

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)

Pinto R. Andrea, Gaibor C. José, Echeverría Henry, Mejía N. David

Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional

andrepin29@gmail.com

albertogaibor89@gmail.com

henry.echeverria@epn.edu.ec

david.mejia@epn.edu.ec

Resumen- En este documento se presenta un resumen del proceso seguido para el diseño e implementación de un prototipo de Sistema Distribuido para coadyuvar en la automatización de los procesos de Otorgamiento de Títulos Habilitantes del Servicio Móvil Terrestre Radio de Dos Vías (MT-RDV) y Registros de Enlaces de Sistemas de Modulación Digital de Banda Ancha (MDBA) de la SENATEL. El prototipo está conformado por tres compones: una aplicación de escritorio que permite al concesionario realizar trámites concernientes a los dos servicios mencionados, un servicio web que valida la información ingresada en la aplicación de escritorio y una aplicación web que permite la visualización de los trámites por parte de los analistas de la SENATEL.

Términos para indexación—REST, SOA, Sistemas Distribuidos, Seguridades.

I. INTRODUCCIÓN

El objetivo del prototipo de sistema distribuido es automatizar los procesos de concesión, renovación, modificación o eliminación de una determinada frecuencia para los servicios MDBA y MT-RDV que administra la Secretaría Nacional de Telecomunicaciones¹ (SENATEL).

Para esto el sistema distribuido se conformará mediante tres componentes, y en el cual el concesionario podrá realizar sus trámites a través de una aplicación de escritorio y enviarlos haciendo uso de un servicio web que valide la información ingresada, mientras que del lado de la SENATEL, un analista podrá visualizar dichos trámites en una aplicación web e importar finalmente la información al sistema denominado SPECTRAplus².

La aplicación de escritorio, permite el ingreso de datos offline por parte del concesionario. En esta aplicación se realiza una validación tanto local, como en línea. Una vez que la información ha sido validada con el servicio web (en línea), se puede almacenar el trámite como un archivo XML importable al SPECTRAplus o un archivo PDF el cual cumple con los formatos establecidos por la SENATEL. Las validaciones, que esta aplicación realiza, toman en consideración las normas técnicas establecidas por la SENATEL para la obtención de un título habilitante de acuerdo al servicio de telecomunicaciones (MT-RDV o MDBA).

¹ El organismo que asume las funciones de regulación y control del espectro radioeléctrico, que cumplan las suprimidas Secretaría Nacional de Telecomunicaciones (SENATEL) y Superintendencia de Telecomunicaciones (SUPATEL) respectivamente, es la Agencia de Regulación y Control de las Telecomunicaciones (ARCOTEL), según lo

estipula la Ley Orgánica de Telecomunicaciones publicada en el registro oficial el Miércoles 18 de Febrero de 2015 con N° 439.

² SPECTRA_plus es un software propietario que permite el ingreso de información concerniente a los servicios de MDBA, MT-RDV entre otros.

El servicio web permite validar la información ingresada en la aplicación de escritorio, en el caso de presentarse errores se generan mensajes de alerta en los campos incompletos o incorrectos, ante lo cual el concesionario se beneficiará de poder enmendar estas posibles equivocaciones antes de enviar el trámite para su revisión, reduciendo el tiempo en la obtención del título habilitante. Si no se presentan errores, permitirá almacenar el archivo con formato XML en un repositorio, al cual podrá acceder el analista. Cabe mencionar que la tecnología para implementar los servicios web empleada se denomina REST (*Representational State Transfer*). Es conveniente aclarar que cierta información no puede ser validada por el servicio web, en particular no se validan las imágenes (fotos y firmas).

La aplicación web permite que un analista de la SENATEL puede realizar una verificación final antes de ingresar el archivo XML al sistema SPECTRAplus. De existir algún error en el trámite la aplicación web le permite al analista generar un reporte para ser remitido al concesionario. La Fig. 1 presenta los elementos del prototipo de sistema distribuido desarrollado.



Fig 1. Elementos del prototipo de sistema distribuido

II. DISEÑO DEL SISTEMA DISTRIBUIDO

Para la realización del prototipo se empleó la metodología Scrum la cual ayuda a describir y entender mejor el proceso de trabajo, conocer los problemas que puedan surgir y tomar decisiones respecto a los posibles cambios que se puedan proponer. En la Fig. 2 se presentan los roles del proyecto de acuerdo con Scrum.

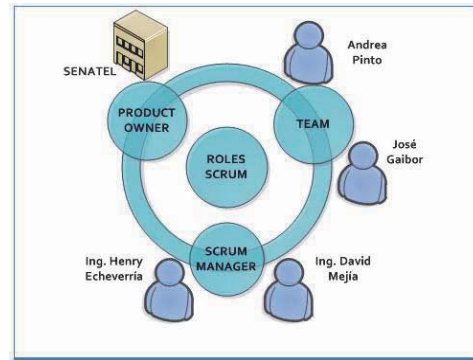


Fig 2. Roles de los participantes de acuerdo a la metodología Scrum

A continuación se describe brevemente el diseño y desarrollo de los tres componentes del Sistema Distribuido sobre Entorno de Desarrollo Integrado (IDE) Visual Studio 2012.

A. 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. Utilizar la tecnología WPF (*Windows Presentation Foundation*), proporciona un amplio abanico de funciones simplificando de forma significativa la creación de interfaces de usuario modernas, facilita la generación de interfaces que incorporan gráficos y vídeo, entre otros elementos, Además, puede mejorar la calidad de la aplicación y la experiencia del usuario optimizando el aprovechamiento de recursos.

Para validaciones locales, 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* mediante el cual se restringe al usuario el ingreso de números, letras o símbolos.

Mediante controles de usuario WPF, que permiten duplicar un conjunto de elementos (*textbox*, *combobox*, *label*, etc.) en tiempo de ejecución, se generaron controles personalizados. Además se empleó *TabControl*, control que permite simplificar la interfaz de usuario evitando la generación de ventanas adicionales, sino en su lugar se usan viñetas, como se muestra en la Fig. 3.

Para imprimir los formularios en formato PDF, se utilizó la librería *iText*, la cual permite crear, editar y generar archivos en formato PDF a partir de cualquier origen de datos, como por ejemplo objetos, los cuales se crearon a partir de los campos ingresados en los distintos formularios de la aplicación.

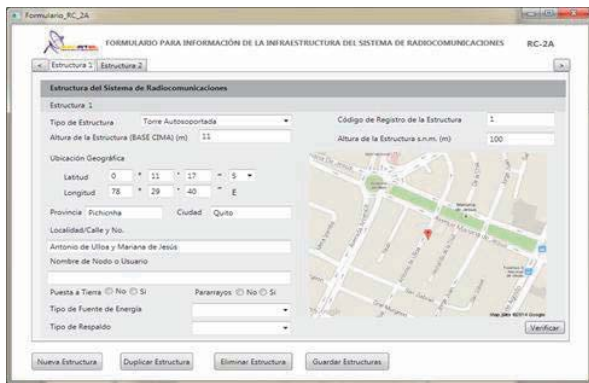


Fig 3. Control Personalizado

Para generar el archivo en formato XML, que una vez validado puede ser importado en el SPECTRAplus, se hizo uso de la librería *System.Xml*, la cual permite la creación del archivo XML. El archivo XML debe generarse con una estructura específica, considerando los campos necesarios y sus respectivos valores en posiciones específicas, de lo contrario el proceso de importación al SPECTRAplus fallará.

Las principales funcionalidades de la aplicación de escritorio son:

- 1) Permitir su descarga y posterior instalación en un equipo con sistema operativo Windows.
- 2) Permitir el ingreso de información offline.
- 3) Realizar validaciones locales offline.
- 4) Realizar cálculos matemáticos para determinar valores como el azimut, distancia entre estructuras, PIRE, etc.
- 5) Conectar con el servicio web para realizar validaciones online.
- 6) Generar automáticamente el Formulario RNI RC-15A el cálculo de la distancia de seguridad para la Radiaciones No Ionizantes.
- 7) Guardar la información ingresada en cada formulario en formato binario.
- 8) Generar los formularios en formato PDF.
- 9) Generar el archivo XML que posteriormente será importado al software SPECTRAplus.
- 10) Cargar nuevamente un trámite guardado en formato binario en caso que el concesionario requiera de alguna modificación o eliminación.

B. Servicio Web

El framework .NET, desde su versión 3.5, proporciona una opción para el desarrollo de servicios WCF, la cual se ha mejorado para la versión 4.0. Un servicio web WCF es un componente de software que no posee interfaz gráfica, pero permite la interoperabilidad o comunicación con otras

aplicaciones, se constituye básicamente de tres elementos principales:

- 1) **IService.cs**: permite establecer la Interfaz que define los métodos que el servicio hará públicos, es decir representa el contrato del servicio.
- 2) **Service.svc**: es la clase que implementará los métodos definidos en la interfaz.
- 3) **Web.config**: es el archivo de configuración XML de las características del servicio web, entre las que están: el tipo de servicio, el protocolo de comunicación que utilizará, entre otros.

Las principales funcionalidades del servicio web son:

- 1) Validar la información online, es decir permite al concesionario validar la información ingresada usando bases de datos confiables.
- 2) Validar los Títulos Profesionales, para lo cual el servicio realiza peticiones a la página de la SENESCYT, obteniendo el código del profesional.
- 3) Validar las Coordenadas Geográficas, para lo cual el servicio realiza peticiones al servicio de Google Maps.
- 4) Transferir archivos XML a un repositorio de pruebas.
- 5) Autenticar usuarios.

El servicio web se basa en la tecnología REST que dinamiza el acceso y el flujo de información, el cliente únicamente debe utilizar un URL para proporcionar los parámetros necesarios al servicio y que este le retorne información. Los métodos del cliente que consumen el servicio pueden emplear un objeto de tipo *webClient* el cual disponen de métodos que permiten, haciendo uso de parámetro como un URL, enviar la solicitud al servicio web; al recibir la solicitud, el servicio podrá obtener información de las distintas bases de datos y retornar la información al cliente. La Fig. 4 presenta el contrato de servicio, el uso del objeto *webClient* para realizar la solicitud y la respuesta obtenida.

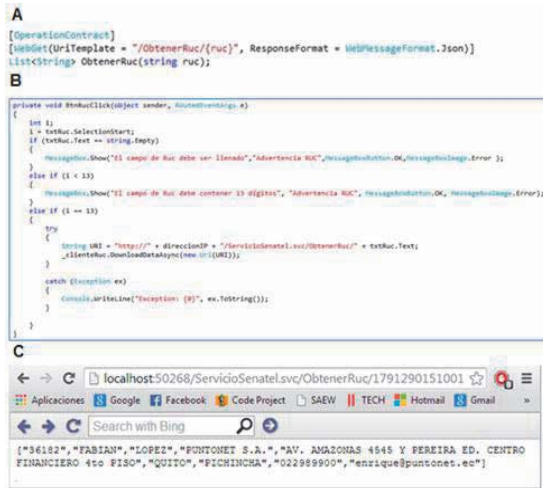


Fig 4. A Contrato de Servicio, B solicitud webClient, C Respuesta del Servicio Web

Para la validación de títulos profesionales, el servicio emplea el método POST de HTTP, mediante un objeto *WebRequest*, el cual permite descargar la información de la página web de la SENESCYT, luego procesa la respuesta haciendo uso de la librería *HtmlAgilityPack*, y obtener la información que es de interés.

Para la validación de coordenadas geográficas, el servicio pasa a ser un cliente de los servicios de geolocalización de Google, el cual se puede consumir mediante un URL, como la siguiente:

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

El servicio de Google espera como parámetros en la URL: latitud, longitud y sensor; las coordenadas previamente deben ser transformadas a formato decimal, el parámetro sensor indica si se dispone de un dispositivo sensor (por ejemplo, un localizador GPS) que no aplica en este caso (por ese motivo se especifica el valor *false* en la URL), y que generalmente se usa en aplicaciones para dispositivos móviles. El servicio de Google envía la respuesta en formato XML, por lo que se puede utilizar un objeto de tipo *XDocument* para almacenar la información y poder procesarla para obtener únicamente lo que es relevante para el servicio.

Para la transferencia de archivo es necesario manipular las propiedades del servicio web, mediante el archivo de configuración. En la Tabla 1 se presentan un resumen de las propiedades que deben modificarse, mientras que en la Fig.5 se presentan los cambios realizados en el archivo de configuración.

PROPIEDAD	DESCRIPCIÓN
MaxReceivedMessageSize	Obtiene o establece el tamaño máximo, en bytes, de un mensaje que puede ser procesado por el enlace.
MaxBufferSize	Obtiene o establece la cantidad máxima de memoria, en bytes, que se asigna para los mensajes que reciben del canal.
TransferMode	Obtiene o establece un valor que indica si el servicio utiliza streaming

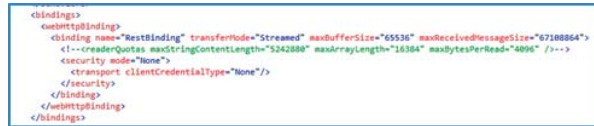


Fig. 5 Valores *TransferMode*, *MaxBufferSize* y *MaxReceivedMessageSize* modificados

C. Aplicación Web

Para el desarrollo de la aplicación web se usó ASP.NET. Para presentar la información de forma legible se usó un control *GridView*, 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 y solo se presenta la información requerida por el analista. Esta aplicación tiene las siguientes funciones:

- 1) Permitir la autenticación del analista y del concesionario.
- 2) Permitir la descarga de la aplicación de escritorio.
- 3) Convertir el trámite enviado por el concesionario en un archivo XML en información que pueda ser entendible por el analista.
- 4) Generar un memorando en caso de existir inconsistencias en la información remitida.
- 5) Permitir enviar notificaciones electrónicas al concesionario.

III. CAPAS ARQUITECTÓNICAS DEL SOFTWARE

La arquitectura a la que corresponde el sistema desarrollado es SOA, esta arquitectura 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, permite una mayor interoperabilidad y lo hace sencillo de consumir convirtiéndola en una arquitectura idónea para el desarrollo de sistemas distribuidos. La Fig. 6 muestra las capas de arquitectura SOA.



Fig. 6. Capas de la arquitectura SOA

A. Capa Aplicación

La Capa Aplicación corresponde a la interfaz de usuario que provee una solución a la necesidad de interacción. El prototipo cuenta con dos interfaces de usuario, una desarrollada mediante WPF que corresponde a la Aplicación de Escritorio y otra desarrollada en ASP.NET correspondiente a la Aplicación Web.

La aplicación WPF desarrollada dispone de interfaces (ventanas) para los trámites MDBA y MT-RDV, cada una de ellas con los controles que el concesionario requiere para el ingreso de la información legal y técnica.

B. Capa de Servicios [1]

La Capa de Servicios proporciona una correcta abstracción de las necesidades y la lógica de negocio que los componentes o funcionalidades deben proveer, se los desarrolló utilizando la plataforma de mensajería WCF incluida en el API de .NET desde su versión 3.0.

WCF es el primer modelo de programación construido desde cero para permitir explícitamente el desarrollo de aplicaciones orientadas a servicios.

D. Capa de 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 [2].

A más del uso de servicios que forman parte del proyecto (Validación RUC, Antenas, Equipos), se incluye servicios ya existentes; como el servicio de geolocalización de Google Maps.

E. Capa de Datos

Finalmente se encuentra la Capa de Datos o Persistencia la cual contiene los objetos que reciben la información de la base de datos.

Para el desarrollo del prototipo no se utilizó una base de datos sino un conjunto de tablas proporcionadas por la SENATEL, las cuales permitieron realizar las consultas necesarias para la validación del RUC, antenas y equipos.

IV. SEGURIDAD Y TOLERANCIA A FALLOS

Ninguna aplicación informática está exenta de sufrir algún tipo de amenaza en contra de su confidencialidad, integridad o disponibilidad, razón por la cual se debe priorizar esfuerzos para la protección de los componentes de la infraestructura física y lógica de la red.

A continuación se detallan las seguridades que se maneja en los componentes del prototipo.

A. Recuperación de la información

Se busca garantizar que las aplicaciones que conforman el sistema reaccionen de manera adecuada ante ciertos fallos, por ejemplo; 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.

- 1) Se marcó como *Serializable*, todas las clases que representan objetos, para de esta manera poder almacenarlos en un archivo binario.
- 2) El método que guarda la información es ejecutado en un hilo paralelo al hilo principal de la aplicación evitando interrupciones no deseadas. En la Fig. 7 se expone la inicialización del proceso para respaldar información, el cual es ejecutado en el evento *Load* del formulario MDBA y MT-RDV.

```
private void WindowLoaded1(object sender, RoutedEventArgs e)
{
    dispatcherTimer.Tick += new EventHandler(SalvarBinMovTerr);
    dispatcherTimer.Interval = new TimeSpan(0, 3, 0);
    dispatcherTimer.Start();
}
```

Fig. 7. Inicialización del mecanismo de respaldo

B. 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:

- 1) Concesionario: la cual permite la descarga de la aplicación de escritorio.
- 2) Analista: la cual permite la visualización de los trámites, el envío de notificaciones electrónicas y la generación del memorando.

C. MD5

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 que circulan por Internet. Debido a que el concesionario tiene la posibilidad de enviar un archivo XML, es imperativo el uso de algoritmos existentes para tratar de brindar seguridad al proceso de transferencia de archivos.

Por este motivo se emplea el algoritmo MD5 (*Message-Digest 5*), un algoritmo criptográfico de 128 bits muy popular, que se puede usar para comprobar la integridad de cualquier tipo archivo, de tal forma que se pueda proporcionar la seguridad de que el archivo transferido por el concesionario no ha sido alterado o modificado al navegar por Internet.

Tanto en la aplicación de escritorio como en el servicio web se definió una clase que permita calcular el código MD5 de un archivo, para luego enviarlo junto con el archivo para que sea recalculado en el servidor y compararlo con el enviado por el concesionario.

D. HTTPS

HTTPS (*Hypertext Transfer Protocol Secure*), es un protocolo de red basado en el protocolo HTTP, se basa en SSL/TLS para crear un canal cifrado más apropiado para el tráfico de información. De este modo se consigue que la información sensible no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de leer [3]. La implementación de HTTPS como mecanismo de seguridad se la realizó mediante la utilización de un certificado autofirmado instalado en el servidor que aloja el servicio web. Para que el servicio reconozca el certificado se debe modificar propiedades del archivo de configuración de tal forma que (ver Fig. 8):

- 1) Se utilice seguridad a nivel de transporte, modificando el atributo *mode* de la etiqueta `<security>`.
- 2) Especificar el protocolo HTTPS para el punto final del canal, modificando *mexHttpBinding* por *mexHttpsBinding*.
- 3) Modificar la configuración de los metadatos, en la etiqueta `<ServiceMetadata>` y habilitar la opción *httpsGetEnabled*.

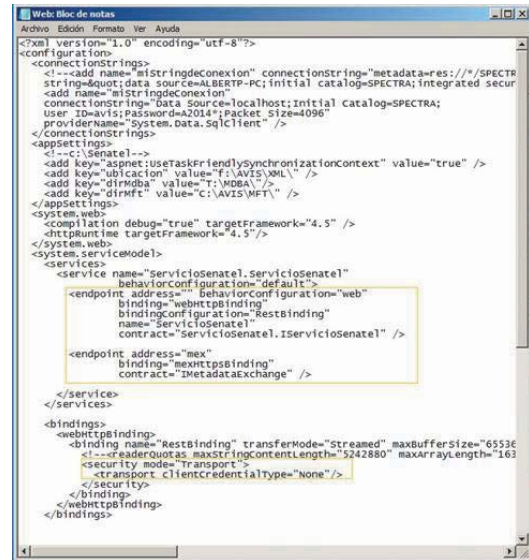


Fig. 8 Archivo Web.config alojado en el servidor web

D. RAID 5

En el servidor que alberga el servicio web se emplea RAID 5 a nivel de software. RAID 5 requiere de mínimo tres discos duros y máximo cinco discos duros. 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 [4], puesto que permite que una falla en el disco duro no afecte el funcionamiento del servicio web.

V. PRUEBAS

Al finalizar las pruebas con los analistas técnicos de la SENATEL se pudo constatar que se reduce el tiempo de elaboración, validación, ingreso de los trámites tanto de MDBA como de MT-RDV y la generación del Certificado, ya que la por una parte la aplicación de escritorio realiza validaciones y cálculos matemáticos que reducen el tiempo de elaboración del trámite y disminución de errores en el llenado de los formularios por parte del concesionario, y por otra la aplicación web permite al analista visualizar el trámite en una sola tabla e importar el archivo en formato XML al SPECTRAplus, en lugar que el analista digite

manualmente toda la información del trámite presentado por el concesionario .

En la TABLA II se pueden apreciar ejemplos de los tipos de trámites elaborados donde se aprecia la disminución de los tiempos de elaboración, validación e ingreso de los trámites.

TABLA II
COMPARACIÓN DE TIEMPO DE ELABORACIÓN E INGRESO DE LOS TRÁMITES

TIPO DE TRÁMITE	TIEMPO ANTERIOR	TIEMPO ACTUAL
Elaboración de un trámite de Sistemas MDBA con 12 enlaces punto-multipunto	El tiempo empleado sin una aplicación que valide es de: 120 minutos	El tiempo empleado en la aplicación de escritorio es de: 60 minutos
Validación e ingreso de un trámite de Sistemas MDBA con 12 enlaces punto-multipunto	El tiempo empleado en validar e ingresar manualmente el trámite es de: 120 minutos	El tiempo empleado en visualizar el trámite en la aplicación web y en importar el archivo al SPECTRAplus es de: 50 minutos
Elaboración de un trámite de Servicio MT-RDV con 12 estaciones	El tiempo empleado sin una aplicación que valide es de: 120 minutos	El tiempo empleado en la aplicación de escritorio es de: 60 minutos
Validación e ingreso de un trámite de Servicio MT-RDV con 12 estaciones	El tiempo empleado en validar e ingresar manualmente el trámite es de: 120 minutos	El tiempo empleado en visualizar el trámite en la aplicación web y en importar el archivo al SPECTRAplus es de: 40 minutos.

VI. COSTOS REFERENCIALES DEL SOFTWARE

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.).

Para comenzar con la estimación de costos se deberán definir las métricas 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 que se calcula fueron 2190 horas. Para realizar un costo aproximado se tomó como referencia un costo de 10 dólares la hora, obteniendo el resultado estimado de \$ 21.900,00.

VII. CONCLUSIONES

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.

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

Una de las principales funciones del prototipo 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.

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.

Al existir una cantidad razonable de solicitudes de registros de enlaces para sistemas MDBA y MT-RDV, se considera que el prototipo será de gran utilidad para el procesamiento de información técnica y legal solicitada, así como también las validaciones consideradas, evitando que el concesionario remita información con errores e impidiendo que el trámite deba ser devuelto por estos errores debido al incumplimiento de lo descrito en las Normas Técnicas.

El desarrollo del prototipo 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.

REFERENCIAS

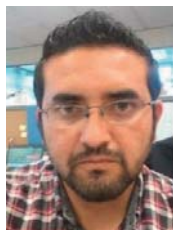
- [1] Nishith Pathak., Pro WCF 4 Practical Microsoft SOA Implementation, Apress, Second Edition.
- [2] E. H. E. G., «SOA (Arquitectura Orientada a Servicios),» [En línea]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/587/1/CAPITULOS.pdf>. [Último acceso: 02 10 2014].
- [3] «HTTPS,» [En línea]. Available: https://docs.google.com/document/d/1jfoOprvOscF4SYaS1QjZFzjjVm3vrV43mYDcjDrUBOU/edit?hl=en_US. [Último acceso: 10 10 2014].
- [4] «RAID,» DLINK, [En línea]. Available: <http://www.dlink.com/-/media/Files/B2B%20Briefs/ES/dlinkraid.pdf>. [Último acceso: 12 05 2014].

BIOGRAFÍAS



Andrea Patricia Pinto Ramírez

Nacida en Quito-Ecuador el 29 de diciembre de 1988. Ingeniera en Electrónica y Redes de Información, Escuela Politécnica Nacional, (Quito-Ecuador), 2014.



José Alberto Gaibor Coloma

Nacido en San Miguel-Ecuador el 07 de febrero de 1989. Sus estudios secundarios fueron realizados en el colegio San Felipe Neri (Riobamba), Bachiller en Ciencias Especialización Físico-Matemático obtenido en el año 2006, Ingeniero en Electrónica y Telecomunicaciones, Escuela Politécnica Nacional (Quito-Ecuador), 2014.



Henry Manolo Echeverría Culqui

Ingeniero de Sistemas de Computación e Informática, Escuela Politécnica Nacional (Quito-Ecuador), 1992; Magíster en Gestión de las Comunicaciones y Tecnologías de la Información, Escuela Politécnica Nacional (Quito-Ecuador), 2010. Profesor a tiempo parcial en la Facultad de Ingeniería de Sistemas en la Escuela Politécnica Nacional.



David Mejía Navarrete

Obtuvo el título de Ingeniero en Electrónica y Redes de Información, en la Escuela Politécnica Nacional en Quito – Ecuador, en 2005. Obtuvo el título de Magister Universitario en Multimedia y Comunicaciones en la Universidad Carlos III de Madrid – España, en 2011. Actualmente se desempeña como Profesor Agregado a Tiempo Completo en el Departamento de Electrónica, Telecomunicaciones y Redes de Información, y como Coordinador de la Carrera de Ingeniería en Electrónica y Redes de Información.