prototipo

A continuación se describen las especificaciones a tener en cuenta para la implementación de la aplicación de escritorio, el servicio Web y la aplicación web en la SENATEL, además de las configuraciones realizadas para el despliegue de los tres componentes.

4.2.1.1. Despliegue de la Aplicación de Escritorio

El concesionario deberá descargar e instalar la aplicación de escritorio de la página web de la SENATEL en las máquinas que se designen para realizar los trámites para el otorgamiento de títulos habilitantes o registro de enlaces, para los dos servicios tratados en este proyecto.

⁴² **Nodo:** Un nodo es a un objeto físico en tiempo de ejecución, que representa un recurso computacional, como por ejemplo los recursos de procesamiento y de memoria de un computador.

⁴³ **Componentes:** Los componentes mejor conocidos como artefactos, representan elementos concretos que son resultado del proceso de desarrollo, por ejemplo, archivos de configuración, bibliotecas, ejecutables, esquemas de base de datos, etc.

Los principales requerimientos de la máquina para esta aplicación se describen en la Tabla 4-1.

Componente	Requisito
Procesador	Mínimo: 1 GHz Recomendado: 2 GHz Óptimo: 3 GHz o más
Memoria	Mínimo: 512 MB de RAM Recomendado: 1 GB de RAM Óptimo: 2 GB de RAM o más
Espacio en disco disponible	Mínimo: 100 MB Recomendado: 1 GB Óptimo: 2 GB o más
Sistema Operativo	Windows 8 (x86/x64), Windows 7 (x86/x64), Windows Vista (x86/x64).

Tabla 4-1 Requerimientos para la máquina que use la aplicación de escritorio

El despliegue de la aplicación de escritorio se la realizó en una computadora con el sistema operativo Windows 7 de 64 bits, procesador de 2.20 GHz y 6 GB de RAM.

En la Figura 4-2 se muestra el diagrama de despliegue de la aplicación de escritorio, como se puede visualizar el concesionario debe considerar dos componentes adicionales a la aplicación de escritorio.

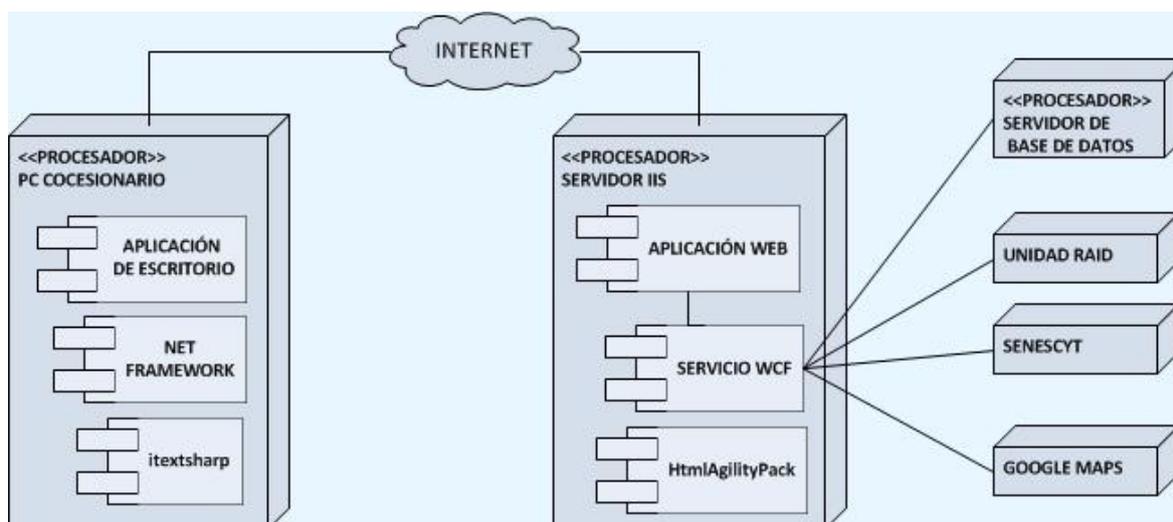


Figura 4-2 Diagrama de despliegue aplicación de escritorio

Estos componentes claves en son el **Net Framework** y **DLL**, que proveen de soluciones o características específicas requeridas por la aplicación de escritorio

para un correcto funcionamiento, al no incluir todos estos elementos en la aplicación desarrollada se consigue que la misma sea más liviana.

Net Framework.- se instala sobre sistemas operativos Windows, permite la ejecución de aplicaciones desarrolladas en la plataforma .NET

DLL.- biblioteca de enlace dinámico por sus siglas en inglés, son archivos que contienen fragmentos de código ejecutable, el mismo que se carga bajo demanda de algún programa y la administra el sistema operativo.

4.2.1.2. Despliegue del Servicio y Aplicación Web

En cuanto al despliegue en la SENATEL, la máquina del analista deberá contar con un navegador que le permita acceder mediante HTTPS a la aplicación web alojada en el servidor IIS. El servidor ubicado en la SENATEL deberá ser configurado para permitir la característica IIS, alojar tanto al servicio Web como a la aplicación web y tener las características que se detallan en la Tabla 4-2.

Componente	Requisito
Procesador	Mínimo: 2 núcleos Recomendado: 4 núcleos Óptimo: 5 núcleos o más
Memoria	Mínimo: 4 GB de RAM Recomendado: 6 GB de RAM Óptimo: 7 GB de RAM o más
Espacio en disco disponible	Mínimo: 50 GB Recomendado: 80 GB Óptimo: 100 GB o más
Sistema Operativo	Windows Server 2008 (x86/x64), en adelante.

Tabla 4-2 Requerimientos del servidor

Para el despliegue tanto del servicio como de la aplicación web, en el presente Proyecto se usó una máquina virtual Windows Server 2008 R2, en la cual se realizaron todas las configuraciones necesarias para alojar a los dos componentes.

En la Figura 4-3 se muestra el diagrama despliegue del servicio Web y de la aplicación web.

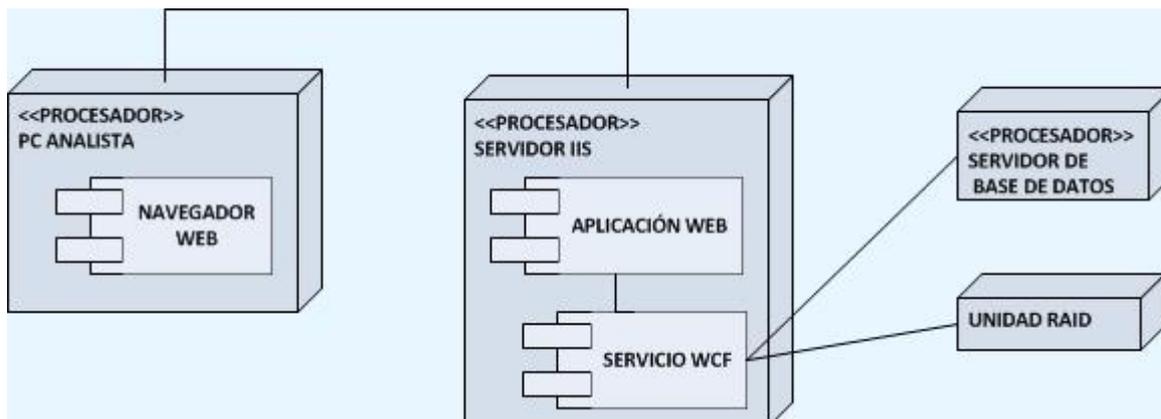


Figura 4-3 Diagrama de despliegue servicio y aplicación web

A continuación se describen los pasos para realizar el despliegue.

Para iniciar se debe agregar el rol o característica al servidor Web IIS (Ver Figura 4-4).



Figura 4-4 Agregar un Rol o característica al servidor

El sistema operativo del servidor también debe contar con el Net Framework adecuado para que permita la ejecución tanto del servicio Web y la aplicación web (Ver Figura 4-5).

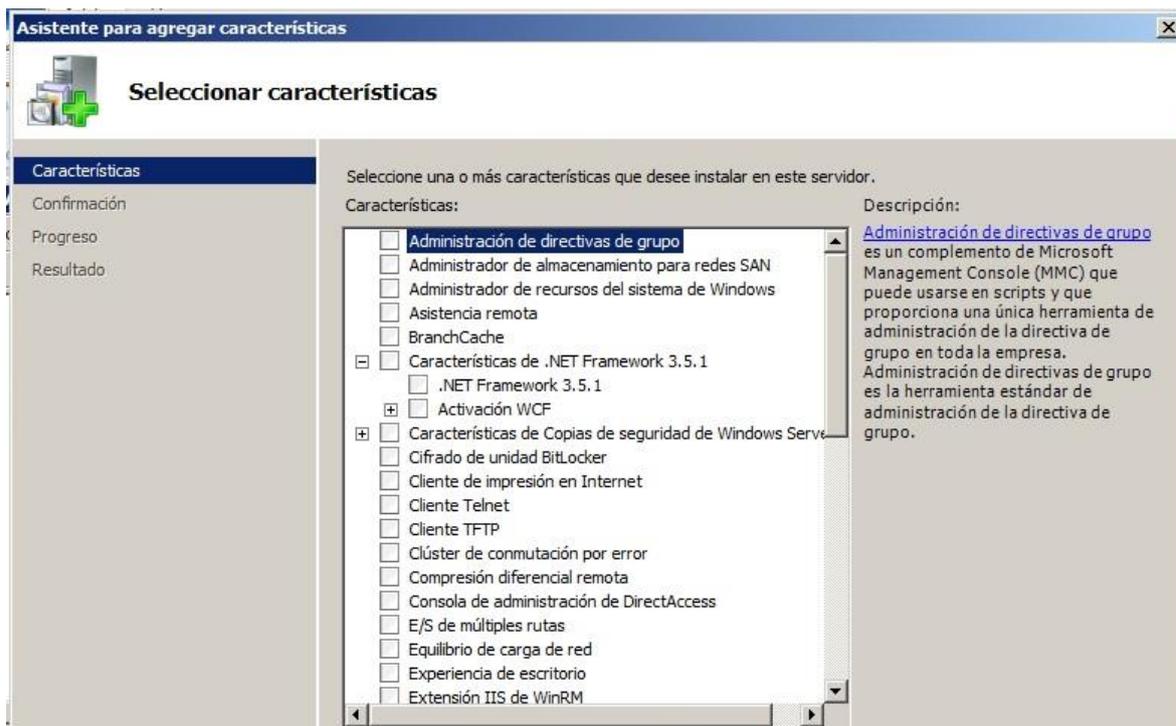


Figura 4-5 Agregar Net Framework al Servidor

El servidor Windows Server 2008 R2 utilizado, por defecto muestra la opción de un Net Framework versión 3.5 (Ver Figura 4-6).

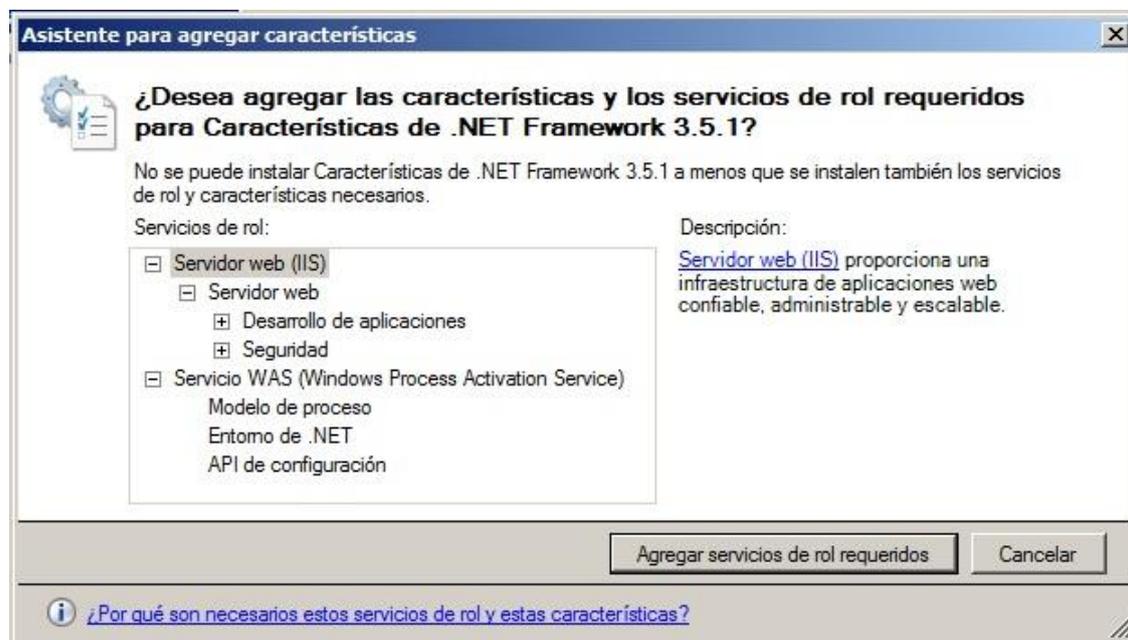


Figura 4-6 Proceso de instalación de la característica IIS y Net Framework

Aun cuando el proceso de instalación indica que se efectuó de forma exitosa, se debe agregar la característica de Net Framework por medio del *PowerShell* (Ver Figura 4-7).



```

Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. Reservados todos los derechos.

PS C:\Users\Administrador> cd ..
PS C:\Users> cd ..
PS C:\> cd Windows
PS C:\Windows> cd system32
PS C:\Windows\system32> Import-Module ServerManager
PS C:\Windows\system32> Add-WindowsFeature as-net-framework

Success Restart Needed Exit Code Feature Result
-----
True      No                Success    <.NET Framework 3.5.1>

PS C:\Windows\system32> _
  
```

Figura 4-7 Agregar Net Framework desde *PowerShell*

Después de realizar todo este proceso puede subsistir algún error, como por ejemplo que el Administrador de Internet Information Services no reconozca la versión de Net Framework en la que está desarrollada una aplicación a la que brinda alojamiento (Ver Figura 4-8).



Figura 4-8 Versión de Net Framework Incompatible

Para solucionar este problema se debe abrir el Administrador de Internet Information Services, ubicar el cursor en el nombre del servidor que se encuentra

en la parte superior izquierda, desplegar los submenús que contiene, ubicar la opción Grupo de Aplicaciones, en la ventana de la derecha aparecerán varias opciones entre las que se encuentra DefaultAppPool y como se puede evidenciar la versión de Net Framework que reconoce es la 2.0, se debe modificar este parámetro para corregir el problema (Ver Figura 4-9).



Figura 4-9 DefaultAppPool en el Administrador de Internet Information Services

Finalmente al hacer clic en la opción Modificación Básica, aparecerá una ventana con las versiones de Net Framework disponibles para elegir, como se muestra en la Figura 4-10 se escogió la versión más reciente.



Figura 4-10 Versiones de Net Framework disponibles.

Después de configurado el servidor, se puede acceder a los servicios y aplicaciones a través de su dirección IP.

4.2.2. INSTRUCTIVOS

Para el presente proyecto se desarrollaron instructivos con el fin de dar a conocer las funcionalidades de las aplicaciones y su respectivo uso.

En el Anexo D y Anexo E se encuentran los instructivos, tanto para Sistemas MDBA como para Sistemas MT-RDV.

Además la aplicación de escritorio cuenta con la opción de descarga de los Instructivos de uso como lo indica la Figura 4-11.



Figura 4-11 Descarga de Instructivos desde la aplicación de escritorio

4.3. SEGURIDAD Y TOLERANCIA A FALLOS

Ninguna aplicación informática está exenta de sufrir algún tipo de amenaza en contra de su confidencialidad⁴⁴, integridad⁴⁵ o disponibilidad⁴⁶, razón por la cual se debe priorizar esfuerzos para la protección física y lógica de los componentes de la infraestructura de la red.

Cuando se habla de seguridad se hace referencia no solo a la seguridad a nivel de software sino también a la seguridad física, razón por la cual a continuación se analizan los tipos de seguridades a tomarse en cuenta.

⁴⁴ **Confidencialidad**, La prevención de revelación de información no autorizada.

⁴⁵ **Integridad**, La prevención de modificación errónea de información.

⁴⁶ **Disponibilidad**, La prevención de retención no autorizada de información o recursos.

4.3.1. SEGURIDAD Y TOLERANCIA A FALLOS A NIVEL DE SOFTWARE

A continuación se detallan las seguridades que se maneja en los componentes del prototipo.

4.3.1.1. Recuperación de la información

En el presente Proyecto se garantiza que las aplicaciones que conforman el sistema reaccionen de manera adecuada ante ciertos fallos, la aplicación de escritorio permite recuperar información en caso de un cierre inesperado, para lo cual se implementó un mecanismo que permite guardar cada cierto tiempo la información ingresada por el concesionario, cabe aclarar que no es posible ejecutar el mecanismo todo el tiempo por lo cual habrán ciertos datos que puedan perderse (datos que fueron creados después de que el mecanismo fue ejecutado), sin embargo se garantiza que algún porcentaje de su trabajo se respaldará ante un fallo inesperado.

Existen dos clases que permiten desarrollar esta funcionalidad en la aplicación:

- `System.Threading.Timer`
- `System.Windows.Threading.DispatcherTimer`

Las dos clases son opciones viables para conseguir el objetivo propuesto, antes de elegir una de las opciones debemos tomar en cuenta dos aspectos importantes, la aplicación que se desarrolla es de tipo WPF y además la interfaz de usuario actualiza sus valores o en este caso el usuario podría modificar los valores que contiene la aplicación.

Tomando en cuenta los aspectos mencionados anteriormente se puede mencionar algunas características principales de estas clases que permitan elegir una opción.

`System.Threading.Timer`, es una clase de temporizador que se activa en un subproceso independiente. Recomendado para intervalos de tiempo en los que no se está tratando de actualizar la interfaz de usuario.

`System.Windows.Threading.DispatcherTimer`, es el mecanismo de tiempo de WPF, se maneja de forma simultánea con la interfaz de usuario (esto no se limita a un solo hilo - cada hilo tiene su propio Dispatcher) y está utilizando WPF.

La opción elegida para implementar la funcionalidad antes descrita en la aplicación de escritorio fue `System.Windows.Threading.DispatcherTimer`, teniendo en cuenta los siguientes aspectos:

- Un temporizador no debe ser motivo para que se bloquee la interfaz de usuario.
- La interfaz de programación de `System.Threading.Timer` no es coherente con las otras clases de temporizadores es un poco más engorroso.
- Al utilizar `System.Threading.Timer`, las operaciones que requieren acceso a los controles de interfaz de usuario deben hacerlo utilizando métodos `Invoke` o `BeginInvoke` del control.

Para comenzar con la implementación se debe iniciar por instanciar un objeto de la clase `System.Windows.Threading.DispatcherTimer` (Ver Figura 4-12).

```
DispatcherTimer dispatcherTimer = new System.Windows.Threading.DispatcherTimer();
```

Figura 4-12 Instanciar `System.Windows.Threading.DispatcherTimer`

Después se debe elegir el evento adecuado del formulario para inicializar el `DispatcherTimer`, en este caso se ha optado por el evento `Loaded` (Ver Figura 4-13).

```
private void WindowLoaded1(object sender, RoutedEventArgs e)
{
    dispatcherTimer.Tick += new EventHandler(SalvarBinMovTerr);
    dispatcherTimer.Interval = new TimeSpan(0, 3, 0);
    dispatcherTimer.Start();
}
```

Figura 4-13 Configurar eventos `DispatcherTimer`

Además se debe definir el evento `Tick`, la propiedad `Interval` y usar el método `Start` para inicializar el `Dispatcher`.

- `Tick`, es el evento o acción que toma efecto una vez que ha transcurrido el intervalo de tiempo, en este caso recibe como parámetro de entrada la función que se encarga de guardar la información existente en la aplicación como un archivo binario.
- `Interval`, el valor predeterminado es `00:00:00`, las posiciones corresponden a la de un reloj digital, de izquierda a derecha, segundos, minutos, horas respectivamente. También permite indicar valores de días (primer valor extremo izquierdo) y milisegundos (valor final extremo derecho).
- `Start`, método que inicializa el `Dispatcher`.

Finalmente es conveniente indicar que podría suscitarse un inconveniente con el consumo de procesador si la propiedad `Interval` no está bien configurada.

En la Figura 4-14 se muestra la función que permite guardar cada uno de los formularios en un archivo binario.

```
public static void SalvarBinMovTerr(object pathBin, EventArgs etime)
{
    PersistenciaBinaria almacenBin = new PersistenciaBinaria();
    almacenBin.Lista1A = Persistencia.Lista1A;
    almacenBin.Lista1B = Persistencia.Lista1B;
    almacenBin.Lista2A = Persistencia.Lista2A;
    almacenBin.Lista3A = Persistencia.Lista3A;
    almacenBin.Lista4A = Persistencia.Lista4A;
    almacenBin.Circuitos5A = Persistencia.Circuitos5A;
    almacenBin.Lista15A = Persistencia.Lista15A;
    string pathAlArchivo = @"F:\AVIS\MFT.bin";
    Stream streamArchivoBinario = File.Create(Convert.ToString(pathAlArchivo));
    BinaryFormatter serializador = new BinaryFormatter();
    serializador.Serialize(streamArchivoBinario, almacenBin);
    streamArchivoBinario.Close();
}
```

Figura 4-14 Función que permite la recuperación de la información

Para implementar esta funcionalidad o mecanismo de tolerancia a fallos se consultaron las siguientes referencias: [51] y [52].

4.3.1.2. Cuentas de usuario

La seguridad a nivel de software se refiere a seguir principios básicos que pueden evitar grandes daños, como el hecho de asignar cuentas de usuario y el mínimo privilegio a cada una de ellas.

En la aplicación web se manejan dos tipos de cuentas de usuario:

- **Concesionario:** la cual le permite la descarga de la aplicación de escritorio.
- **Analista:** la cual le permite la visualización de los trámites, el envío de correo y la generación del Memo.

En la Figura 4-15 se muestra la ventana de autenticación de la aplicación web y en la Figura 4-16 se muestra un fragmento de código de la autenticación de la aplicación web.

El código de la autenticación de la aplicación web, se encuentra disponible en el Anexo K.



ESCUELA POLITÉCNICA NACIONAL

FORMULARIOS EN LÍNEA

Secretaría Nacional de Telecomunicaciones

Ingrese su usuario y contraseña

Usuario:

Contraseña:

Tipo de Usuario:
Analista
Concesionario

Aceptar Cancelar

© 2014 EPN - SENATEL

Figura 4-15 Ventana principal de la aplicación web

```

public partial class Login : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    StreamReader _readerProfesional;
    readonly WebClient _clienteProfesional = new WebClient();
    readonly string _ipServidor = ConfigurationManager.AppSettings["ipServicio"];

    protected void BtnAceptarClick(object sender, EventArgs e)
    {
        if (txtUsuario.Text == string.Empty)
        {
            lblMensaje.Text = "Campo de usuario Vacio";
        }
        if (cmbTipoUsuario.SelectedIndex == 0)
        {
            lblMensaje.Text = "Debe seleccionar un tipo de usuario";
        }
        else
        {
            if (cmbTipoUsuario.SelectedIndex == 1)
            {
                try
                {
                    string URI = "http://" + _ipServidor + "/Serviciosenatel.svc/ValidarUsrInt/" + txtUsuario.Text + "/" + txtContraseña.Text;
                    byte[] validar = _clienteProfesional.DownloadData(new Uri(URI));
                    _readerProfesional = new StreamReader(new MemoryStream(validar));
                    string line;
                    while ((line = _readerProfesional.ReadLine()) != null)
                    {
                        string datoslimpios = RemoverCaracteres(line);
                        string[] param1 = datoslimpios.Split(',');
                        if (param1[0] == "T")
                        {
                            FormsAuthentication.SetAuthCookie(txtUsuario.Text, false);
                            Response.Redirect("/Default.aspx");
                        }
                        else
                        {
                            lblMensaje.Text = "Autenticación Incorrecta" + param1[1];
                        }
                    }
                }
                catch (Exception ex)
                {
                    lblMensaje.Text = ex.ToString() + "--" + "Usuario Incorrecto";
                }
            }
        }
    }
}

```

Figura 4-16 Fragmento de código de la autenticación de la aplicación web

4.3.1.3. MD5

El contenido que circula en internet no goza de gran confianza, puede ser modificado de tal manera que existe menor o mayor contenido que el original, sin conocer el archivo o la fuente original sería una tarea complicada tratar de evidenciar si el usuario cuenta con una versión completa o el archivo es corrupto.

Al hacer uso de algoritmos de seguridades se puede disminuir en un gran porcentaje esta incertidumbre respecto a la integridad y autenticidad de los mensajes y su contenido.

Como el presente Proyecto contempla la posibilidad de que el concesionario envíe un archivo XML, es imperativo el uso de algoritmos existentes para tratar de brindar seguridad al proceso de transferencia de archivos.

Se plantea entonces el uso de MD5 (Message-Digest 5), un algoritmo criptográfico de 128 bits muy popular, que se usa no solamente para comprobar la integridad de cualquier tipo de archivo, de tal forma que se pueda proporcionar la seguridad de

que el archivo transferido por el concesionario no sea alterado o modificado al navegar por la subred de comunicaciones.

Tanto en la aplicación de escritorio como en el servicio Web se definió una clase que permita calcular el código MD5 de un archivo, enviarlo junto con el archivo para que sea recalculado en el servidor y compararlo con el enviado por el concesionario, el archivo se recibe como parámetro por una de las tres funciones que posee la clase puede ser un `String`, `Stream` o la ubicación de un archivo en el computador, para después especificar qué algoritmo se desea aplicar al archivo, aunque no de manera explícita, se permitió además que se pueda elegir entre las variedades existentes de SHA, en caso de que el hash MD5 no fuese suficiente.

En la Figura 4-17 se visualiza la función implementada tanto en la aplicación de escritorio como en el servicio Web para el cálculo del hash. En la Figura 4-18 se visualiza la clase que implementa el Algoritmo MD5.

```
public class CalcularHash
{
    public FileStream fs;
    private string ByteArrayToString(byte[] ByteArray)
    {
        StringBuilder sb = new StringBuilder(ByteArray.Length);
        for (int i = 0; i < ByteArray.Length; i++)
        {
            sb.Append(ByteArray[i].ToString("X2"));
        }
        return sb.ToString();
    }

    private System.Security.Cryptography.HashAlgorithm GetHashProvider(EnumAlgoritmoHashs Alg)
    {
        switch (Alg)
        {
            case EnumAlgoritmoHashs.MD5:
                return new MD5CryptoServiceProvider();

            case EnumAlgoritmoHashs.SHA1:
                return new SHA1Managed();

            case EnumAlgoritmoHashs.SHA256:
                return new SHA256Managed();
        }
    }
}
```

Figura 4-17 Función para el cálculo del Hash

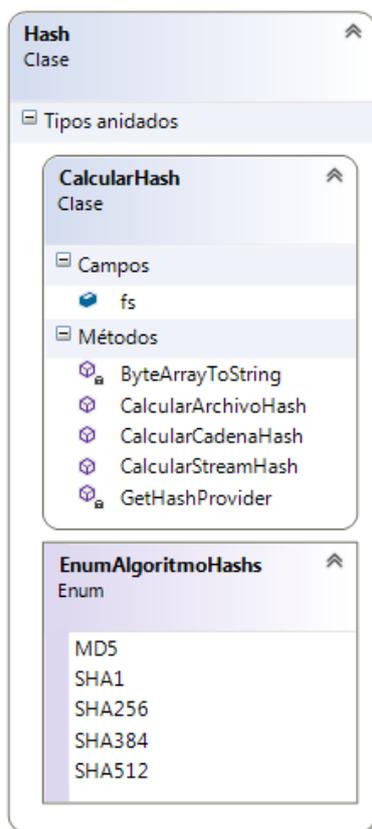


Figura 4-18 Clase que implementa el Algoritmo MD5

4.3.1.4. HTTPS

Como se mencionó en el Capítulo I, HTTPS es la versión segura del protocolo de transporte HTTP, el objetivo es convertir un canal inseguro en un medio seguro de transmisión, para ello hace uso de los protocolos de cifrado SSL y TLS.

Para que un servidor Web acepte conexiones HTTPS, el mismo debe contar con un certificado de clave pública.

En la Figura 4-19 se muestra la comunicación Cliente-Servidor mediante el protocolo HTTPS.

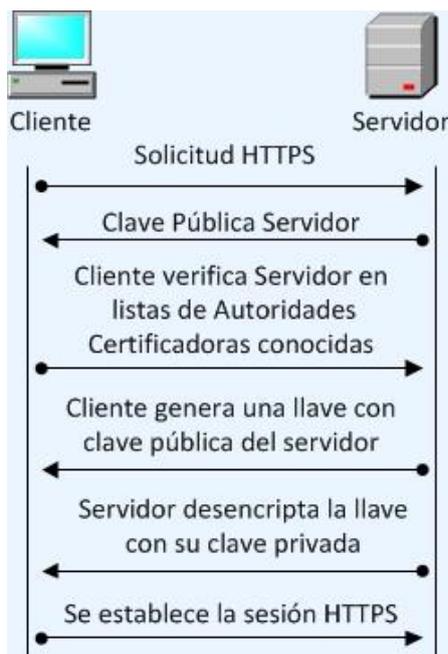


Figura 4-19 Comunicación Cliente-Servidor mediante protocolo HTTPS

En el Anexo F se muestra el procedimiento para crear un certificado autofirmado y cómo aplicarlo o enlazarlo a un sitio Web en la administración de IIS. En el Anexo G se presentan además las pruebas que verifican la funcionalidad de HTTPS.

4.3.1.4.1. Servicio WCF con HTTPS

Al tratar la seguridad de WCF, hay dos formas:

- **Seguridad de nivel de transporte;** no es más que la seguridad integrada por los protocolos, es el objetivo que interesa alcanzar.
- **Seguridad de nivel de mensaje;** se deben cifrar los datos, es decir, se introduce la seguridad en los datos.

El procedimiento para configurar la seguridad a nivel de protocolos en un servicio WCF RESTful es el siguiente [53], [54] y [55]:

1. Habilitar la seguridad del nivel de transporte en el archivo `web.config` del servicio.

La seguridad del transporte se habilita utilizando la etiqueta `<security>`, y el atributo `mode`, especificando este último como `Transport` como se muestra en el fragmento de código de la Figura 4-20.

```
<bindings>
  <webHttpBinding>
    <binding name="webHttpTransportSecurity">
      <security mode="Transport" />
    </binding>
  </webHttpBinding>
</bindings>
```

Figura 4-20 Seguridad de transporte `web.config`

2. Enlazar el binding y especificar la configuración de HTTPS

Se debe vincular el binding con el punto final, utilizando la etiqueta `bindingConfiguration` para detallar el nombre del binding, además de especificar la dirección donde se hospeda el servicio.

- ✓ Modificar el `mexHttpBinding` por `mexHttpsBinding`, de ser necesario agregando un nuevo `endpoint` (Ver Figura 4-21).

```
<endpoint address="mex"
  binding="mexHttpsBinding"
  contract="IMetadataExchange" />
```

Figura 4-21 Cambiar `mexHttpBinding` por `mexHttpsBinding`

- ✓ En `serviceMetadata` se debe cambiar `httpGetEnabled` por `httpsGetEnabled`, para indicar que ahora los métodos no se darán a conocer o podían ser accedidos por medio de HTTP, sino HTTPS (Ver Figura 4-22).

```
<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
```

Figura 4-22 Cambiar `httpGetEnabled` por `httpsGetEnabled`

Finalmente la Figura 4-23 muestra el Archivo `Web.config` alojado en el servidor.

```

Web: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <connectionStrings>
    <!--<add name="miStringdeConexion" connectionString="metadata=res://*/SPECTRA
    string=&quot;data source=ALBERTP-PC;initial catalog=SPECTRA;integrated secur
    <add name="miStringdeConexion"
    connectionString="Data Source=localhost;Initial Catalog=SPECTRA;
    User ID=avis;Password=A2014*;Packet Size=4096"
    providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <!--c:\Senatel-->
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
    <add key="ubicacion" value="f:\AVIS\XML\" />
    <add key="dirMdba" value="T:\MDBA\" />
    <add key="dirMft" value="C:\AVIS\MFT\" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5"/>
  </system.web>
  <system.serviceModel>
    <services>
      <service name="Serviciosenatel.Serviciosenatel"
        behaviorConfiguration="default">
        <endpoint address="" behaviorConfiguration="web"
          binding="webHttpBinding"
          bindingConfiguration="RestBinding"
          name="Serviciosenatel"
          contract="Serviciosenatel.IServiciosenatel" />
        <endpoint address="mex"
          binding="mexHttpsBinding"
          contract="IMetadataExchange" />
      </service>
    </services>
    <bindings>
      <webHttpBinding>
        <binding name="RestBinding" transferMode="Streamed" maxBufferSize="65536
          <!--<readerQuotas maxStringLength="5242880" maxArrayLength="163
          <security mode="Transport">
            <transport clientCredentialType="None" />
          </security>
        </binding>
      </webHttpBinding>
    </bindings>
  
```

Figura 4-23 Archivo `Web.config` alojado en el servidor

4.3.1.5. RAID 5

RAID 5 permite mínimo tres arreglos de discos y máximo cinco como lo muestra la Figura 4-24.

Según [56]: RAID 5 es recomendable para entornos de procesamiento de transacciones donde el nivel de entrada-salida y de lectura-escritura resultan intensos. Por esta razón se implementó este tipo o nivel de RAID a nivel de software en el presente Proyecto.

Cabe mencionar que en el servidor Windows Server 2008 se emplea RAID 5 a nivel de software.

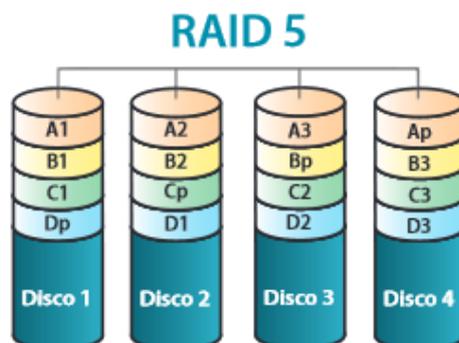


Figura 4-24 RAID 5 [56]

En el Anexo H se muestra la implementación de RAID 5 realizada en Windows Server 2008.

4.3.2. SEGURIDAD Y TOLERANCIA A FALLOS A NIVEL FÍSICO

En cuanto a la seguridad física, el prototipo contará con un UPS de baja gama, sin embargo es importante mencionar que las aplicaciones del prototipo estarán alojadas físicamente en las instalaciones de la SENATEL. Esta institución precautela por la seguridad interna y externa de los dispositivos informáticos especialmente de los servidores. Cuentan con varios servidores físicos debido a la gran cantidad de aplicaciones que se manejan, particularmente estos servidores permiten reducir el número de dispositivos y optimizar entornos virtualizados. Debido a la importancia de la información y de las aplicaciones que se manejan en la SENATEL, existen varias seguridades a nivel físico.

En la Tabla 4-3 se presentan una lista de algunas de las seguridades físicas tanto internas como externas.

SEGURIDAD INTERNA	SEGURIDAD EXTERNA
<ul style="list-style-type: none"> • Protección contra fuego • UPS (<i>LIBERT NFINITY</i>) • Tarjetas llave • Biométricas • Video vigilancia 	<ul style="list-style-type: none"> • Guardias de seguridad • Paredes • Video vigilancia • Señales • Sistemas de alarma • Detectores de intrusos

Tabla 4-3 Seguridades Físicas SENATEL

4.4. PRUEBAS

Antes de que cualquier software o aplicación entre en producción es necesario realizar pruebas con el objetivo de:

- Demostrar que la aplicación realiza las funciones para las cuales fue construida.
- Corregir posibles fallas o defectos para conseguir un funcionamiento adecuado y seguro.
- Realizar observaciones para futuras mejoras.

Por estas razones se realizó un manual de pruebas para que los ingenieros analistas de la DGGER y los ingenieros de la DGSI de la SENATEL realicen las pruebas necesarias en cada uno de los componentes del prototipo, documentando los resultados de las pruebas en los manuales y demostrando así que las aplicaciones puedan ser puestas en producción. En el Anexo I se presentan los manuales de pruebas tanto para Servicio MT-RDV como para Sistemas MDBA. Cabe mencionar que en este manual de pruebas constan todas las observaciones realizadas por los analistas de la DGGER como los ingenieros de la DGSI.

Al finalizar las pruebas se pudo constatar que se reduce el tiempo de elaboración, validación e ingreso de los trámites tanto de MDBA como de MT-RDV, ya que por una parte la aplicación de escritorio, realiza validaciones y cálculos matemáticos que reducen el tiempo de elaboración del trámite por parte del concesionario y por otra la aplicación web le permite al analista visualizar el trámite en una sola tabla e

importar el archivo en formato XML al SPECTRAplus, en lugar de digitar manualmente toda la información del trámite presentado por el concesionario.

En la Tabla 4-4 se puede apreciar ejemplos de tipos de trámites elaborados por un analista de DGER, sin usar las aplicaciones (TIEMPO ANTERIOR) y usando las aplicaciones (TIEMPO ACTUAL).

FUNCION	TIEMPO ANTERIOR	TIEMPO ACTUAL
ELABORACIÓN DE UN TRÁMITE MDBA CON 12 ENLACES PUNTO-MULTIPUNTO	Tiempo aproximado sin la aplicación de escritorio es: 120 minutos	Tiempo aproximado con la aplicación de escritorio: 60 minutos
VALIDACIÓN E INGRESO DE UN TRÁMITE MDBA CON 12 ENLACES PUNTO-MULTIPUNTO	El tiempo aproximado en validar e ingresar manualmente el trámite: 120 minutos	El tiempo aproximado en visualizar el trámite en la aplicación web e importarlo al SPECTRAplus: 50 minutos
ELABORACIÓN DE UN TRÁMITE MT-RDV CON 12 ESTACIONES	Tiempo aproximado sin la aplicación de escritorio es: 120 minutos	Tiempo aproximado con la aplicación de escritorio: 60 minutos
VALIDACIÓN E INGRESO DE UN TRÁMITE MT-RDV CON 12 ESTACIONES	El tiempo aproximado en validar e ingresar manualmente el trámite: 120 minutos	El tiempo aproximado en visualizar el trámite en la aplicación web e importarlo al SPECTRAplus: 40 minutos

Tabla 4-4 Comparación de tiempo de elaboración e ingreso de los trámites

4.5.COSTOS REFERENCIALES

4.5.1. Costos referenciales del hardware

Para la posible implementación del Prototipo se requiere de al menos: un computador para la instalación de la aplicación de escritorio y de un servidor en el que se pueda alojar el Servicio Web y la aplicación web. Además si se requiere de la implementación física de RAID 5 se necesita al menos tres discos duros.

En la Tabla 4-5 se detallan las características y costos del servidor, tomando en cuenta las características mencionadas en el despliegue del Servicio y aplicación web y que el servidor debe soportar la posible implementación física de RAID 5.

En la Tabla 4-6 se detallan las características y costos de los discos duros, tomando en cuenta que deberán implementarse en el servidor.

Distribuidor	IT3
Marca	HP
Modelo	HP ML350e Gen8 E5-2407 4LFF Base US Svr
Características	(1) Intel Xeon 4-Core E5-2407 (2.2GHz) / 10MB L3 cache / 12GB (3x4GB) PC3L-10600E / HP Ethernet 1Gb 2-port 361i Adapter / HP Smart Array B120i/512MB FBWC SATA Controller (RAID 0,1, 5, 10) / (4) LFF Hot Plug SATA HDD bahias / SATA DVD-ROM / (2) slots PCIe 3.0 y (2) slots PCIe 2.0 / (1) 460 Watt Power Supply no hot plug, no redundante / 2 No Hot Plug, No Redundante Fans / Teclado y mouse USB / HP iLO Management Engine / Torre (5U)
Costo	1344,00

Tabla 4-5 Características y costos del servidor

Distribuidor	IT3
Marca	HP
Características	DISCO DURO HP 600GB 6G SAS 10K 2.5in SC ENT HDD, PN: 652583-B21
Costo	235,20

Tabla 4-6 Características y costos de discos duros

En la Tabla 4-7 se detallan las características y costos del computador (*desktop*), tomando en cuenta las características mencionadas en el despliegue de la aplicación de escritorio.

Distribuidor	IT3
Marca	ASUS
Modelo <i>Mainboard</i>	MBO ASUS MAXIMUS VII RANGER LGA1150 4TA G Z97/USB3/SATA 6G
Características	INTEL Z97 CORE I5/ PENTIUM / CELERON (LGA1150) DDR3 3200(OC)/3100(OC) 4TA GENERACION DISCO DURO SEAGATE INTERNO 1TB SATAIII 6.0GB/S 7200RPM
Costo	1097,60

Tabla 4-7 Características y costos de computador

Como se puede apreciar solo se presentan una opción por cada equipo ya que no se pudo obtener cotizaciones de los mismos equipos en otras empresas. El costo referencial de hardware para la posible implementación sería de \$ 3147,20 tomando en cuenta la implantación física de RAID 5 con mínimo tres discos duros.

En el ANEXO J se encuentran las cotizaciones del hardware descrito.

4.5.2. Costos referenciales del Software Desarrollado

Para realizar estimaciones de costos en un proyecto de desarrollo de software es necesario considerar su tamaño, el esfuerzo comprometido, el tiempo, la tecnología de desarrollo y además agregar otros costos asociados (equipos, infraestructura, gastos administrativos, etc.). Hay que tener en cuenta que una mala estimación probablemente hará que el proyecto fracase.

Para comenzar con la estimación de costos se deberán definir las métricas⁴⁷ que se utilizarán para estimar el tamaño y costo de una aplicación, además del tiempo empleado y número de personas destinadas a trabajar en el proyecto.

En algunos casos se usa como métrica al número de líneas de código, particularmente para el desarrollo del prototipo no se la consideró como una métrica adecuada ya que se puede tener dos programas que realicen la misma funcionalidad pero que tengan diferente número de líneas de código, y no por ello se podría considerar que el programa con mayor número de líneas sea mejor, ya que depende en gran medida del lenguaje de programación además de la eficiencia o habilidad del programador para simplificar el código.

Existen varias técnicas de estimación de costos en las que intervienen muchas métricas para determinar el tiempo y el número de personas a emplearse en el proyecto. Debido a que el tiempo y el número de personas para el presente Proyecto ya fueron establecidos se optó por el desarrollo de una estimación práctica.

La métrica utilizada para el presente fue el número de horas empleadas en el desarrollo de cada una de las aplicaciones. Para determinar el tiempo se hizo uso de las historias de usuario en las cuales se detalla el tiempo utilizado en cada una de las funciones que cumplirán las aplicaciones.

El costo de la hora depende de la experiencia que tenga el desarrollador, a mayor experiencia y conocimiento mayor costo por hora. Para realizar un costo aproximado se tomó como referencia un costo de 10 dólares la hora, tomando en cuenta que los desarrolladores cuentan experiencia en desarrollo de aplicaciones y servicios Web. En la Tabla 4-8 se muestra el valor estimado del proyecto, cabe mencionar que el valor no tiene considerado rubros adicionales como: IVA (Impuesto al Valor Agregado), costos administrativos, de equipos, de tecnología de desarrollo, mantenimiento, entre otros.

⁴⁷ **Métrica** es una medida destinada a estimar el tamaño u otra característica de un software, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

APLICACIÓN	ACTIVIDAD O TAREA	TIEMPO ESTIMADO EN HORAS
APLICACIÓN DE ESCRITORIO MDBA	Creación de los Formularios (RC-1B, RC-2A, RC-3A, RC-4A, RC-9A, RC-9B y RC-15A)	180
	Validaciones locales en cada uno de los formularios	120
	Fórmulas Matemáticas	120
	Imprimir Formularios en PDF	30
	Guardar y reconstruir formularios (.bin)	30
	Generar XML importable al SPECTRAplus	70
APLICACIÓN WEB MDBA	Autenticación	30
	Crear y llenar tabla	50
	Generar Memo	20
SERVICIO WEB MDBA	Configuraciones	60
	SENECYT	30
	Google Maps	30
	BDD	30
APLICACIÓN DE ESCRITORIO MT-RDV	Creación de los Formularios (RC-1A, RC-2A, RC-3A, RC-4A, RC-9A, RC-9B y RC-15A)	180
	Validaciones locales en cada uno de los formularios	120
	Fórmulas Matemáticas	120
	Imprimir Formularios en PDF	30
	Guardar y reconstruir formularios (.bin)	30
	Generar XML importable al SPECTRAplus	70
APLICACIÓN WEB MT-RDV	Autenticación	30
	Crear y llenar tabla	50
	Generar Memo	20
SERVICIO WEB MT-RDV	Configuraciones	60
	SENECYT	30
	Google Maps	30
	BDD	30
SEGURIDAD Y TOLERANCIA A FALLOS		140
DESPLIEGUE DEL SISTEMA		100
PRUEBAS		150
REALIMENTACIÓN, CORRECCIONES Y MODIFICACIONES		200
TIEMPO TOTAL EN HORAS		2190
CONSIDERANDO EL RUBRO DE 10 DÓLARES LA HORA		\$ 21.900,00

Tabla 4-8 Costos Referenciales

4.5.3. Costos referenciales de licenciamiento

Como se mencionó en el Capítulo II los componentes del Sistema Distribuido se desarrollaron sobre Visual Studio 2012, el cual tiene un costo de licenciamiento al igual que el Sistema operativo Windows Server 2008 R2, sistema sobre el cual se alojaron el Servicio Web y la aplicación web.

La Tabla 4-9 muestra los costos de licenciamiento en el Ecuador proporcionados por la empresa AKROS CIA.LTDA., cabe mencionar que los costos varían en función de la ubicación geográfica. En el Anexo J se encuentra la cotización de las licencias mencionadas.

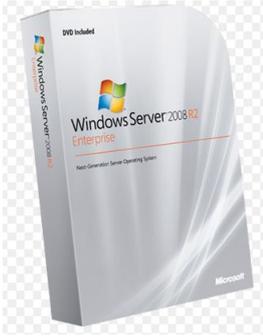
Cantidad	Código	Descripción	Valor Total
1	P71-07845	WinSvrDataCtr 2012R2 OLP NL Gov 2Proc Qlfd 	\$ 4,903.07
1	C5E-01141	VSPro 2012 OLP NL Gov 	\$ 396.52

Tabla 4-9 Costos referenciales Licenciamiento

La importancia de analizar el costo de licenciamiento tanto de las herramientas de desarrollo como de los sistemas operativos radica en la correcta estimación de los

costos del software a desarrollar. Sumando los costos de software, hardware y licenciamiento se obtiene un costo referencial de \$ 30.346,79.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

1. Al implementar una herramienta informática como la desarrollada en el presente Proyecto se puede reducir la cantidad de trabajo, agilizando los procesos de elaboración, validación e ingreso de los trámites, puesto que por una parte el concesionario ingresará un trámite con información validada a la SENATEL, y por la otra, el ingeniero analista de la SENATEL visualizará el trámite en una sola tabla para posteriormente ingresarlo al software SPECTRAplus, sin necesidad de transcribir manualmente toda la información técnica y legal del trámite ingresado por el concesionario. Cabe mencionar que mientras más experiencia se tenga con las aplicaciones se reducirá el tiempo empleado en la elaboración de trámites.
2. Pese a la automatización en el ingreso de los trámites elaborados por parte de los concesionarios, se debe considerar que el presente Proyecto no realiza la validación de firmas e imágenes ya que estaba fuera de su alcance, motivo por lo cual el concesionario aún deberá imprimir el trámite adjuntando las gráficas y formularios restantes, para presentarlos al CAU (Centro de Atención al Usuario) de la SENATEL.
3. La aplicación de escritorio realiza validaciones *online*, para lo cual el concesionario deberá tener conexión a Internet, sin embargo la aplicación le permite completar información que no requiere de conexión a Internet realizando validaciones locales.
4. Una de las principales funciones del Sistema desarrollado es la validación de campos existentes a través de la base de datos del SPECTRAplus, por lo cual antes de poner el prototipo en producción se deberá realizar una depuración de la base de datos con el fin de actualizar la información existente y eliminar posibles datos duplicados.

5. La aplicación de escritorio tiene la funcionalidad de validar que cada uno de los formularios se encuentren completos, y solo cuando todos los formulario estén validados se habilitará la opción del envío del trámite en formato XML a la SENATEL ya que si se envía un archivo en formato XML con datos incompletos o erróneos el archivo no se importa al SPECTRAplus. El concesionario tendrá la facilidad de visualizar el estado de cada uno de los formularios en su trámite, lo que le permitirá conocer si un formulario está: vacío, guardado o validado.
6. El Sistema Distribuido agilizará el proceso de ingreso de información al SPECTRAplus ya que el analista solo deberá realizar la importación del archivo XML, en lugar de ingresar los datos manualmente, reduciendo así los tiempos establecidos para estos procesos y cumpliendo con la norma ISO 9001-2008 que propone la mejora continua de los procesos.
7. Establecer el esquema adecuado basado en XML compatible con el software SPECTRAplus, para poder importar los datos generados a dicho software fue una tarea muy ardua, debido a que este software fuertemente acoplado maneja tipos de datos específicos y un orden particular en cada una de sus etiquetas.
8. La seguridad en las aplicaciones informáticas debe ser considerada como un aspecto prioritario en toda organización, precautelar la confidencialidad, disponibilidad e integridad de los datos es imperativo cuando se trabaja con información susceptible a ataques informáticos. El presente Proyecto maneja cuentas de usuario, recuperación de la información, manejo de HTTPS, entre otras.
9. El Sistema Distribuido maneja cierto nivel de tolerancia a fallos; haciendo uso de *time-outs*, llamadas asincrónicas y evitando la pérdida de información en caso de un cierre inesperado en la aplicación de escritorio, además de un servidor virtualizado con RAID 5 y un servidor de back-up offline que permitirá recuperar la información del servidor principal en caso de una falla grave.

10. La aplicación de escritorio no sólo valida la información menciona en el Plan de Proyecto de Titulación, realiza además validaciones de potencia y cálculos matemáticos. La implementación de fórmulas en la aplicación de escritorio permite reducir el tiempo en el cual el concesionario realizaba su trámite ya que evita que realice los cálculos matemáticos para la distancia, azimut y ángulo de elevación. Cabe mencionar que los resultados de cada uno de los cálculos fueron comprobados con los resultados que realiza el SPECTRAplus.
11. La información ingresada en la aplicación web puede ser guardada en formato binario, este tipo de archivo permite el ahorro de espacio físico para el almacenamiento de trámites.
12. Al existir una cantidad razonable de solicitudes de registros de enlaces para sistemas MDBA y sistemas Móvil Terrestre Radio de Dos Vías, se considera que el prototipo diseñado será de gran utilidad para el procesamiento de la información técnica y legal solicitada, así como también las validaciones consideradas, evitando que el concesionario remita información con errores impidiendo que el trámite deba ser devuelto por errores cometidos debido al incumplimiento de lo descrito en la Normas Técnicas.
13. El análisis de los procesos internos tanto de Servicio FM-RDV como de Sistemas MDBA permitió dilucidar que se requiere de una reestructuración en estos procesos, como por ejemplo evitar el ingreso de trámites a través del CAU, enviándolos en formato PDF a un repositorio en la SENATEL, de la misma manera que se envía el archivo en formato XML desde la aplicación de escritorio. Sin embargo este proceso solo se podría llevar a cabo cuando se implemente la validación de firmas, imágenes y formularios restantes.
14. El desarrollo del presente Proyecto se basó en SOA. Al desarrollar un sistema SOA, la mayor parte del esfuerzo y trabajo se enfocan en la consecución del servicio permitiendo el desarrollo de aplicaciones ágiles reutilizando código, de esta manera el prototipo de Sistema Distribuido usa servicios existentes para la validación de; coordenadas geográficas a través del servicio de Google Maps, y Título Profesional a través de la SENESCYT.

15. El Sistema Distribuido desarrollado permitió verificar que cumple con todas las especificaciones determinadas en los requerimientos, por tanto el mismo se encuentra listo para ser implementado en las instalaciones de la SENATEL a fin de realizar pruebas con los concesionarios y analistas.
16. El vertiginoso avance en las TIC (Tecnologías de la Información y Comunicación) conlleva al desarrollo de sistemas que permitan satisfacer nuevos requerimientos. Determinar el tipo de aplicaciones que se necesita es primordial, ya que no siempre es necesario realizar aplicaciones web en lugar de aplicaciones de escritorio y viceversa.

5.2.RECOMENDACIONES

- Generalmente en la elaboración de un software se requieren cambios durante su proceso por ello se recomienda el uso de metodologías que permitan la adecuada realimentación en los procesos como SCRUM, ya que al ser una metodología flexible y ágil, permite la realimentación de procesos, modificación de requerimientos, colaboración dinámica entre los miembros del equipo, flexibilidad en el desarrollo, entre otros.
- Se sugiere contar con versiones recientes de software ya que esto permite aplicar nuevas funcionalidades a las aplicaciones y obtener productos de calidad. Particularmente se recomienda el uso de Visual Studio 2012 ya que a la fecha es una herramienta que cuenta con una colección completa de servicios que le permite crear una gran variedad de aplicaciones. Se recomienda además el uso de TFS ya que permite gestionar todo el proceso de desarrollo independiente del lenguaje de programación.
- Para realizar el despliegue del sistema en la SENATEL se sugiere que:
 - I. Se aloje el Servicio Web en un servidor con las funcionalidades mínimas establecidas en el Capítulo IV.

- II. El Servicio Web deba conectarse a la base de datos del SPECTRAplus en lugar de a la base de datos de prueba que se manejó para el desarrollo del prototipo.
 - III. Se deban realizar las configuraciones necesarias para que la aplicación web funcione únicamente dentro de la Intranet de la Institución.
 - IV. Para el envío de correo electrónico al concesionario se deba cambiar la configuración SMTP, para permitir el envío desde el correo institucional en lugar de uno de GMAIL como se realizó en el prototipo.
- Se recomienda un profundo análisis y revisión de los formularios tanto de Servicio MT-RDV como de Sistemas MDBA, con el objetivo de ingresar información estrictamente necesaria y eliminar inconsistencias en la información como por ejemplo:
 - i. En el formulario RC-1A/1B debería ingresarse el Número de Registro del Título Profesional de la SENESCYT en lugar de la Licencia Profesional ya que actualmente no se usa.
 - ii. En el formulario RC-1A/1B debería cambiarse el campo denominado casilla por código postal.
 - iii. En el formulario RC-3A no se debería ingresar los cálculos de azimut ni ángulo de elevación, ya que estos valores no son propios de las antenas; estos valores debería ser calculados en los formularios RC-9A o RC-9B.
 - Se sugiere implementar en el prototipo la validación de las firmas e imágenes; para las firmas se recomienda el uso de firmas digitales, y para las imágenes, por ejemplo el caso de las radiaciones de las antenas, el uso de patrones que establezcan la validez de las radiaciones. Al implementarse la validación de imágenes y formularios se podría enviar el trámite digitalmente a la SENATEL.

- Se sugiere realizar la validación de la altura sobre el nivel del mar (s.n.m.) y del perfil topográfico utilizando el servicio de Google, de la misma manera como se realizó la validación de las coordenadas geográficas.
- Se recomienda que el prototipo de Sistema Distribuido gestione el estado de los trámites ingresados a la SENATEL por parte del concesionario; por ejemplo a través de un correo electrónico se podría enviar una notificación del estado del trámite, informándole así al concesionario si su trámite ha sido recibido satisfactoriamente, está siendo procesado, tiene algún error o ya ha sido realizado con éxito.
- Se sugiere que el Memorando que genera la aplicación web se enlace al Sistema de Gestión Documental QUIPUX, ya que esta herramienta gestiona la documentación digital o impresa, interna o externa de la SENATEL y de la mayoría de instituciones públicas.
- Se recomienda precautelar por la seguridad física y lógica tanto de las aplicaciones informáticas como de la seguridad que se utilicen en toda Institución pública o privada. Al ser la SENATEL una institución pública que brinda servicios a nivel nacional, maneja mucha información es de carácter confidencial, la cual solo debe ser divulgada con previa autorización.
- Se sugiere el uso del presente Proyecto como base para el desarrollo de nuevos sistemas que usan la herramienta SPECTRAplus, ya que solo en la DGGER se manejan alrededor de 18 servicios, como por ejemplo:
 - I. Servicio Móvil Avanzado (SMA) cuya particularidad radica en el uso del formulario RC-18A.
 - II. Servicio Fijo Terrestre bajo 1 GHz, que usa el formulario RC-6A en lo referente a la operación de enlaces punto-punto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] CONATEL, «Resolución de Telecomunicaciones,» [En línea]. Available: http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC4QFjAA&url=http%3A%2F%2Fwww.supertel.gob.ec%2Fpdf%2Fleyes_reglamentos%2Freglamento_radio_comunicaciones.doc&ei=HDYBU6zaFsfmkAejyIHACQ&usg=AFQjCNGYpn2Q0eV8szFOg8eQ_ghf9rTsOw&sig2=ExJEK. [Último acceso: 02 01 2014].
- [2] «Ley especial de Telñecomunicaciones Reformada,» [En línea]. Available: http://www.cnt.gob.ec/images/Pdfs/lotaip/ley_especial_telecomunicaciones.pdf. [Último acceso: 2013 01 01].
- [3] SUPERTEL, «SUPERTEL,» [En línea]. Available: <http://www.supertel.gob.ec/index.php/Estadisticas/Servicios-de-Telecomunicaciones.html>. [Último acceso: 10 08 2010].
- [4] «TIC´s 2013,» INEC, [En línea]. Available: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf. [Último acceso: 03 02 2014].
- [5] SENATEL, «Plan Nacional de Frecuencias,» 2012. [En línea]. Available: http://www.regulaciontelecomunicaciones.gob.ec/wp-content/uploads/downloads/2013/07/plan_nacional_frecuencias_2012.pdf. [Último acceso: 02 01 2014].
- [6] «Resolución N. 417-15 CONATEL-2005,» CONATEL, [En línea]. Available: http://www.regulaciontelecomunicaciones.gob.ec/wp-content/uploads/downloads/2013/07/2005_a417_15.pdf. [Último acceso: 06 01 2014].
- [7] S. Web, «W3C,» [En línea]. Available: <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>. [Último acceso: 5 11 2013].
- [8] «Hipertexto,» [En línea]. Available: <http://es.wikipedia.org/wiki/Hipertexto>. [Último acceso: 10 10 2013].
- [9] «Navegadores Web,» [En línea]. Available: <http://blog.uptodown.com/las-nuevas-versiones-de-opera-se-basaran-en-el-codigo-de-chrome/>. [Último acceso: 10 10 2013].
- [10] «eXtensible Markup Language (XML),» María Jesús Lamarca Lapuente. Hipertexto: El nuevo concepto de documento en la cultura de la imagen., [En línea]. Available: <http://www.hipertexto.info/documentos/xml.htm>. [Último acceso: 26 05 2014].

- [11] R. T. Fielding, «REST,» 2000. [En línea]. Available: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. [Último acceso: 2014 02 02].
- [12] «El Cifrado Web (SSL/TLS),» Universidad Nacional Autónoma de México, [En línea]. Available: <http://revista.seguridad.unam.mx/numero-10/el-cifrado-web-sslts>. [Último acceso: 01 02 2014].
- [13] «Informática Hoy,» RAID, [En línea]. Available: <http://www.informatica-hoy.com.ar/hardware-pc-desktop/Que-es-RAID.php>. [Último acceso: 03 01 2014].
- [14] «Los niveles y tipos RAID,» Informática Hoy, [En línea]. Available: <http://www.informatica-hoy.com.ar/hardware-pc-desktop/Los-niveles-de-RAID.php>. [Último acceso: 05 01 2014].
- [15] «SOA,» [En línea]. Available: <http://tecnologiasemergentesuva.wordpress.com/2013/11/27/soa-arquitectura-orientada-a-servicios/>. [Último acceso: 10 06 2013].
- [16] S. O. Isaac Gutierrez, «SOA,» [En línea]. Available: <http://ceur-ws.org/Vol-132/paper09.pdf>. [Último acceso: 15 08 2013].
- [17] «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» [En línea]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Último acceso: 10 01 2014].
- [18] SCRUM, «Manifiesto Agil,» [En línea]. Available: <http://agilemanifesto.org/iso/es/>.
- [19] J. P. y. C. Ruata, Libro Scrum Manager Gestión de Proyectos, 2010.
- [20] SENATEL, «SENATEL,» [En línea]. Available: <http://www.regulaciontelecomunicaciones.gob.ec/>. [Último acceso: 2014 02 02].
- [21] «Instructivo_formularios_concesion_frecuencias,» SENATEL, [En línea]. Available: http://www.regulaciontelecomunicaciones.gob.ec/wp-content/uploads/downloads/2013/07/instructivo_formularios_concesion_frecuencias.pdf. [Último acceso: 05 05 2014].
- [22] «Overview of MWA and MWM for IIS 7.0,» Microsoft, [En línea]. Available: <http://www.iis.net/learn/develop/extending-the-management-ui/overview-of-mwa-and-mwm-for-iis>. [Último acceso: 10 05 2013].
- [23] «TFS,» Microsoft, [En línea]. Available: <http://tfs.visualstudio.com/>. [Último acceso: 15 10 2013].
- [24] «How do I get a TextBox to only accept numeric input in WPF?,» StackOverflow, [En línea]. Available: <http://stackoverflow.com/questions/1268552/how-do-i-get-a-textbox-to-only-accept-numeric-input-in-wpf>. [Último acceso: 10 01 2014].

- [25] «WPF Textbox Preview events related,» StackOverflow, [En línea]. Available: <http://stackoverflow.com/questions/3003711/wpf-textbox-preview-events-related>.
- [26] «Convertir coordenadas geográficas en UTM y UTM en geográficas,» Gabriel Ortiz, [En línea]. Available: <http://www.gabrielortiz.com/index.asp?Info=058a>. [Último acceso: 01 03 2014].
- [27] «Blog José Guerrero,» [En línea]. Available: <http://joseguerreroa.wordpress.com/2011/10/01/metodo-para-convertir-coordenadas-geograficas-en-utm-clase-coordenadas-cc/>. [Último acceso: 20 03 2014].
- [28] «User controls,» CodeProject, [En línea]. Available: <http://www.codeproject.com/Articles/32825/How-to-Creating-a-WPF-User-Control-using-it-in-a-W>. [Último acceso: 01 11 2013].
- [29] «User control,» Microsoft, [En línea]. Available: <http://msdn.microsoft.com/en-us/library/system.windows.controls.usercontrol%28v=vs.110%29.aspx>. [Último acceso: 15 10 2013].
- [30] «ITEXT,» Source Force, [En línea]. Available: <http://sourceforge.net/projects/itextsharp/>. [Último acceso: 5 10 2013].
- [31] «iText,» SourceForce, [En línea]. Available: <http://sourceforge.net/projects/itextsharp/>. [Último acceso: 10 08 2014].
- [32] «iTextSharp,» [En línea]. Available: http://www.lawebdelprogramador.com/foros/ASP.NET/892872-Crear_pdf_con_iTextSharp.html. [Último acceso: 10 10 2013].
- [33] «XML,» Microsoft, [En línea]. Available: <http://msdn.microsoft.com/es-es/library/x6c1kb0s%28v=vs.80%29.aspx>. [Último acceso: 02 01 2014].
- [34] «Creating an XML file based on XSD,» Codeproject, [En línea]. Available: <http://www.codeproject.com/Articles/15989/Creating-an-XML-file-based-on-XSD>.
- [35] «An Introduction To RESTful Services With WCF,» Microsoft, [En línea]. Available: <http://msdn.microsoft.com/es-es/magazine/dd315413.aspx>. [Último acceso: 03 03 2014].
- [36] «Async WCF REST services using WCF WebApi,» MSDN, [En línea]. Available: <http://social.msdn.microsoft.com/Forums/vstudio/en-US/901ff37c-9843-47e8-b804-a4ae8841b30b/async-wcf-rest-services-using-wcf-webapi?forum=wcf>.
- [37] «C# 5: Async / Await,» [En línea]. Available: <http://geeks.ms/blogs/etomas/archive/2011/09/17/c-5-async-await.aspx>.
- [38] «Streaming Files (for Upload/Download) in WCF (Message Contracts),» [En línea]. Available: <http://www.seesharpdot.net/?p=214>. [Último acceso: 04 04 2014].

- [39] «WCF Streaming: Upload files over HTTP,» InfoWorker Solutions, [En línea]. Available: <http://kjellsj.blogspot.com/2007/02/wcf-streaming-upload-files-over-http.html>. [Último acceso: 04 04 2014].
- [40] «RESTful WCF Part 2 of n,» CodeProject, [En línea]. Available: <http://www.codeproject.com/Articles/35278/RESTful-WCF-Part-2-of-n>.
- [41] «Google Maps,» [En línea]. Available: <http://www.google.com/intl/es/enterprise/mapsearch/products/mapsapi.html>. [Último acceso: 02 01 2014].
- [42] «GoogleMaps,» [En línea]. Available: <http://maximn.github.io/google-maps/>. [Último acceso: 10 12 2013].
- [43] «Working with the Google Location API,» StackOverflow, [En línea]. Available: <http://stackoverflow.com/questions/4824348/working-with-the-google-location-api>. [Último acceso: 20 12 2013].
- [44] «Guía para desarrolladores de la API de Google Earth,» Google, [En línea]. Available: <https://developers.google.com/earth/documentation/?hl=es>. [Último acceso: 01 01 2014].
- [45] «Geocode Dataflow Sample Input and Output Data Version 2.0,» Microsoft, [En línea]. Available: <http://msdn.microsoft.com/en-us/library/jj735475.aspx>.
- [46] «Microsoft Word Documents from ASP.NET,» CodeProject, [En línea]. Available: <http://www.codeproject.com/Articles/3959/Microsoft-Word-Documents-from-ASP-NET>. [Último acceso: 15 01 2014].
- [47] «Como generar Documentos de Word desde C#,» WordPress, [En línea]. Available: <http://miguelalvareza.wordpress.com/2013/01/12/como-generar-documentos-de-word-desde-c/>. [Último acceso: 02 02 2014].
- [48] «Enviar correo electrónico por código,» [En línea]. Available: <http://geeks.ms/blogs/jalarcon/archive/2007/06/23/c-243-mo-enviar-correo-electr-243-nico-por-c-243-digo-usando-cuentas-de-gmail.aspx>. [Último acceso: 01 03 2014].
- [49] «Web Services Description Language (WSDL),» [En línea]. Available: <http://www.w3.org/TR/wsdl>. [Último acceso: 03 03 2014].
- [50] E. H. E. G., «SOA (Arquitectura Orientada a Servicios),» [En línea]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/587/1/CAPITULOS.pdf>. [Último acceso: 02 10 2014].
- [51] «Proceso automatico.. cada cierto tiempo,» Microsoft, [En línea]. Available: <http://social.msdn.microsoft.com/forums/es-ES/d95beb32-a9a2-4f84-bbcf-bf75f51d0e36/proceso-automatico-cada-cierto-tiempo>. [Último acceso: 02 05 2014].

- [52] «How do I create a timer in WPF?,» StackOverflow, [En línea]. Available: <http://stackoverflow.com/questions/11559999/how-do-i-create-a-timer-in-wpf>. [Último acceso: 03 05 2014].
- [53] «Seven simple steps to enable HTTPS on WCF WsHttp bindings,» CodeProject, [En línea]. Available: <http://www.codeproject.com/Articles/36705/7-simple-steps-to-enable-HTTPS-on-WCF-WsHttp-bindings>. [Último acceso: 01 04 2014].
- [54] « Creating a WCF RESTful Service And Secure It Using HTTPS Over SSL,» [En línea]. Available: <http://allen-conway-dotnet.blogspot.com/2012/05/creating-wcf-restful-service-and-secure.html>. [Último acceso: 06 04 2014].
- [55] «SSL binding for WCF REST service,» Microsoft, [En línea]. Available: <http://social.msdn.microsoft.com/Forums/vstudio/en-US/9bf26984-f6e9-4ff8-ad89-38903fa55c7c/ssl-binding-for-wcf-rest-service?forum=wcf>. [Último acceso: 01 05 2014].
- [56] «RAID,» DLINK, [En línea]. Available: <http://www.dlink.com/-/media/Files/B2B%20Briefs/ES/dlinkraid.pdf>.
- [57] «Implementar un RAID 5 con Virtualbox y Windows 2008 (cliente),» [En línea]. Available: <http://fundamentoshardwareantonio.blogspot.com/2013/02/implementar-un-raid-5-con-virtualbox-y.html>. [Último acceso: 01 06 2014].

ANEXO A

ANÁLISIS DEL FORMATO XML

En el presente documento se presenta el análisis del formato XML importable al SPECTRAplus correspondiente a cada campo de los trámites tanto para MDBA como para Servicio Móvil Terrestre UHF/VHF Radio de Dos Vías.

MDBA

Formulario RC-1B

Objeto de la Solicitud	<AP_CATEGORY>4</AP_CATEGORY >
Tipo de Uso	<LI_CAT>2</LI_CAT>
Persona Natural o Representante Legal	
Apellido Paterno	*
Apellido Materno	*
Nombres	*
CI	*
Cargo	*
Persona Jurídica	
Nombre de la Empresa	<AD_COMPANY></AD_COMPANY>
Actividad de la Empresa	<AD_ACTIVITY></AD_ACTIVITY>
RUC	<AD_TAX_ADMIN_NUM></AD_TAX_ADMIN_NUM>
Dirección	
Provincia	<AD_COUNTY></AD_COUNTY>
Ciudad	<AD_CITY></AD_CITY>
Dirección	<AD_STREET></AD_STREET>
Email	<AD_E_MAIL></AD_E_MAIL>
Teléfono	<AD_PHONE> </AD_PHONE>
Certificado del Profesional Técnico	
Apellido Paterno	*
Apellido Materno	*
Nombres	*
LicProf	*
Email	*

Formulario RC-2A

Tipo de Estructura de Soporte	<SID_LOC>N</SID_LOC>
Altura de la Estructura S.N.M	<SID_H_NN>2955</SID_H_NN>
Código de Registro de la Estructura	<SID_SITE_IDENT>SNF0036</SID_SITE_IDENT>
Altura de la Estructura	<SID_STRUCT_HEIGHT>12</SID_STRUCT_HEIGHT>
UBICACIÓN DE LA ESTRUCTURA	
Provincia	<AD_COUNTY>COTOPAXI</AD_COUNTY>
Cantón	<AD_DISTRICT>LATACUNGA</AD_DISTRICT>
Localidad/Calle y N	<AD_STREET> SECTOR EL CALVARIO, CALLE ISLA ESPAÑOLA Y FERNANDINA ESQUINA </AD_STREET>
Latitud	<SID_LAT_DEG>0</SID_LAT_DEG> <SID_LAT_N_S>S</SID_LAT_N_S> <SID_LAT_MIN>7</SID_LAT_MIN> <SID_LAT_SEC>32.72</SID_LAT_SEC>
Longitud	<SID_LONG_DEG>78</SID_LONG_DEG> <SID_LONG_E_W>W</SID_LONG_E_W>

	<SID_LONG_MIN>28</SID_LONG_MIN> <SID_LONG_SEC>9.36</SID_LONG_SEC>
Tipo de Fuente de Energía	<SID_COST_CATEGORY>1</SID_COST_CATEGORY>
Tipo de Respaldo	<SID_NOISE_CODE>3</SID_NOISE_CODE>

Formulario RC-3A

Código de Antena	<EAN_ANT_IDENT>AHEF0003</EAN_ANT_IDENT>
Marca	<EAN_MANUFACTURE>MOTOROLA</EAN_MANUFACTURE>
Modelo	<EAN_NAME>CANOPY 5701 SM</EAN_NAME>
Rango de Frecuencias	*
Tipo	<EAN_TYPE>PANNEL</EAN_TYPE>
Impedancia	<EAN_VARIANT>50</EAN_VARIANT>
Polarización	<EAC_AN_POL>H</EAC_AN_POL>
Ganancia	<EAN_GAIN>4.85</EAN_GAIN>
Diámetro	*
Azimut de Radiación Máxima	<EAC_AN_AZI>78.703</EAC_AN_AZI>
Angulo de Elevación	<EAC_AN_ELEV>2.507</EAC_AN_ELEV>
Altura Base-Antena	<EAC_AN_H>11</EAC_AN_H>

Formulario RC-4A

Tipo de Estación	
Código de Equipo	<EQP_EQUIP_IDENT>EMO0151</EQP_EQUIP_IDENT>
Marca	<EQP_EQUIP_PROD>MOTOROLA</EQP_EQUIP_PROD>
Modelo	<EQP_EQUIP_MODEL>CANOPY 5701SM</EQP_EQUIP_MODEL>
Achura de Banda	<EQ_EQUIP_OTHER>5250 5350</EQ_EQUIP_OTHER >
Separación entre Tx y Rx	*
Tipo de Modulación	<EQ_MODULATION>OFDM</EQ_MODULATION>
Velocidad de Transmisión	<EQ_FS_TRANSFER_RATE>10000000000</EQ_FS_TRANSFER_RATE> <EQ_FS_TRANSFER_RATE_DUNIT>Mbit/s</EQ_FS_TRANSFER_RATE_DUNIT>
Potencia de Salida	<EQ_COMMENT>0.2</EQ_COMMENT>
Clase de Emisión	*
Rango de Operación	<EQ_FREQ_BAND>2</EQ_FREQ_BAND>
Sensibilidad	*
Máxima Desviación de Frecuencia	*

Formulario RC-9A (SISTEMAS MDBA PUNTOPUNTO)

Clase de Sistema	
Características Técnicas y de Operación del Sistema Fijo Punto-Punto	
Número de Enlace	*
Banda de Frecuencias	*
Tipo de Operación	<EQ_EQUIP_TYPE>T</EQ_EQUIP_TYPE >
Distancia del Enlace	*
Características de las Estaciones Fijas	
Indicativo	*

AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Estructura Asociada	<SID_SITE_IDENT>SNF0036</SID_SITE_IDENT>
Antenas Asociadas	<EAN_ANT_IDENT>AHEF0003</EAN_ANT_IDENT>
Potencia de Operación	<ETX_EQ_OUTPUT>0.0397</ ETX_EQ_OUTPUT >
Equipo Utilizado	<EQP_EQUIP_IDENT>EMO0151</EQP_EQUIP_IDENT>
Perfil Topográfico	
Distancia	*
Altura s.n.m	*

Formulario RC-9B (SISTEMAS MDBA PUNTOMULTIPUNTO)

Clase de Sistema	
Características Técnicas y de Operación del Sistema Fijo Punto-Multipunto	
Número de Sistema	*
Número de Estaciones por Sistema	*
Banda de Frecuencias	*
Tipo de Operación	<EQ_EQUIP_TYPE>T</ EQ_EQUIP_TYPE >
Características de las Estaciones Fijas	
Indicativo	
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Estructura Asociada	<SID_SITE_IDENT>SNF0036</SID_SITE_IDENT>
Antenas Asociadas	<EAN_ANT_IDENT>AHEF0003</EAN_ANT_IDENT>
Potencia de Operación	<ETX_EQ_OUTPUT>0.0397</ ETX_EQ_OUTPUT >
Equipo Utilizado	<EQP_EQUIP_IDENT>EMO0151</EQP_EQUIP_IDENT>
Características de las Estaciones Fijas	
Indicativo	*
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Estructura Asociada	<SID_SITE_IDENT>SNF0036</SID_SITE_IDENT>
Antena Asociada	*
Potencia de Operación	*
Equipo Utilizado	*

MT-RDV (SISTEMA MOVIL TERRESTRE RADIO DE DOS VÍAS)

Formulario RC-1A

Objeto de la Solicitud	<AP_CATEGORY>4</AP_CATEGORY >
Tipo de Uso de Frecuencias	
Tipo de Sistema	<LI_CAT>1</LI_CAT> <LI_PAY_CATEGORY>PRI</ LI_PAY_CATEGORY >
Servicio	*
Persona Natural o Representante Legal	
Apellido Paterno	*
Apellido Materno	*
Nombres	*
CI	*
Cargo	
Persona Jurídica	
Nombre de la Empresa	<AD_COMPANY>POLICIA METROPOLITANA DE QUITO</ AD_COMPANY >
Actividad de la Empresa	<AD_TAX_ADMIN_NUM>1768120360001</ AD_TAX_ADMIN_NUM >
RUC	<AD_TAX_ADMIN_NUM></AD_TAX_ADMIN_NUM>
Dirección	
Provincia	<AD_COUNTY></AD_COUNTY>
Ciudad	<AD_CITY></AD_CITY>
Dirección	<AD_STREET></AD_STREET>
Email	<AD_E_MAIL></AD_E_MAIL>
Teléfono	
Certificado del Profesional Técnico	
Apellido Paterno	*
Apellido Materno	*
Nombres	*
LicProf	*
Email	*

Formulario RC-2A

Tipo de Estructura de Soporte	<SID_LOC>M</SID_LOC>
Altura de la Estructura S.N.M	<SID_H_NN>2911</SID_H_NN>
Código de Registro de la Estructura	<SID_SITE_IDENT>SMA1511</ SID_SITE_IDENT >
Altura de la Estructura	<SID_STRUCT_HEIGHT>12</SID_STRUCT_HEIGHT>
UBICACIÓN DE LA ESTRUCTURA	
Provincia	<AD_COUNTY>PICHINCHA</AD_COUNTY>
Ciudad	<AD_CITY>QUITO</AD_CITY>
Cantón	<AD_DISTRICT>QUITO</AD_DISTRICT>
Localidad/Calle y N	<AD_STREET>FRANCISCO ENRIQUEZ N6329 Y CAMILO ECHANIQUE</AD_STREET>
Latitud	<SID_LAT_DEG>0</SID_LAT_DEG> <SID_LAT_N_S>S</SID_LAT_N_S> <SID_LAT_MIN>7</SID_LAT_MIN> <SID_LAT_SEC>32.72</SID_LAT_SEC>
Longitud	<SID_LONG_DEG>78</SID_LONG_DEG> <SID_LONG_E_W>W</SID_LONG_E_W> <SID_LONG_MIN>28</SID_LONG_MIN> <SID_LONG_SEC>9.36</SID_LONG_SEC>
Tipo de Fuente de Energía	<SID_COST_CATEGORY>1</SID_COST_CATEGORY>

Tipo de Respaldo	<SID_NOISE_CODE>3</SID_NOISE_CODE>
------------------	------------------------------------

Formulario RC-3A

Código de Antena	<EAN_ANT_IDENT>AUBL0003</EAN_ANT_IDENT>
Marca	<EAN_MANUFACTURE>UBIQUITI NETWORKS</EAN_MANUFACTURE>
Modelo	<EAN_NAME>NANO STATIONS 2</EAN_NAME>
Rango de Frecuencias	
Tipo	<EAN_TYPE>PANNEL</EAN_TYPE>
Impedancia	
Polarización	<EAC_AN_POL>V</EAC_AN_POL>
Ganancia	<EAN_GAIN>7.85</EAN_GAIN>
Diámetro	<EAN_DIAM>
Azimut de Radiación Máxima	<EAC_AN_AZI>54.821</EAC_AN_AZI>
Angulo de Elevación	<EAC_AN_ELEV>2.507</EAC_AN_ELEV>
Altura Base Antena	<EAC_AN_H>8</EAC_AN_H>

Formulario RC-4A

Tipo de Estación	<EQP_EQUIP_TYPE>RP</EQP_EQUIP_TYPE >
Código de Equipo	<EQP_EQUIP_IDENT>EKE0003</EQP_EQUIP_IDENT >
Marca	<EQP_EQUIP_PROD>MOTOROLA</EQP_EQUIP_PROD>
Modelo	<EQP_EQUIP_MODEL>TKR750</EQP_EQUIP_MODEL>
Achura de Banda	<EQP_RF_BWIDTH>12500</EQP_RF_BWIDTH > <EQP_RF_BWIDTH_DUNIT >kHz</EQP_RF_BWIDTH_DUNIT >
Separación entre Tx y Rx	*
Tipo de Modulación	<EQ_MODULATION>FM</EQ_MODULATION>
Velocidad de Transmisión	
Potencia de Salida	<ETX_EQ_OUTPUT>0.135831</ETX_EQ_OUTPUT>
Clase de Emisión	*
Rango de Operación	<EQ_FREQ_BAND>2</EQ_FREQ_BAND>
Sensibilidad	<ERX_MIN_SENSE>0.0000022</ERX_MIN_SENSE > <ERX_MIN_SENSE_DUNIT>uV</ERX_MIN_SENSE_DUNIT >
Máxima Desviación de Frecuencia	*

Formulario RC-5A

Características de Operación por Circuito	
No. Circuito	*
Banda de Frecuencias	*
Rango en la Banda Requerida	<EFL_FREQ>160075000</EFL_FREQ > <EFL_FREQ_DUNIT>MHz</EFL_FREQ_DUNIT > <EFL_FREQ >161075000</EFL_FREQ > <EFL_FREQ_DUNIT>MHz</EFL_FREQ_DUNIT >
No. de Frecuencias por Circuito	<TCC_NW_NAME>2</TCC_NW_NAME>
Región a Operar	
Provincia(s)	*
Ciudad(es)	*
Modo de Operación	<TCC_NW_SERVICE_TYPE>NA</TCC_NW_SERVICE_TYPE>
Horario de operación	*
Potencia de Salida	*
Anchura de Banda	*

Clase de Emisión	*
Número de Estaciones	
No. de Estaciones Repetidoras	*
No. de Estaciones Fijas	*
No. de Estaciones Móviles	*
No. de Estaciones Portátiles	*
No. Total de Estaciones	*
Características de las Estaciones Repetidoras	
Nombre	<TCS_NAME>REPETIDOR001</TCS_NAME> <TCS_NAME>FIJA001</TCS_NAME> <TCS_NAME>MOVIL001</TCS_NAME> <TCS_NAME>PORTATIL001</TCS_NAME>
Indicativo	<TCS_CALL>HCZ9128</TCS_CALL>
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Estructura Asociada	*
Antena Asociada	*
Altura Efectiva de Antena	*
Equipo utilizado	<EQP_EQUIP_IDENT>EKE0003</ EQP_EQUIP_IDENT >
Características de las Estaciones Fijas	
Indicativo	<TCS_CALL>HCZ9145</TCS_CALL>
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Estructura Asociada	*
Antena Asociada	*
Equipo Utilizado	<EQP_EQUIP_IDENT>EMO0007</ EQP_EQUIP_IDENT >
Características de las Estaciones Móviles	
Indicativo	<TCS_CALL>HCZ9146</TCS_CALL>
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Equipo Utilizado	<EQP_EQUIP_IDENT>EMO0007</ EQP_EQUIP_IDENT >
Características de las Estaciones Portátiles	
Indicativo	<TCS_CALL>HCZ9148</TCS_CALL>
AC	<AD_UPDATE_STATUS>A</ AD_UPDATE_STATUS >
Equipo Utilizado	<EQP_EQUIP_IDENT>EMO0011</ EQP_EQUIP_IDENT >

ANEXO B

HISTORIAS DE USUARIO

MDBA

Usuario 1: Concesionario

Usuario 2: Ingeniero analista de la SENATEL

Usuario 1	Historia 1
Nombre historia: Crear ventana Principal MDBA	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
Descripción: En la ventana principal se deberá presentar cada uno de los formularios a llenar en MDBA (1B, 2A, 3A, 4A, 9A, 9B y 15A)	
Observaciones: Se crearan los botones de opciones Guardar, Abrir, Nuevo y Salir.	
Usuario 1	Historia 2
Nombre historia: Crear Formulario RC-1B	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
Descripción: En el formulario RC-1B el usuario deberá especificar información de tipo legal.	
Observaciones: Formulario obligatorio.	
Usuario 1	Historia 3
Nombre historia: Formulario RC-1B, Solicitud	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 día	
Descripción: En el formulario de información legal RC-1B el usuario deberá especificar el objeto de la solicitud y el tipo de uso del sistema de una lista que se le presentará por cada uno.	
Observaciones: Los dos campos son obligatorios.	
Usuario 1	Historia 4
Nombre historia: Formulario RC-1B, Datos del Solicitante	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
Descripción: En el formulario de información legal RC-1B el usuario deberá indicar la siguiente información de la empresa: el RUC, de ser el caso: la actividad y dirección de la empresa, además del nombre, teléfono y correo del representante legal.	
Observaciones: Todos los campos son obligatorios. Si trabaja de manera online se validará que el ruc de la empresa conste en la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	
Usuario 1	Historia 5

Nombre historia: Formulario RC-1B, Datos del Profesional Técnico	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
Descripción: En el formulario de información legal RC-1B el usuario deberá ingresar la cédula del profesional técnico. Además se ingresará la dirección, teléfono, correo electrónico y licencia profesional del mismo.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se verificará que el título exista en la SENECYT y se autocompletarán el nombre del profesional técnico. La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 6
Nombre historia: Crear Formulario RC-2A	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
Descripción: Crear el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.	

Usuario 1	Historia 7
Nombre historia: Formulario RC-2A, Información de la Estructura	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 día	
Descripción: En el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A el usuario deberá especificar el tipo de estructura de soporte, las protecciones eléctricas y el tipo de respaldo de una lista que se le presentará por cada uno. Además ingresará los valores numéricos de la altura de la estructura y de la altura base cima.	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 8
Nombre historia: Formulario RC-2A, Coordenadas geográficas	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
Descripción: En el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A el usuario deberá ingresar las coordenadas geográficas, Además de ser el caso se ingresará la dirección exacta de la estructura.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una validación a nivel de Google Maps. La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos, estén dentro del rango asignado y que los campos no estén vacíos).	

Usuario 1	Historia 9
Nombre historia: Crear Formulario RC-3A	

Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: Crear el formulario de información de antenas RC-3A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.	

Usuario 1	Historia 10
Nombre historia: Formulario RC-3A, Antenas	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario de información de antenas RC-3A el usuario deberá seleccionar la marca y modelo de la antena. Se ingresará el rango de frecuencias, el tipo, la impedancia, la polarización y la altura. El azimut y el ángulo de elevación se calcularán automáticamente.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una consulta de las antenas a la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales.	

Usuario 1	Historia 11
Nombre historia: Crear Formulario RC-4A	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: Crear el formulario de información de equipos RC-4A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.	

Usuario 1	Historia 12
Nombre historia: Formulario RC-4A, Equipos	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario de información de equipos RC-4A el usuario deberá ingresar la marca y modelo del equipo, si se trabaja de manera online se realizará una consulta a la base de datos de la SENATEL. Además se ingresará la sensibilidad y tipo de modulación.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una consulta de los equipos a la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 13
Nombre historia: Crear Formulario RC-9A	
Prioridad en negocio: 1	Iteración asignada: 2

Tiempo estimado: 1 semana
Descripción: Crear el formulario para MDBA (Sistemas Punto-Punto) RC-9A.
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Eliminar y Validar.

Usuario 1	Historia 14
Nombre historia: Formulario RC-9A, Sistemas Punto-punto	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario para MDBA (Sistemas Punto-Punto) RC-9A el usuario deberá seleccionar la clase de sistema la banda de frecuencias y el tipo de operación.	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 15
Nombre historia: Formulario RC-9A, Estaciones Fijas	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 día	
Descripción: En el formulario MDBA RC-9A el usuario deberá ingresar la estructura asociada la antena asociada la potencia máxima de salida, el quipo utilizado y la distancia a la estación fija central.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 16
Nombre historia: Formulario RC-9A, Perfil topográfico	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 día	
Descripción: En el formulario MDBA RC-9A el usuario deberá ingresar las alturas concernientes a cada enlace.	
Observaciones: Todos los campos son obligatorios. No se validarán los esquemas del sistema.	

Usuario 1	Historia 17
Nombre historia: Crear Formulario RC-9B	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 2 semanas	

Descripción: Crear el formulario de información de antenas RC-9B.
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Eliminar y Validar.

Usuario 1	Historia 18
Nombre historia: Formulario RC-9B, Sistemas Punto-multipunto	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semanas	
Descripción: En el formulario de información de antenas RC-9B el usuario deberá seleccionar la clase de sistema la banda de frecuencias y el tipo de operación.	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 19
Nombre historia: Formulario RC-9B, Estación Fija Central	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	
Descripción: En el formulario para MDBA (Sistemas Punto-Punto) RC-9B, el usuario deberá ingresar la estructura asociada, la antena asociada, la potencia máxima de salida y el equipo utilizado.	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 20
Nombre historia: Formulario RC-9B, Estaciones Fijas	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	
Descripción: En el formulario MDBA RC-9B se deberá insertar el número de las estaciones fijas y para cada una de ellas ingresar: la estructura asociada, la antena asociada, la potencia máxima de salida y el equipo utilizado. La distancia a la estación fija central se calculará automáticamente	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 21
Nombre historia: Formulario RC-9B, Perfil topográfico	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	

Descripción: En el formulario MDBA RC-9B el usuario deberá ingresar las alturas concernientes a cada enlace.
Observaciones: Todos los campos son obligatorios. No se validarán los esquemas del sistema.

Usuario 1	Historia 22
Nombre historia: Formulario RC-15A, RNI	
Prioridad en negocio: 2	Iteración asignada: 3
Tiempo estimado: 2 semanas	
Descripción: En el formulario RNI RC-15A se deberá generar automáticamente con la información proporcionada por los otros formularios con el fin de que la aplicación realice los cálculos de R ² y del Slim Teórico	
Observaciones: No se validarán los esquemas del sistema.	

Usuario 1	Historia 23
Nombre historia: Imprimir Formularios	
Prioridad en negocio: 2	Iteración asignada: 4
Tiempo estimado: 1 semana	
Descripción: La aplicación permitirá generar los formularios de MDBA en el formato PDF establecido por la SENATEL	
Observaciones: No se dará tratamiento a las imágenes por lo cual el usuario deberá ingresarlas al PDF generado por la aplicación, después imprimirlas y firmar en los campos establecidos para la posterior entrega en la SENATEL.	

Usuario 1	Historia 24
Nombre historia: Guardar formularios	
Prioridad en negocio: 2	Iteración asignada: 4
Tiempo estimado: 2 semana	
Descripción: La aplicación permitirá guardar en una ubicación local los trámites generados.	
Observaciones: En caso de un cierre inesperado además se propone guardar la información de la aplicación en un archivo .bin	

Usuario 1	Historia 25
Nombre historia: Reconstruir formularios	
Prioridad en negocio: 2	Iteración asignada: 4
Tiempo estimado: 1 semana	
Descripción: La aplicación permitirá reconstruir los formularios para realizar modificaciones o renovaciones.	
Observaciones:	
Usuario 2	Historia 26
Nombre historia: Aplicación web, visualización.	
Prioridad en negocio: 3	Iteración asignada: 4
Tiempo estimado: 2 semanas	

Descripción: Crear una aplicación que permita visualizar el trámite de MDBA ingresado a través del Servicio Web.
Observaciones:

Usuario 2	Historia 27
Nombre historia: Aplicación Web, memo	
Prioridad en negocio: 3	Iteración asignada: 4
Tiempo estimado: 2 semanas	
Descripción: La aplicación permitirá generar un oficio para ser emitido al concesionario en caso de existir errores en la información visualizada. La aplicación web permitirá el envío de un correo electrónico al concesionario.	
Observaciones:	

Usuario 1	Historia 28
Nombre historia: Ingresar formularios	
Prioridad en negocio: 3	Iteración asignada: 5
Tiempo estimado: 4 semanas	
Descripción: La aplicación permitirá ingresar el trámite de MDBA en formato XML a un repositorio en la SENATEL a través de una opción en el mismo aplicativo.	

Usuario 1	Historia 29
Nombre historia: Servicio Web, RUC	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
Descripción: El Servicio web REST deberá permitir la validación del RUC del concesionario ingresada en el formulario 1A online ingresada a través de la aplicación web.	
Observaciones: Conexión a la BDD	

Usuario 1	Historia 30
Nombre historia: Servicio Web, Profesional Técnico	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 2 semanas	
Descripción: El Servicio web REST deberá permitir la validación del título profesional del técnico a través de la cédula ingresada a través de la aplicación web.	
Observaciones: Conexión con la SENESCYT	

Usuario 1	Historia 31
Nombre historia: Servicio Web, Coordenadas geográficas	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 2 semanas	

<p>Descripción: El Servicio REST deberá permitir la validación de las coordenadas geográficas ingresada a través de la aplicación web.</p>
<p>Observaciones: Conexión con el Servicio de Google Maps</p>

Usuario 1	Historia 32
Nombre historia: Servicio Web, Antenas	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
<p>Descripción: El Servicio Web REST deberá permitir la validación online de las marcas y modelos de antenas ingresados por parte del concesionario en los formularios 3A a través de la aplicación de escritorio.</p>	
<p>Observaciones: Conexión a la BDD</p>	

Usuario 1	Historia 33
Nombre historia: Servicio Web, Equipos	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
<p>Descripción: El Servicio Web REST deberá permitir la validación online de las marcas y modelos de los equipos ingresados por parte del concesionario en los formularios 4A a través de la aplicación de escritorio.</p>	
<p>Observaciones: Conexión a la BDD</p>	

MT-RDV

Usuario 1: Concesionario

Usuario 2: Ingeniero analista de la SENATEL

Usuario 1	Historia 1
Nombre historia: Crear ventana Principal MT-RDV	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
Descripción: En la ventana principal se deberá presentar cada uno de los formularios a llenar en MDBA (1A, 2A, 3A, 4A, 5A y 15A)	
Observaciones: Se crearan los botones de opciones Guardar, Abrir, Nuevo y Salir.	

Usuario 1	Historia 2
Nombre historia: Crear Formulario RC-1A	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
Descripción: En el formulario RC-1A el usuario deberá especificar información de tipo legal.	
Observaciones: Formulario obligatorio.	

Usuario 1	Historia 3
Nombre historia: Formulario RC-1A, Solicitud	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 día	
Descripción: En el formulario de información legal RC-1A el usuario deberá especificar el objeto de la solicitud, tipo de uso de frecuencias, el tipo de uso del sistema y el servicio de una lista que se le presentará por cada uno.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 4
Nombre historia: Formulario RC-1A, Datos del Solicitante	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
Descripción: En el formulario de información legal RC-1A el usuario deberá indicar la siguiente información de la empresa: el RUC, de ser el caso: la actividad y dirección de la empresa, además del nombre, teléfono y correo del representante legal.	
Observaciones: Todos los campos son obligatorios. Si trabaja de manera online se validará que el RUC de la empresa conste en la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 5
Nombre historia: Formulario RC-1A, Datos del Profesional Técnico.	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
<p>Descripción: En el formulario de información legal RC-1A el usuario deberá ingresar la cédula del profesional técnico. Además se ingresará la dirección, teléfono, correo electrónico y licencia profesional del mismo.</p>	
<p>Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se verificará que el título exista en la SENECYT y se autocompletarán el nombre del profesional técnico. La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).</p>	

Usuario 1	Historia 6
Nombre historia: Crear Formulario RC-2A	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 2 días	
<p>Descripción: Crear el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A.</p>	
<p>Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.</p>	

Usuario 1	Historia 7
Nombre historia: Formulario RC-2A, Información de la Estructura	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 día	
<p>Descripción: En el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A el usuario deberá especificar el tipo de estructura de soporte, las protecciones eléctricas y el tipo de respaldo de una lista que se le presentará por cada uno. Además ingresará los valores numéricos de la altura de la estructura y de la altura base cima.</p>	
<p>Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).</p>	

Usuario 1	Historia 8
Nombre historia: Formulario RC-2A, Coordenadas geográficas.	
Prioridad en negocio: 1	Iteración asignada: 1
Tiempo estimado: 1 semana	
<p>Descripción: En el formulario de información de la Infraestructura del sistema de radiocomunicación RC-2A el usuario deberá ingresar las coordenadas geográficas, Además de ser el caso se ingresará la dirección exacta de la estructura.</p>	
<p>Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una validación a nivel de Google Maps. La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos, estén dentro del rango asignado y que los campos no estén vacíos).</p>	

Usuario 1	Historia 9
-----------	------------

Nombre historia: Crear Formulario RC-3A	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: Crear el formulario de información de antenas RC-3A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.	

Usuario 1	Historia 10
Nombre historia: Formulario RC-3A, Antenas.	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario de información de antenas RC-3A el usuario deberá seleccionar la marca y modelo de la antena. Se ingresará el rango de frecuencias, el tipo, la impedancia, la polarización y la altura.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una consulta de las antenas a la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales.	

Usuario 1	Historia 11
Nombre historia: Crear Formulario RC-4A	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: Crear el formulario de información de equipos RC-4A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Duplicar, Eliminar y Validar.	

Usuario 1	Historia 12
Nombre historia: Formulario RC-4A, Equipos.	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario de información de equipos RC-4A el usuario deberá ingresar la marca y modelo del equipo, si se trabaja de manera online se realizará una consulta a la base de datos de la SENATEL. Además se ingresará la sensibilidad y tipo de modulación.	
Observaciones: Todos los campos son obligatorios. Si se trabaja de manera online se realizará una consulta de los equipos a la base de datos (en caso de tener acceso a la base de datos de la SENATEL). La validación online se la realizará al momento de implementar el servicio web. Si se trabaja de manera offline solo se realizará validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 13
------------------	--------------------

Nombre historia: Formulario RC-5A	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: Crear el formulario para Servicio Fijo y Móvil Terrestre (Radio de dos vías) RC-5A.	
Observaciones: Formulario obligatorio. Este formulario debe permitir las opciones, Nuevo, Eliminar y Validar.	

Usuario 1	Historia 14
Nombre historia: Formulario RC-5A	
Prioridad en negocio: 1	Iteración asignada: 2
Tiempo estimado: 1 semana	
Descripción: En el formulario para Servicio Fijo y Móvil Terrestre (Radio de dos vías) RC-5A el usuario deberá seleccionar la banda de frecuencias, el rango en la banda requerida, el número de frecuencias por circuito, la región a operar, es decir provincia(s) y ciudad (es), el modo de operación, el horario de operación, la anchura de banda, la clase de emisión y el número de estaciones.	
Observaciones: Todos los campos son obligatorios. Se realizarán validaciones locales (verificar que los valores ingresados sean numéricos y que los campos no estén vacíos).	

Usuario 1	Historia 15
Nombre historia: Formulario RC-5A, estaciones repetidoras	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 2 semanas	
Descripción: En el formulario RC-5A el usuario deberá ingresarlas características de las estaciones repetidoras como: el indicativo, la actividad a realizar con la estación indicada, la estructura asociada, la antena asociada, la altura efectiva de antena y el equipo utilizado.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 16
Nombre historia: Formulario RC-5A, estaciones fijas	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	
Descripción: En el formulario RC-5A el usuario deberá ingresarlas características de las estaciones fijas como: el indicativo, la actividad a realizar con la estación indicada, la estructura asociada, la antena asociada y el equipo utilizado.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 17
Nombre historia: Formulario RC-5A, estaciones móviles	

Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	
Descripción: En el formulario RC-5A el usuario deberá ingresarlas características de las estaciones móviles como: el indicativo, la actividad a realizar con la estación indicada y el equipo utilizado.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 18
Nombre historia: Formulario RC-5A, estaciones portátiles	
Prioridad en negocio: 1	Iteración asignada: 3
Tiempo estimado: 1 semana	
Descripción: En el formulario RC-5A el usuario deberá ingresarlas características de las estaciones portátiles como: el indicativo, la actividad a realizar con la estación indicada y el equipo utilizado.	
Observaciones: Todos los campos son obligatorios.	

Usuario 1	Historia 19
Nombre historia: Formulario RC-15A, RNI	
Prioridad en negocio: 2	Iteración asignada: 3
Tiempo estimado: 2 semanas	
Descripción: En el formulario RNI RC-15A se deberá generar automáticamente con la información proporcionada por los otros formularios con el fin de que la aplicación realice los cálculos de R ² y del Slim Teórico	
Observaciones: No se validarán los esquemas del sistema.	

Usuario 1	Historia 20
Nombre historia: Imprimir Formularios	
Prioridad en negocio: 2	Iteración asignada: 4
Tiempo estimado: 2 semanas	
Descripción: La aplicación permitirá generar los formularios de MDBA en el formato PDF establecido por la SENATEL	
Observaciones: No se dará tratamiento a las imágenes por lo cual el usuario deberá ingresarlas al PDF generado por la aplicación, después imprimirlas y firmar en los campos establecidos para la posterior entrega en la SENATEL.	

Usuario 1	Historia 21
Nombre historia: Guardar formularios	
Prioridad en negocio: 2	Iteración asignada: 4
Tiempo estimado: 2 semanas	
Descripción: La aplicación permitirá guardar en una ubicación local los trámites generados.	
Observaciones: En caso de un cierre inesperado además se propone guardar la información de la aplicación en un archivo .bin	

Usuario 1	Historia 22
Nombre historia: Reconstruir formularios	
Prioridad en negocio: 2	Iteración asignada: 4

Tiempo estimado: 1 semana
Descripción: La aplicación permitirá reconstruir los formularios para realizar modificaciones o renovaciones.
Observaciones:

Usuario 2	Historia 23
Nombre historia: Aplicación web, visualización	
Prioridad en negocio: 3	Iteración asignada: 4
Tiempo estimado: 2 semanas	
Descripción: Crear una aplicación que permita visualizar el trámite de MT-RDV ingresado a través del Servicio Web.	
Observaciones:	

Usuario 2	Historia 24
Nombre historia: Aplicación Web, memo	
Prioridad en negocio: 3	Iteración asignada: 4
Tiempo estimado: 1 semana	
Descripción: La aplicación permitirá generar un oficio para ser emitido al concesionario en caso de existir errores en la información visualizada. La aplicación web permitirá el envío de un correo electrónico al concesionario.	
Observaciones:	

Usuario 1	Historia 25
Nombre historia: Ingresar formularios	
Prioridad en negocio: 3	Iteración asignada: 5
Tiempo estimado: 4 semanas	
Descripción: La aplicación permitirá ingresar el trámite de MDBA en formato XML a un repositorio en la SENATEL a través de una opción en el mismo aplicativo.	
Observaciones:	

Usuario 1	Historia 26
Nombre historia: Servicio Web, RUC	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
Descripción: El Servicio web REST deberá permitir la validación el RUC del concesionario ingresada en el formulario 1A online ingresada a través de la aplicación web.	
Observaciones: Conexión a la BDD	

Usuario 1	Historia 27
Nombre historia: Servicio Web, Profesional Técnico	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 2 semanas	

<p>Descripción: El Servicio web REST deberá permitir la validación del título profesional del técnico a través de la cédula ingresada a través de la aplicación web.</p>
<p>Observaciones: Conexión con la SENESCYT</p>

Usuario 1	Historia 28
Nombre historia: Servicio Web, Coordenadas Geográficas	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 2 semanas	
<p>Descripción: El Servicio REST deberá permitir la validación de las coordenadas geográficas ingresada a través de la aplicación web.</p>	
<p>Observaciones: Conexión con el Servicio de Google Maps</p>	

Usuario 1	Historia 29
Nombre historia: Servicio Web, Antenas	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
<p>Descripción: El Servicio web REST deberá permitir la validación online de las marcas y modelos de antenas ingresados por parte del concesionario en los formularios 3A a través de la aplicación de escritorio.</p>	
<p>Observaciones: Conexión a la BDD</p>	

Usuario 1	Historia 30
Nombre historia: Servicio Web, Equipos	
Prioridad en negocio: 4	Iteración asignada: 5
Tiempo estimado: 1 semana	
<p>Descripción: El Servicio web REST deberá permitir la validación online de las marcas y modelos de los equipos ingresados por parte del concesionario en los formularios 4A a través de la aplicación de escritorio.</p>	
<p>Observaciones: Conexión a la BDD</p>	

ANEXO C

**FUNCIONES PARA EL CÁLCULO DE LA DISTANCIA AZIMUT Y
ÁNGULO DE ELEVACIÓN**

(DVD Adjunto)

ANEXO D
MANUALES DE USUARIO DE MDBA
(DVD Adjunto)

ANEXO E
MANUALES DE USUARIO DE MT-RDV
(DVD Adjunto)

ANEXO F
CONFIGURACIÓN DE HTTPS (Certificado Autofirmado)
(DVD Adjunto)

ANEXO G
PRUEBAS DE CONFIGURACIÓN DE HTTPS
(DVD Adjunto)

ANEXO H
IMPLEMENTACION DE RAID 5
(DVD Adjunto)

ANEXO I
MANUAL DE PRUEBAS
(DVD Adjunto)

ANEXO J
COTIZACIONES
(DVD Adjunto)

ANEXO K
INSTALADORES
(DVD Adjunto)

ANEXO L
CÓDIGO
(DVD Adjunto)