

MANUAL TECNICO CONSUMO SERVICIO WEB

Contenido

MANUAL TECNICO CONSUMO SERVICIO WEB.....	1
1. INTRODUCCIÓN	2
2. OBJETIVOS	2
2.1. Objetivo General	2
2.2. Objetivos Específicos	2
3. Alcance	2
4. Instalación del Sistema	3
4.1. PRODUCCIÓN	3
5. APLICATIVO	5
5.1. CODIGO FUENTE	5
6. EXTRAS.....	6

1. INTRODUCCIÓN

El servicio Windows para consumo de soap enviando XML fue diseñado para poder realizar las actualizaciones de información de facturas ante la DIAN enviando la información de las facturas una a una mediante la generación de un XML y su correspondiente envío a un servicio web SOAP, recibiendo el CUFÉ y demás data generada por este servicio.

Se crea un servicio Windows que correrá cada minuto tomando las facturas que no se han podido actualizar o que son nuevas.

2. OBJETIVOS

2.1. *Objetivo General*

Mantener actualizadas las facturas frente a la DIAN y mantener la información resultante en el sistema para futuras consultas.

2.2. *Objetivos Específicos*

- Generar un aplicativo que permita correr como un servicio de Windows
- Mantener actualizada la información del sistema frente a la DIAN
- Poder actualizar la información mediante consumo de servicios web

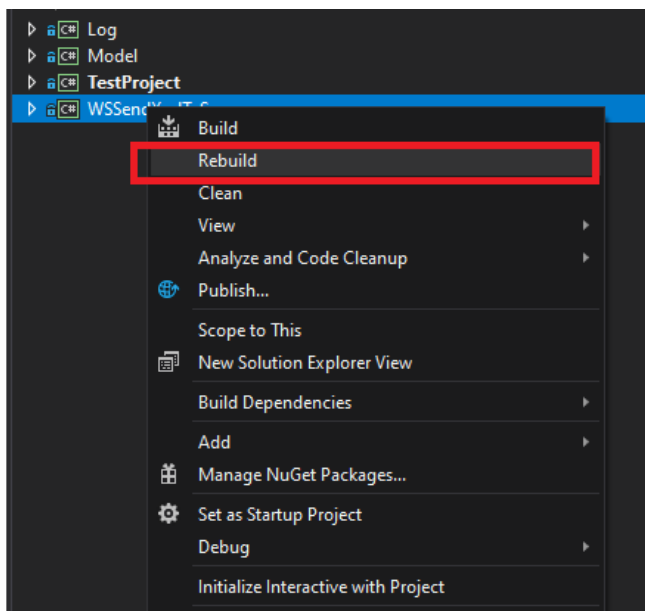
3. Alcance

El alcance de este manual es permitir ilustrar al usuario sobre la instalación del aplicativo y como está estructurado el código. Con esto se busca que se puedan realizar las respectivas modificaciones, añadir nuevas funcionalidades al aplicativo y poder publicar nuevas actualizaciones del mismo.

4. Instalación del Sistema

4.1. *PRODUCCIÓN*

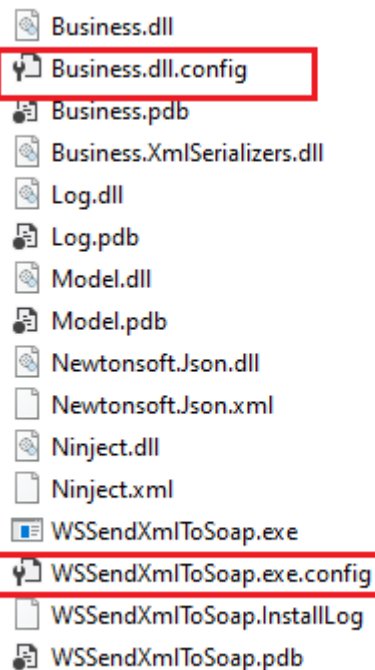
En visual estudio (2019) se debe abrir la solución del proyecto y generar una nueva versión del proyecto WSSendXmlToSoap.



Al ejecutar esta opción (se puede hacer mediante publish) se generará un publicable en la carpeta del proyecto dependiendo si se genera en debug o reléase (se recomienda release)

WSSendXmlToSoapSolution > WSSendXmlToSoap > bin			
Name	Date modified	Type	Size
Debug	3/22/2020 7:59 PM	File folder	
Release	3/24/2020 7:03 PM	File folder	

El contenido de esta carpeta se debe reemplazar en la carpeta donde está instalado el servicio Windows, y cambiar los datos en los archivos de configuración.



En estos dos archivos se encuentran la configuración del endpoint, la ruta de almacenamiento del log y la cadena de conexión de la base de datos (Se recomienda no hacer cambios sobre estos archivos, no copiarlos).

5. APLICATIVO

5.1. *CODIGO FUENTE*

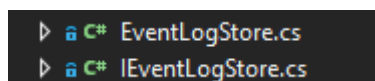
El código fuente será entregado al usuario, junto con este manual para que pueda realizar cambios de una manera más simple. Todo el aplicativo está desarrollado con los siguientes componentes o lenguajes:

- .Net Framework 4.7 C#
- Ninject (Manejo inyección de dependencias)

La aplicación fue hecha como un servicio Windows, igualmente se creó un proyecto adicional de consola para poder realizar las pruebas sin necesidad de instalar el servicio.

En esta solución se crearon 5 proyectos diferentes con el fin de estructurar de una mejor manera el código y sea más fácil su posterior entendimiento y cambios de ser necesarios.

- **Log:** Este proyecto es una librería de clases en la cual se almacena el log del aplicativo completo consta de una clase y su interfaz



```
▶ C# EventLogStore.cs
▶ C# IEventLogStore.cs
```

En esta se tiene la función que creo un log diario y almacena la información que se le envía en cada uno de los pasos o procesos de la aplicación.

- **Business:** Este proyecto es una librería de clases en la cual se maneja la parte lógica de la aplicación, creación de XML, consumo de servicios web, y demás procesos que son necesarios en el aplicativo. Consta de tres partes, las carpetas Interfaces, Soap y las clases que están fuera de estas carpetas.
 - En la carpeta Interfaces se encuentran el total de las interfaces que son usadas en esta librería.
 - En la carpeta Soap se encuentra la clase donde se realizan todas las llamadas al servicio Soap.
 - Las clases que se encuentran fuera de las carpetas representan el proceso base de la aplicación, y la generación del XML a partir de una clase de datos.

- **Model:** Este proyecto es una librería de clases en la cual se tiene el acceso a la base de datos y las estructuras de datos que se usan para manejo de XML, respuestas de servicio Soap y envío de data, en este proyecto se manejan 3 carpetas y una clase fuera de estas.
 - En la carpeta Interfaces se encuentran el total de las interfaces que son usadas en esta librería.
 - En la carpeta Data se tiene el acceso a la base de datos (DbQuery) y la conversión de las tablas devueltas en la base de datos a las estructuras que se manejan en el proyecto, listados de objetos.
 - En la carpeta XmlModel se tiene todos los modelos que se usan para la generación del XML, poniendo sus respectivas data Annotation en cada una de las clases y atributos, si se modifica alguna de estas clases, la estructura del archivo XML que se envía al servicio Soap podría cambiar afectando su rendimiento.
 - La clase SoapResponse es la estructura de respuesta por parte del soap tanto para las funciones básicas (ping) como para el inicio de sesión y envío de XML.
- **WSSendXmlToSoap:** Este es el servicio Windows el cual maneja el tiempo de ejecución del aplicativo, consta de la clase DependencyInjection donde se maneja toda la parte de inyección de dependencia, la clase program donde inicia la aplicación y process donde se inicia el timer y se le configura lo que debe hacer (función(es) que se deben ejecutar).
- **TestProject:** Aplicación de consola para realizar las pruebas sin instalar el servicio.

6. EXTRAS

Este proyecto está configurado para traer la información de una base de datos SQL Server y consumir un endpoint Soap con una estructura especificada, si se llegan a hacer cambios ya sea en la base de datos, o en los inputs u outputs del endpoint será necesario realizarlos en la aplicación para poder asegurar su correcto funcionamiento.