

INTEGRACIÓN DE SOLICITUD DE MENSAJE DE DIAGNÓSTICO Y
HERRAMIENTA DE VISUALIZACIÓN DE DATOS EN TIEMPO REAL PARA
SONDAS LEVEL SHIELD™

JUAN SEBASTIAN BERNAL CUBIDES

UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, COLOMBIA

2017

INTEGRACIÓN DE SOLICITUD DE MENSAJE DE DIAGNÓSTICO Y
HERRAMIENTA DE VISUALIZACIÓN DE DATOS EN TIEMPO REAL PARA
SONDAS LEVEL SHIELD™

JUAN SEBASTIAN BERNAL CUBIDES

MONOGRAFÍA PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO

TUTOR: ING. CARLOS ANDRÉS TORRES PINZÓN

UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, COLOMBIA

2017

HOJA DE ACEPTACIÓN

Juan Sebastian Bernal cubides

Estudiante

Ing. Diego Andres Rodriguez

Gerente de tecnología, SCI Sur America

Ing. Carlos Andres Torres Pinzon

Tutor de proyecto de grado

CONTENIDO

1.	Glosario.....	7
2.	Problema.....	8
3.	Antecedentes	10
4.	Justificación.....	11
5.	Objetivos.....	12
5.1.	Objetivo general	12
5.2.	Objetivos específicos	12
6.	Factibilidad.....	13
7.	Marco teórico.....	14
7.1.	SCI Suramérica SAS	14
7.2.	C#.....	15
7.2.1.	Winform	15
7.2.2.	Web service	15
7.3.	Level Shield™	16
7.3.1.	Proceso de comunicación con las sondas inalámbricas Level Shield™	17
7.3.1.1	Requisitos mínimos para establecer la comunicación	18
7.3.1.2	Esquema de mensaje de diagnóstico (pregunta y respuesta)	19
8.	Diseño metodológico.....	22
8.1.	Etapa de levantamiento de requerimientos	22
8.2.	Etapa de documentación	22
8.3.	Etapa de estructuración del desarrollo.....	22
8.4.	Etapa de desarrollo del proyecto	22
8.5.	Etapa de pruebas y correcciones.....	23
8.6.	Etapa de documentación, entrega y capacitación.....	23
9.	Componente Humanístico	24
10.	Administración del proyecto.....	26
11.	Desarrollo del proyecto	27
11.1.	Integración de solicitud de mensaje de diagnóstico al servicio	28
11.2.	Creación de la herramienta de visualización de datos en tiempo real.....	33

11.2.1.	Pestaña para la comunicación serial RS232 tipo pull	34
11.2.1.1	Botón “Search For Probes In All Serial Com Ports”	35
11.2.1.2	Botón “Request Data To All Configured Probes”	35
11.2.1.3	Botón “Request Diagnostic To All Configured Probes”	36
11.2.1.4	Botón “Request Data”	36
11.2.1.5	Botón “Request Diagnostic”	37
11.2.2.	Pestaña para la comunicación serial RS232 tipo push	38
11.2.2.1	Botón “Open Port And Wait For Data”	39
11.2.3.	Pestaña para la comunicación TCP-IP tipo pull.....	40
11.2.3.1	Botón “Open TCP Connection And Wait For Data”	41
11.3.	Análisis y mejora en el rendimiento del servicio	42
12.	Resultados.....	46
13.	Conclusiones	50
14.	Referencias y bibliografía.....	51
15.	Anexos	53

LISTA DE TABLAS

Tabla 1.	Distribución de tareas por etapas.....	26
Tabla 2.	Estructura tabla dboDiagnosticProbeInformation.....	30
Tabla 3.	Caso de uso botón “Search For Probes In All Serial Com Ports”	35
Tabla 4.	Caso de uso botón “Request Data To All Configured Probes”	35
Tabla 5.	Caso de uso botón “Request Diagnostic To All Configured Probes”	36
Tabla 6.	Caso de uso botón “Request Data”	36
Tabla 7.	Caso de uso botón “Request Diagnostic”	37
Tabla 8.	Caso de uso botón “Open Port And Wait For Data”	39
Tabla 9.	Caso de uso botón “Open TCP Connection And Wait For Data”	41
Tabla 10.	Resultados del análisis al código del servicio	42

LISTA DE FIGURAS

Figura 1. Esquema de funcionamiento básico Level Shield™ [1]	8
Figura 2. Esquema de funcionamiento sondas inalámbricas Level Shield™ [9]	18
Figura 3. Esquema de funcionamiento detallado Level Shield™ [10]	27
Figura 4. Diagrama de flujo proceso de solicitud de mensaje de diagnóstico [11]	29
Figura 5. Ventana de configuración del servicio.....	33
Figura 6. Interfaz gráfica pestaña para la comunicación serial RS232 tipo pull	34
Figura 7. Interfaz gráfica pestaña para la comunicación serial RS232 tipo push.....	38
Figura 8. Interfaz gráfica pestaña para la comunicación TCP-IP tipo pull	40
Figura 9. Resultados de análisis al código del servicio	43
Figura 10. Resultados de análisis específico	44
Figura 11. Resultados de análisis luego de modificaciones.....	45
Figura 12. Valores de diagnóstico mostrados en FleetSAP [13]	46
Figura 13. Datos de nivel visualizados en la herramienta	47
Figura 14. Datos de diagnóstico visualizados en la herramienta.....	47
Figura 15. Datos de diagnóstico visualizados en la herramienta.....	48
Figura 16. Datos de diagnóstico visualizados en la herramienta.....	48

1. GLOSARIO

A continuación se expone una breve definición del autor de algunos términos presentes en el documento.

Servicio de Windows: Un servicio de Windows es un programa de computador que funciona en segundo plano y tiene como finalidad cumplir con una o varias tareas determinadas.

Level Shield™: Producto de la compañía SCI cuya función principal es medir los niveles de combustible en tanques de almacenamiento móviles o estacionarios.

Lantronix: Dispositivo diseñado por la empresa de nombre homónimo cuya principal funcionalidad es convertir una comunicación en protocolo serial a una en protocolo TCP-IP sobre redes cableadas o móviles.

RS232: Estándar de conexión bajo el cual se puede llevar a cabo una comunicación serial.

Checksum: Numero que se obtiene luego de aplicar un determinado procedimiento matemático sobre un grupo de datos y cuya finalidad es verificar la integridad de los mismos.

Script SQL: Archivo con instrucciones para generar un determinado cambio o procedimiento sobre una base de datos.

FleetSAP: Plataforma desarrollada por la compañía SCI que permite a los clientes tener contacto directo con sus productos y obtener informes detallados de los mismos.

2. PROBLEMA

Level Shield™ es un producto de la compañía SCI que permite monitorear niveles de combustible, niveles de agua y temperatura en tanques de almacenamiento. La medición de dichos parámetros es realizada por sondas instaladas directamente en los tanques, posteriormente estas transmiten los datos de manera alámbrica o inalámbrica según el tipo de sonda. Los datos son procesados por un servicio (FMLevelShieldII) instalado en un computador o servidor y posteriormente son almacenados en una base de datos para que luego puedan ser consultados por el cliente en la aplicación web de la compañía. Para el caso de las sondas inalámbricas es posible no solo obtener los datos de nivel y temperatura si no adicionalmente información relativa al estado de la sonda o también llamada información de diagnóstico, tal como intensidad de la señal, número de flotadores instalados, niveles de ruido y nivel de batería entre otros.

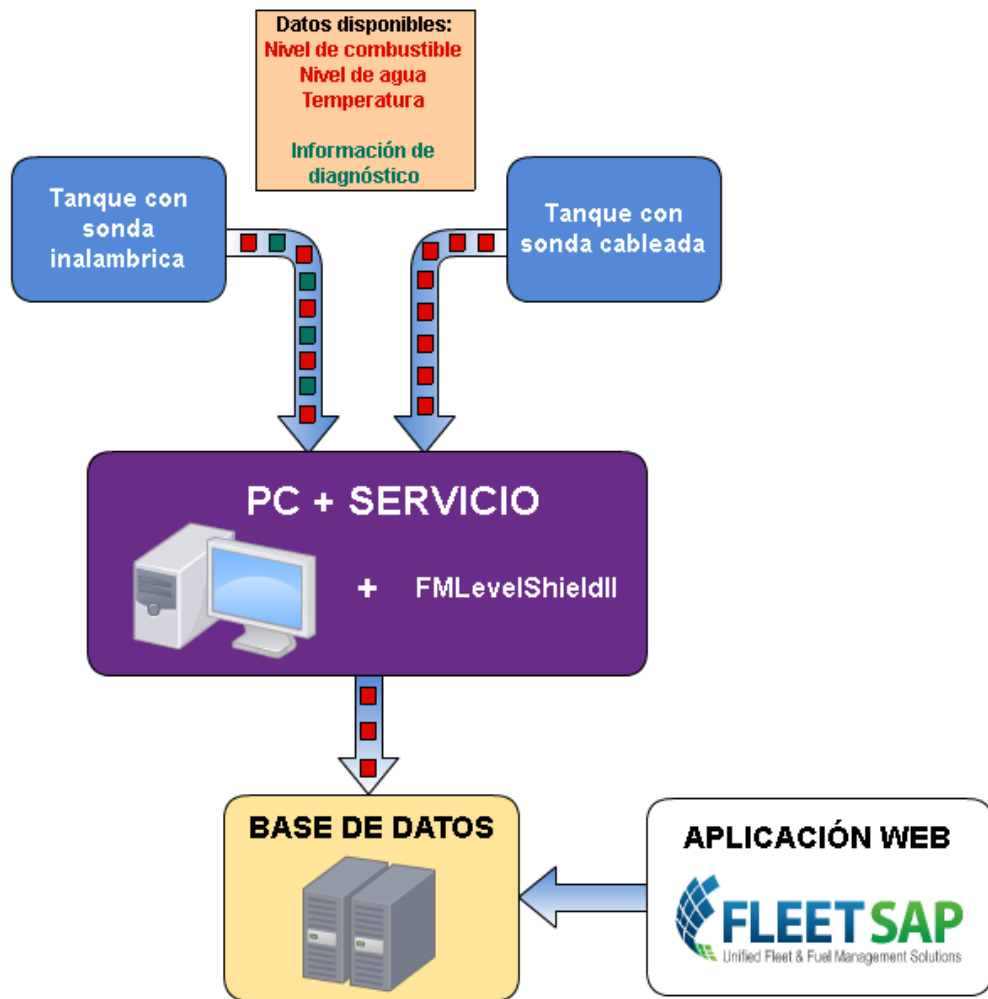


Figura 1. Esquema de funcionamiento básico Level Shield™ [1]

Actualmente el servicio solo se limita a recuperar información de niveles y temperatura, de manera que no se está sacando provecho a todas las funcionalidades del dispositivo y se está ignorando información valiosa que permite evitar problemas tales como transmisiones defectuosas por elevados niveles de ruido, pérdidas de información por nivel crítico de batería, etc. Se plantea entonces integrar al servicio la capacidad de solicitar y procesar la información de diagnóstico para luego almacenarla en la base de datos y que pueda ser consultada en la plataforma web por el cliente.

Teniendo en cuenta este nuevo desarrollo se determina que también sería de gran utilidad disponer de esta información de diagnóstico en tiempo real, sin depender de la base de datos o de la plataforma web, pues esto permitirá verificar el correcto funcionamiento e instalación de las sondas y facilitaría la identificación de problemas en campo. Se agrega entonces como requerimiento para el proyecto, la integración de una herramienta que permita obtener la información de diagnóstico y niveles de las sondas en tiempo real sin depender de la existencia de la base de datos u otros elementos externos.

3. ANTECEDENTES

El servicio FMLevelShieldII se encarga de comunicarse con las sondas bajo diversos protocolos de comunicación para de esta manera poder recuperar la información que estas ofrecen. Las sondas están en la capacidad de enviar diferentes tipos de información según el comando que reciban de parte del servicio. En la actualidad el servicio solo cuenta con la posibilidad de enviar el comando bajo el cual se solicita la información de niveles y temperatura a las sondas, sin embargo estas pueden recibir otros comandos y enviar otro tipo de datos tal como la información de diagnóstico.

Teniendo en cuenta el desarrollo ya existente en el servicio se agregará a este la posibilidad de manejar el comando que obtiene la información de diagnóstico, de igual manera es necesario adicionarle los diversos procedimientos para que pueda procesar la respuesta de las sondas y organizarla de la manera adecuada. Adicionalmente basándose en los ya existentes parámetros que se tienen para el manejo de la base de datos y los servicios web, se realizarán las adiciones y modificaciones que sean pertinentes para de esta manera permitir el correcto funcionamiento del servicio y sus nuevas características.

Para el desarrollo de la herramienta que permita visualizar los datos de las sondas en tiempo real se usarán los parámetros que se manejaron para la realización de la consola de configuración del servicio, se buscará una integración que esté acorde con el formato existente y se planea mostrar los datos de la manera más adecuada para así lograr una interpretación clara que no se preste para ambigüedades y que sea transparente para los que la analizan.

4. JUSTIFICACIÓN

En el mundo actual, la globalización y la apertura de mercados han generado gran cantidad de competencia en todas las industrias, esto ha llevado a que cada vez sea más complicado fidelizar al cliente, por lo que se deben implementar estrategias para ofrecer un mayor valor agregado a los productos, al igual que se debe asegurar funcionalidad continua y eficacia a la hora de solucionar problemas.

La integración de la solicitud de diagnóstico contribuye a dar valor agregado al producto pues se cuenta con más funcionalidad y versatilidad, además permite asegurarle al cliente una mayor estabilidad en el funcionamiento, ya que con esta se posibilita la detección temprana de situaciones que a largo plazo pueden generar pérdidas de información o fallas en las sondas.

Adicionalmente la herramienta de visualización de datos en tiempo real permitirá facilitar el trabajo a instaladores en campo y personal de soporte técnico, ya que con ella podrán realizar pruebas parciales sin dependencias externas, lo cual contribuirá a detectar problemas de instalación más fácilmente o a identificar posibles fallas sin necesidad de usar software alternativo o tener que remitirse a documentación del fabricante, logrando así un mayor bienestar para el personal de planta y contratistas.

5. OBJETIVOS

5.1. Objetivo general

Integrar la funcionalidad de solicitud de diagnóstico de las sondas Level Shield™ al servicio FMLevelShieldII e incluirla en una herramienta que permita visualizar los datos de las sondas en tiempo real, sin ninguna dependencia externa, que sea de fácil manejo, intuitiva y que permita reducir los tiempos de respuesta en servicio al cliente, favoreciendo a los técnicos e instaladores en campo y a las personas que prestan asistencia remota, aumentando la productividad interna y ofreciendo mayores beneficios al cliente.

5.2. Objetivos específicos

- Reducir los tiempos de respuesta en situaciones de diagnóstico de fallas en instalaciones de sondas Level Shield™.
- Reducir el número de fallas reportadas por los clientes detectando de manera temprana situaciones que puedan generarlas y corrigiéndolas de manera adecuada.
- Aumentar la satisfacción de los clientes respecto al producto ofreciendo nuevas funcionalidades y mayor estabilidad.
- Mejorar el rendimiento del servicio identificando y solucionando malas prácticas de programación en el código fuente, con el fin de reducir la carga de procesamiento en el servidor y en la base de datos.

6. FACTIBILIDAD

Para llevar a buen término los objetivos planteados en el proyecto y lograr solucionar el problema que se presenta, es necesario tener conocimiento suficiente en tres componentes primordiales, bases de programación y manejo de lenguaje C#, manejo y comprensión del código fuente del servicio (software) y claridad en el funcionamiento del dispositivo (hardware).

En cuanto a las bases y manejo de lenguajes de programación, la universidad durante el periodo de formación proporcionó bastantes conocimientos que permiten desarrollar software eficiente y funcional, contribuyendo al desarrollo del razonamiento y la lógica, cualidades necesarias para llevar a buen término proyectos relacionados con programación. Adicionalmente la empresa proporcionó un entrenamiento centrado en C# y sus principales características, que se aprovechó de la mejor manera pues se articulaba perfectamente con las bases adquiridas en el pregrado.

Luego de obtener los conocimientos para entrar a trabajar con el desarrollo, se debe proceder a estudiar y analizar el código fuente del servicio para tener plena conciencia de dónde y de qué manera se deben realizar los cambios y adiciones pertinentes, dicha tarea puede llevarse a cabo de manera satisfactoria pues el código está bien estructurado y comentado y adicionalmente se posee asesoría de otros desarrolladores que lo conocen a fondo.

En cuanto al manejo y conocimiento del dispositivo, SCI proporciona variedad de herramientas entre las cuales se encuentran simuladores, sondas de pruebas y manuales, que en conjunto permitirán definir de manera clara el comportamiento del mismo, facilitando el proceso de desarrollo al tener conciencia de cómo se debe manejar la comunicación con el dispositivo.

Es claro entonces que para el desarrollo del proyecto se tiene una factibilidad positiva pues se cuentan con las herramientas, conocimientos y apoyo necesarios para llevar a cabo las tareas y requerimientos en base a los tres pilares anteriormente enunciados.

7. MARCO TEÓRICO

A continuación se hará una breve presentación de algunos elementos relevantes en el desarrollo del proyecto.

7.1. SCI Suramérica SAS

Compañía que forma parte de la empresa SCI Distribution LLC con casa matriz en Clearwater, Florida en los Estados Unidos, se especializa en ofrecer soluciones inalámbricas para el control y monitoreo de flota y despacho de combustible a organizaciones estatales y privadas. La compañía cuenta con presencia en el Caribe, Europa, África, Oriente Medio, Rusia, India, Indonesia y las Filipinas [2].

Desde la sede en Colombia se realiza la distribución, venta y soporte de los productos en América Latina, de igual manera en el país se encuentra la división de investigación y desarrollo de la compañía, desde la cual se llevan a cabo todos los procesos de creación y modificación de software y el manejo de la plataforma web de consulta de la compañía conocida como FLEETSAP.

Algunas de las soluciones que cubre la compañía con sus productos son :

- Identificación de vehículos por radio frecuencia.
- Monitoreo de nivel de tanques de combustible.
- Monitoreo de condiciones de vehículos mediante GPS.
- Monitoreo de entradas y salidas de combustible en estaciones de servicio.
- Monitoreo de presión de aire y temperatura en las llantas.
- Monitoreo en vivo de estaciones de servicio, tanques y vehículos mediante video y audio.

Gracias a estas funcionalidades los productos de SCI han contribuido a reducir los robos de combustible, los costos de compra del mismo y los costos de mantenimiento de flota, adicionalmente con sus productos la compañía asegura el aumento en la utilización de la flota, aumentando la productividad, la seguridad las ganancias de las empresas [3].

7.2. C#

Teniendo en cuenta que el desarrollo realizado es una adición de funcionalidades a un servicio ya existente, se debe adquirir conocimiento en la manera en que está desarrollado dicho servicio. En SCI los programas se desarrollan en C#.

C# es un lenguaje de programación orientado a objetos creado en el año 2000 por Microsoft, con el fin de mejorar el desarrollo de aplicaciones, buscando que estas fuesen accesibles desde cualquier parte (basadas en la web) y desde cualquier tipo de dispositivo (soporte en móviles). C# fue diseñado específicamente para la plataforma .NET de Microsoft con la intención de facilitar la migración a esta, unificando las bondades de otros lenguajes de programación tales como C, C++ y Java y adicionando nuevas funcionalidades para añadirle versatilidad. [4]

7.2.1. Winform

Dado que el proyecto incluye la creación de una interfaz gráfica para visualizar los nuevos datos obtenidos, se deben estudiar las herramientas que dispone C# para realizar esta tarea. Las interfaces gráficas para C# se conocen como *WinForms*.

WinForms (Windows Forms) es el nombre que se le da a la interfaz de programación de aplicaciones graficas (API) de .NET, dicha herramienta provee un ambiente de desarrollo basado en las formas y controles nativos de Windows, ofreciendo acceso mediante código a las propiedades y acciones de estos. Adicionalmente, las *WinForms* con C# ofrecen una característica especial, permiten implementar funcionalidades basadas en HTML cumpliendo de esta manera, con los requerimientos iniciales de C# en cuanto a accesibilidad desde cualquier parte. [5]

7.2.2. Web service

Luego de obtener los conocimientos necesarios para desarrollar código e interfaces graficas mediante C#, es necesario aprender la forma en que el servicio se comunica a través de la web, con la base de datos donde se almacenan los registros que se procesan. Para este proceso se utilizan web services.

Un web service es un componente de software almacenado en un equipo, al cual se puede acceder desde una aplicación alojada en otro equipo por medio de una red. La máquina en la cual reside el web service se conoce como equipo remoto. La aplicación que accede al web service llama a un método sobre internet (web method) en el equipo remoto, este procesa el llamado y devuelve una respuesta sobre la red a la aplicación. El uso de web services facilita la comunicación entre distintas aplicaciones en diferentes orígenes, mediante un código sencillo y bajo un protocolo de internet estándar. [6]

Los requerimientos para la creación y correcto funcionamiento de un web service son [7]:

- **Estándar de mecanismo para llamado a procedimiento remoto:** este establece el procedimiento o protocolo para que una maquina pueda ejecutar una tarea en otra, sin necesidad de preocuparse por la manera en que estas se comunican.
- **Estándar de formato de datos:** como lo indica el nombre, establece el formato en el que se van a intercambiar los datos entre las dos máquinas, debe contemplar todo tipo de datos (texto, video, imágenes, audio, etc).
- **Estándar para describir el lenguaje del servicio:** este establece los requerimientos en cuanto a entrada y salida de datos para el web service.
- **Estándar de descubrimiento del servicio web:** como lo indica el nombre, este ayuda al programador a que su web service sea dado a conocer a los interesados en su uso.

Para el caso del web service que se maneja en SCI se tiene un web service de tipo SOAP, los estándares que este maneja son:

- SOAP como mecanismo para llamado a procedimiento remoto
- XML como formato de datos
- WSDL para describir el lenguaje del servicio
- UDDI para dar a conocer el web service

7.3. Level Shield™

Es un sistema de monitoreo de nivel de tanques de combustible móviles y estacionarios que cuenta con la capacidad de generar alertas en diversas situaciones configurables por el usuario, adicionalmente posee un software inteligente de compra de combustible que permite al poseedor reducir consistentemente los costos haciendo uso de sistemas de subasta buscando siempre al mejor oferente [8].

El sistema de monitoreo de tanques Level Shield™ posee tres elementos esenciales, una sonda (alámbrica o inalámbrica) con dos flotadores instalada en el interior del tanque que se encarga de medir los niveles de agua, combustible y temperatura, un elemento (o dos en algunos casos) que permite comunicar la sonda con el computador y un servicio instalado en un computador o servidor, que se encarga de solicitar los datos de la sonda, procesarlos y subirlos a una base de

datos para que posteriormente estos puedan ser consultados desde la plataforma FLEETSAP.

Algunos de los beneficios de usar el sistema de monitoreo de tanques Level Shield™ son:

- Eliminación de robos de combustible en los tanques.
- Identificación de presencia de agua en el combustible.
- Verificación física de entrega de combustible y comparación en tiempo real con órdenes de compra generadas desde la aplicación.
- Análisis de facturación, impuestos y recargos.
- Alertas de alto nivel, bajo nivel y nivel crítico totalmente configurables.
- Generación automática de subastas para compra de combustible mediante órdenes de compra.
- Uso de un avanzado algoritmo de compra según precio y necesidad.

7.3.1. Proceso de comunicación con las sondas inalámbricas Level Shield™

Teniendo en cuenta que el desarrollo realizado se limitaba a las sondas inalámbricas, se analizará más a fondo la comunicación que se realiza entre el servicio y las mismas.

La sonda inalámbrica va ligada automáticamente a un receptor RF, este receptor será el que recibirá los comandos necesarios para recuperar la información de las sondas, los transmitirá a estas y posteriormente recuperará las respuestas correspondientes.

Dependiendo del tipo de instalación y las condiciones del sitio, existen diversas maneras de comunicar al receptor con el computador que tiene instalado el servicio, estas son:

- Usar un cable serial entre el receptor y el computador, de esta manera se establecerá una comunicación serial RS232 para el intercambio de la información.
- Conectar el receptor a un dispositivo Lantronix Wifi mediante un cable serial, el dispositivo Lantronix llevará esta comunicación serial a una comunicación TCP sobre una red local. Posteriormente mediante un software conocido como CPR o COM PORT Redirector (instalado en el computador que posee el servicio), redireccionar dicha comunicación TCP a un puerto serial virtual. En este caso, a pesar de tener elementos intermedios entre el receptor y el computador, aún se tiene en esencia una comunicación serial RS232 como en el caso anterior.

- Conectar el receptor a un dispositivo Lantronix GPRS mediante un cable serial, el dispositivo Lantronix llevará esta comunicación serial a una comunicación TCP sobre una red móvil. Posteriormente en el computador que tiene instalado el servicio se configura un puerto de red público, de esta manera la Lantronix pueda conectarse mediante un túnel TCP. Para este caso se tiene en esencia una comunicación TCP para el intercambio de la información.

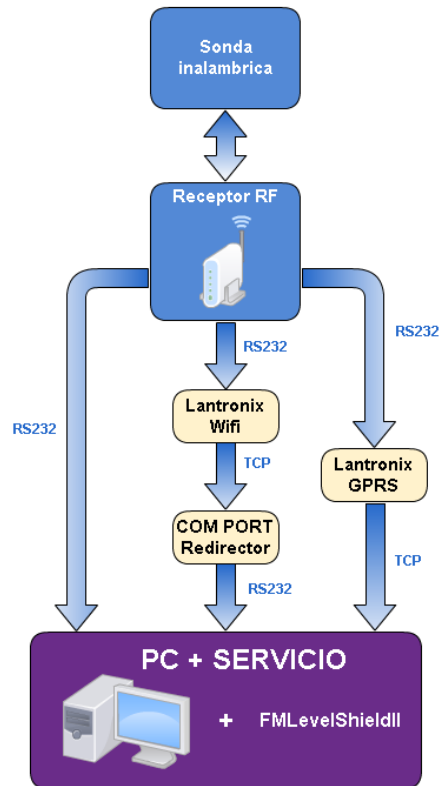


Figura 2. Esquema de funcionamiento sondas inalámbricas Level Shield™ [9]

A pesar de que existen varios métodos físicos para comunicar el receptor con el computador que posee el servicio, solo se utilizan dos protocolos de comunicación: serial RS232 y túnel TCP-IP.

7.3.1.1 Requisitos mínimos para establecer la comunicación

Para establecer una comunicación serial RS232 entre el servicio y el receptor y recuperar la información adecuada, se deben cumplir con los siguientes requisitos:

- Puerto COM (físico o virtual) ligado al receptor
- Velocidad de transmisión: 9600 bits por segundo

- Paridad: ninguna
- Bits de datos: 8 bits
- Bits de parada: un bit
- Si se tiene Lantronix WIFI, que esta esté conectada a la misma red del computador que tiene el COM PORT Redirector y el servicio.

Para establecer una comunicación mediante túnel TCP-IP entre el servicio y el receptor y recuperar la información adecuada, se deben cumplir con los siguientes requisitos:

- Dirección IP y puerto públicos que permitan conexiones externas
- SIM card con datos y alcance de red móvil en la Lantronix GPRS.

7.3.1.2 Esquema de mensaje de diagnóstico (pregunta y respuesta)

Para obtener el mensaje de diagnóstico de las sondas se debe enviar un mensaje de pregunta con el siguiente formato:

Dxxxxx + Char(13)

En total son siete caracteres repartidos así:

- La letra D mayúscula [un carácter]
- Un numero de cinco cifras (representado por xxxxx) que hace referencia al identificador único de la sonda [cinco caracteres]
- Carácter ASCII numero 13 también conocido como “Carriage return” [un carácter]

La respuesta obtenida llegará con el siguiente formato:

*xxxxxDaaa=bbbbbbb=ccccccc=ddddddd=eee=fff=ggg=hhh=iii=jjj=k
kk=lll=mmm=nnn=zzz + Char(10) + Char(13)*

En total son ochenta y dos caracteres distribuidos de la siguiente forma:

- Un numero de cinco cifras (representado por xxxxx) que hace referencia al identificador único de la sonda [cinco caracteres]
- La letra D mayúscula [un carácter]

- Un numero de tres cifras (representado por aaa) que hace referencia a la ventana de la señal [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de ocho cifras (representado por bbbbbbbb) que hace referencia a la cantidad de mensajes enviados por la sonda desde el ultimo reinicio [ocho caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de ocho cifras (representado por cccccccc) que hace referencia a la cantidad de mensajes enviados por la sonda desde su instalación [ocho caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de ocho cifras (representado por dddddddd) que hace referencia a la cantidad de mensajes recibidos por el receptor [ocho caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por eee) que hace referencia a la cantidad de flotadores que tiene la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por fff) que hace referencia al canal de transmisión configurado en la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por ggg) que hace referencia a la potencia de la señal inalámbrica entre la sonda y el receptor [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por hhh) que hace referencia al modo de operación configurado en la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por iii) que hace referencia a la ganancia en decibels de la señal local [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por jjj) que hace referencia a la ganancia en decibels del ruido local [tres caracteres]

- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por kkk) que al ser dividido entre 21.1 hace referencia al voltaje del receptor [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por III) que hace referencia a la ganancia en decibels de la señal en la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por mmm) que hace referencia a la ganancia en decibels del ruido en la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un numero de tres cifras (representado por nnn) que al ser dividido entre 21.1 hace referencia al voltaje de la batería de la sonda [tres caracteres]
- El carácter igual (=) como separador [un carácter]
- Un número de tres cifras (representado por zzz) que hace referencia al checksum del mensaje de respuesta [tres caracteres].

El checksum se calcula sumando el valor ASCII de los primeros 77 caracteres del mensaje de respuesta (se obvia el checksum y los dos caracteres especiales de final) y luego haciendo modulo con 255 al resultado de la suma.

- Carácter ASCII numero 10 también conocido como “Line feed” [un carácter]
- Carácter ASCII numero 13 también conocido como “Carriage return” [un carácter]

8. DISEÑO METODOLÓGICO

Con el fin de llevar a cabo el proyecto y cumplir con los objetivos de manera exitosa es necesario definir distintas etapas que lo estructuren y organicen. Al ser un proyecto basado en desarrollo de software se debe iniciar con una etapa de definición de requerimientos, posteriormente vendrá una etapa de documentación seguida de una etapa de estructuración del desarrollo. Luego estará la etapa de desarrollo del proyecto que vendrá seguida por una etapa de pruebas y correcciones y se finalizará con la etapa de documentación, entrega y capacitación.

8.1. Etapa de levantamiento de requerimientos

En esta etapa se definirá de manera clara que se va a realizar para así tener claro el horizonte del proyecto, los objetivos finales y las necesidades que se deben cubrir con el mismo. Para dicha etapa se contará con el apoyo del director del área de desarrollo que es quien conoce las falencias del producto y los alcances que debe tener el mismo. La meta será lograr que el producto final entregado cumpla con todos los lineamientos que se definieron en esta etapa.

8.2. Etapa de documentación

En esta etapa se realizará una búsqueda de información encaminada en lograr dos objetivos, familiarizarse con el software y hardware involucrado en el desarrollo y solidificar los conocimientos referentes a la manera de desarrollar y los estándares usados en la compañía. Para llevar a cabo esta etapa se hará uso de documentación del fabricante del dispositivo y del código fuente del servicio sobre el cual se va a trabajar, también se usará documentación de estándares y recomendaciones, manejo de bases de datos y servicios web creada dentro de la compañía.

8.3. Etapa de estructuración del desarrollo

Luego de tener claro que se necesita hacer y qué recursos se tienen para hacerlo se pasará a la presente etapa en la cual se define como se va a realizar el proyecto. En este paso serán muy útiles diagramas que ilustren claramente el proceso que va a seguir el servicio para realizar sus tareas y bosquejos de las interfaces gráficas en caso de que aplique.

8.4. Etapa de desarrollo del proyecto

Posterior a la definición del qué, el con qué y el cómo, se procederá con el desarrollo del proyecto entrando a realizar los cambios que sean pertinentes en el código del servicio. Adicionalmente se debe tener en cuenta los factores externos tales como base de datos y servicios web que sean necesarios para lograr cumplir con los requerimientos definidos en la primera etapa.

8.5. Etapa de pruebas y correcciones

Al finalizar el desarrollo se continuará con la etapa de pruebas, en esta se verificará que los requerimientos estén cubiertos completamente y adicionalmente se verificará que los cambios no afecten ningún proceso anterior, también se validará que el desarrollo sea capaz de manejar errores. De ser necesario se realizarán los cambios pertinentes y se repetirán las pruebas hasta obtener un resultado satisfactorio.

8.6. Etapa de documentación, entrega y capacitación

En esta etapa inicialmente se realizará la documentación (manuales y guías de uso) que explicara el uso de las nuevas funcionalidades del servicio, posteriormente se hará entrega del servicio con sus nuevos cambios y la documentación anteriormente realizada. Se realizará un monitoreo constante de las instalaciones y el uso del desarrollo y de ser necesario se ofrecerán capacitaciones a las personas que así lo requieran.

9. COMPONENTE HUMANISTICO

La universidad Santo Tomas ofrece al estudiante, no solo conocimientos técnicos y habilidades para ser competitivos en el ambiente específico de cada carrera, si no que adicionalmente, brinda diversas cátedras que contribuyen con la formación integral de la persona. Dentro de estas cátedras, se resalta la importancia de entregar un beneficio a la sociedad desde el ámbito específico de cada profesión, así como de buscar el crecimiento en sociedad, ayudando a todos aquellos que se mueven en nuestro entorno.

Particularmente en la ingeniería electrónica, este enfoque adquiere especial importancia dado que todo el trabajo realizado en esta profesión va encaminado a mejorar procesos realizados por personas. En ese orden de ideas cuando se realizan desarrollos, el ingeniero electrónico debe procurar el mayor beneficio para todos los involucrados, facilitando los procesos de los trabajadores y minimizando inconvenientes con el fin de generar herramientas que hagan más fácil y productiva la labor de los colaboradores de la compañía.

El trabajo que se expone en esta monografía no está exento de dichas consideraciones, el resultado final otorga gran cantidad de beneficios no solo a la compañía y sus empleados, sino además a los distribuidores y clientes que están involucrados con el producto.

Básicamente son tres ámbitos los que enmarcan dichos beneficios. En primer lugar el resultado final del trabajo realizado afecta directamente al personal de soporte e instalaciones de la compañía y de los distribuidores asociados, pues con las soluciones entregadas se facilita el trabajo que estas personas realizan, ya que se reducen considerablemente los tiempos que deben emplear en solucionar problemas con el producto en cuestión y se acelera el proceso de instalación en campo. Estos beneficios se ven reflejados en el aumento de la productividad de los trabajadores y el mejoramiento del ambiente laboral y la calidad de vida pues se reducen las situaciones de tensión y estrés frente a problemas con los productos.

El segundo ámbito que se ve beneficiado es el del cliente que adquiere el producto, ya que con los cambios realizados se aumentan las prestaciones y funcionalidades que éste estaba recibiendo sin aumentar el precio, de esta forma el cliente termina recibiendo un producto más completo a un precio competitivo. Adicionalmente se mejora la satisfacción frente al producto y la compañía generando así la posibilidad de que la empresa consiga más negocios y aumente la participación en el mercado, lo que favorece el crecimiento de la compañía, la generación de empleos y una afectación positiva frente a la sociedad.

Finalmente, con el mejoramiento del rendimiento en el producto se afectó de manera directa al medio ambiente, pues se realiza un mejor uso de los recursos como la energía eléctrica y se aumenta la vida útil de los equipos, lo cual reduce la generación de basura electrónica que impacta negativamente al ambiente. En general los beneficios que adquiere el ambiente en dónde se desarrolla este proyecto son importantes dado que reducen costos, aumentan efectividad y mejoran un producto que busca mayor eficiencia en el uso de los combustibles fósiles.

10. ADMINISTRACIÓN DEL PROYECTO

Tabla 1. Distribución de tareas por etapas

NÚMERO	TAREA	DURACIÓN (DÍAS)	PREDECESORES
1	Eta de levantamiento de requerimientos	2	-
1,1	Reunión con líder de equipo de desarrollo para definir necesidades y objetivos finales del desarrollo	1	-
1,2	Creación del documento de especificaciones	1	1,1
2	Eta de documentación	15	1
2,1	Entrenamiento en programación orientada a objetos en C#	5	1,2
2,2	Desarrollo de proyecto de prueba para verificar conocimientos adquiridos	5	2,1
2,3	Familiarización con el dispositivo y el código del servicio	5	2,1
3	Eta de estructuración del desarrollo	5	2
3,1	Creación de diagramas que ilustren el funcionamiento de los cambios a agregar	2	2,3
3,2	Creación de bosquejos del aspecto de la interfaz de la herramienta grafica que se va a crear	2	3,1
3,3	Definición de los cambios a realizar en base de datos y servicios web	1	3,2
4	Eta de desarrollo del proyecto	25	3
4,1	Versionamiento de base de datos y servicio web	3	3,3
4,2	Ejecución de los cambios en el servicio y su consola	22	4,1
5	Eta de pruebas y correcciones	10	4
5,1	Realización de pruebas unitarias e integradas	7	4,2
5,2	Modificación y corrección de errores	2	5,1
5,3	Pruebas finales	1	5,2
6	Eta de documentación, entrega y capacitación	15	5
6,1	Realización de manuales	3	5,3
6,2	Entrega y liberación de los cambios	1	6,1
6,3	Monitoreo de las instalaciones y capacitación (si es requerida)	11	6,2

Para mayor detalle remitirse al Anexo A que contiene el diagrama de Gantt.

11. DESARROLLO DEL PROYECTO

Al empezar la pasantía se realizó un entrenamiento intensivo en programación orientada a objetos y C#, se estudiaron gran variedad de conceptos referentes a estos dos temas y posteriormente se llevaron a la práctica en un proyecto de prueba que involucraba adicionalmente, creación de formularios Windows Form y manejo de bases de datos y web service, componentes primordiales en todos los desarrollos que se realizan en la compañía.

El siguiente proceso se centró en la familiarización con el dispositivo y el código del servicio sobre el cual se iba a empezar a trabajar. Inicialmente se realizó la lectura del manual de las sondas (entregado por el fabricante), para identificar el protocolo de comunicación y los diferentes mensajes que el dispositivo puede manejar. Posteriormente se realizó la lectura de los manuales internos con el fin de conocer la arquitectura de la solución Level Shield™, en base a esto se elaboró un diagrama detallado (figura 3) de todas las alternativas de funcionamiento del servicio con las sondas.

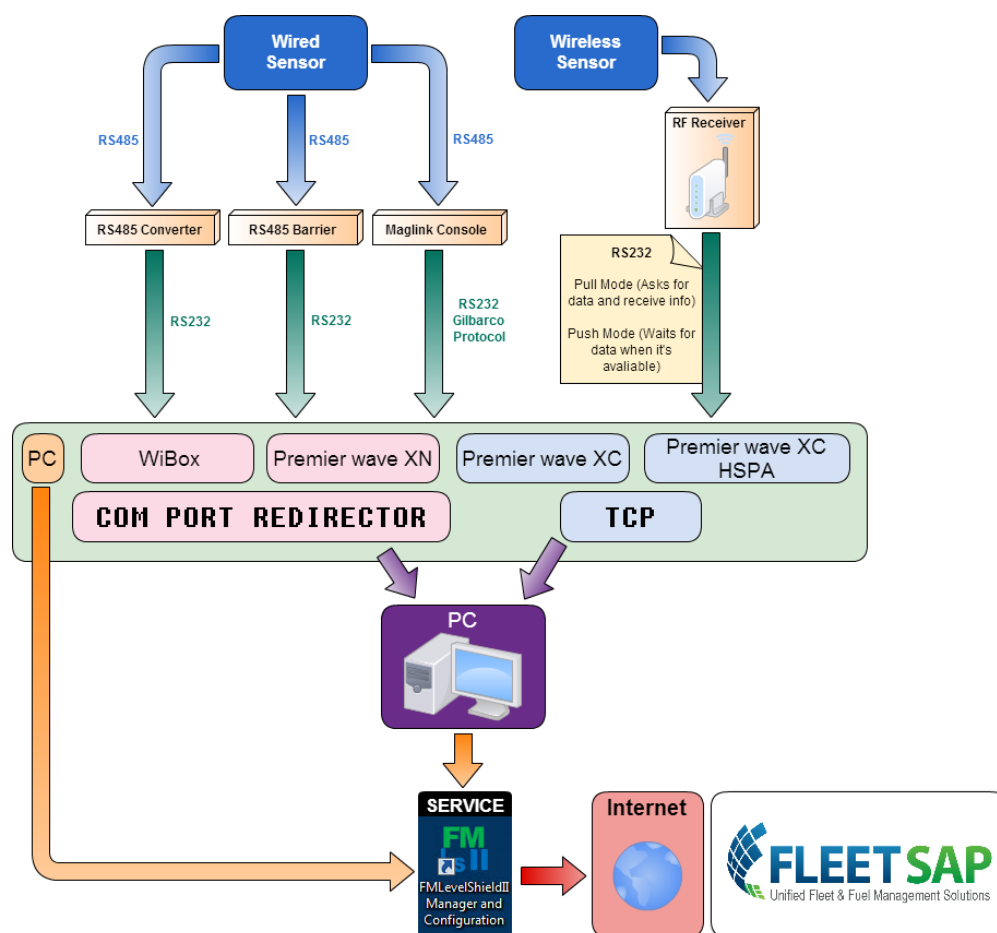


Figura 3. Esquema de funcionamiento detallado Level Shield™ [10]

Luego de finalizar el entrenamiento y la capacitación básica se procedió a organizar el trabajo que se debía realizar para dar buen término al proyecto. Teniendo en cuenta los objetivos planteados, se separa todo el desarrollo en tres grandes tareas, a saber: Integración de solicitud de mensaje de diagnóstico al servicio, creación de la herramienta de visualización de datos en tiempo real y análisis y mejora en el rendimiento del servicio.

11.1. Integración de solicitud de mensaje de diagnóstico al servicio

Teniendo en cuenta que el servicio ya tiene un procedimiento para preguntar a las sondas y esperar por una determinada respuesta, se establece que no es necesario modificar la estructura general del servicio. Se define que se debe incluir código nuevo, que permita al servicio enviar la solicitud de diagnóstico tal como se envía la solicitud de datos de nivel, esperar por respuesta, recibir la información, procesarla y publicarla en la base de datos, para luego continuar preguntando por niveles tal como se hace normalmente.

Específicamente, el proceso que se va a agregar debe hacer lo siguiente:

- Preguntar a las sondas por la información de diagnóstico enviando el respectivo mensaje.
- Esperar por la respuesta de las sondas y si no recibe nada informar de la situación (usar logs del servicio).
- Si recibe información, verificar la integridad de la misma (checksum), procesarla y separarla obteniendo los valores para cada propiedad del diagnóstico.
- Subir la información de diagnóstico a la base de datos.
- Informar en logs que el proceso finalizo correctamente.

En la figura 4 puede observarse el diagrama de flujo con el proceso anteriormente descrito.

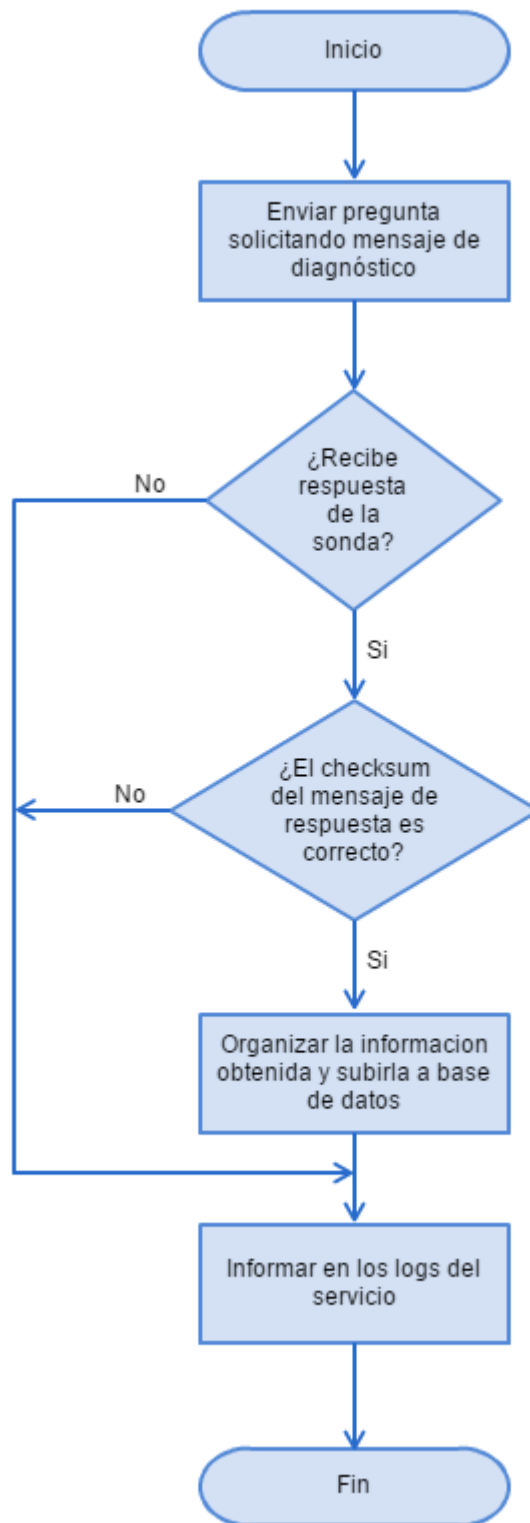


Figura 4. Diagrama de flujo proceso de solicitud de mensaje de diagnóstico [11]

Como prerequisites para implementar el código que realice este proceso, se debe crear una tabla en la base de datos de la compañía, dicha tabla almacenará la información de diagnóstico de todas las sondas así como los identificadores necesarios para poder asociar la información de cada sonda a los clientes fácilmente. El nombre de la tabla será *dboDiagnosticProbeInformation*, la estructura de la misma se ilustra en la tabla 2.

Tabla 2. Estructura tabla *dboDiagnosticProbeInformation*

Nombre Campo	Tipo de dato	Descripción	Otros
IMEI	Cadena de caracteres (15)	Identificador de la sonda	• No admite nulos. • Llave primaria
VehicleID	Cadena de caracteres (20)	Nombre del tanque asociado a la sonda	• No admite nulos. • Llave primaria
CompanyID	Entero	Identificador de la compañía	• No admite nulos. • Llave primaria
LastReceived	Fecha y hora	Fecha y hora en la que se recibió el mensaje de diagnóstico	• Admite nulos.
WindowSignal	Entero	Indica la ventana de la señal	• Admite nulos.
PacketsTransmitted	Entero	Indica la cantidad de paquetes transmitidos por la sonda	• Admite nulos.
CounterOfPackets	Entero	Indica la cantidad de paquetes que ha transmitido la sonda desde que está encendida	• Admite nulos.
PacketsReceived	Entero	Indica la cantidad de paquetes recibidos por el receptor	• Admite nulos.
NumberOfFloats	Entero	Indica el número de flotadores en la sonda	• Admite nulos.
TransmissionChannel	Entero	Indica el canal por el cual la sonda está transmitiendo	• Admite nulos.

SignalPower	Entero	Indica la potencia de la señal	• Admite nulos.
OperatingMode	Entero	Indica el modo de operación de la sonda	• Admite nulos.
LevelLocalsignal	Entero	Indica la ganancia (en decibeles) de la señal local	• Admite nulos.
LevelLocalNoise	Entero	Indica la ganancia (en decibeles) del ruido local	• Admite nulos.
ReceiverTensionA	Float	Indica la tensión (en voltios) del receptor	• Admite nulos.
LevelProbesignal	Entero	Indica la ganancia (en decibeles) de la señal de la sonda	• Admite nulos.
LevelProbeNoise	Entero	Indica la ganancia (en decibeles) del ruido de la sonda	• Admite nulos.
ReceiverTensionB	Float	Indica la tensión (en voltios) del receptor	• Admite nulos.

Para la creación de la tabla se genera un script que se envía al encargado de base de datos para aprobación, cuando este es aprobado se versiona en una base de datos de prueba y posteriormente en la base de datos oficial de la compañía.

Luego de creada la tabla, se crean tres nuevos métodos en el web service para manejar los datos que se van a almacenar en la misma. Se establece que solo se va a mantener un registro de diagnóstico por cada sonda, es decir que cada que se recibe un diagnostico nuevo se va a sobrescribir el existente o se va a crear en caso de que no exista. Teniendo en cuenta esto se crean tres nuevos métodos en el web service:

- GetDiagnoseByIMEI: este método permite buscar un registro de diagnóstico dentro de la tabla para una determinada sonda.
- ModifyDiagnosticsProbeInformation: este método permite sobrescribir un registro de diagnóstico dentro de la tabla para una determinada sonda.
- AddDiagnosticsProbeInformation: este método permite adicionar un registro de diagnóstico dentro de la tabla para una determinada sonda.

Los métodos de web service pasan por una revisión del líder de desarrollo web y cuando son aprobados, se publican en el web service de pruebas y en el web service oficial de la compañía.

Con los prerequisites cubiertos, se procede a implementar el código en el servicio para poder recibir el diagnóstico de las sondas. Primero se define la creación del mensaje a enviar a las sondas, para ello se crea la clase *cLevelShieldMessage*. Esta clase se encarga de crear el mensaje teniendo en cuenta el identificador de las sondas y el tipo de información que se desee obtener (nivel o diagnostico). Con la creación de esta clase se añadió versatilidad al servicio, pues anteriormente el único mensaje que se manejaba (solicitud de nivel) se creaba en cada uno de los tipos de comunicación que se manejan, ahora con la clase se puede llamar a la misma desde cualquier parte y se reduce la cantidad de código del programa.

Posteriormente se crean dos procedimientos para pedir el diagnostico según el tipo de comunicación que maneje la sonda (TCP-IP o serial RS232).

Las sondas que manejan comunicación serial RS232 están configuradas directamente en el servicio, esto quiere decir que se conoce el identificador de cada una de ellas (parámetro necesario para preguntar por el diagnostico). Teniendo en cuenta esto solo es necesario crear un método que se llama cada cierto tiempo (definido por el usuario en la ventana de configuración del servicio). Dicho método cambia el mensaje que se envía a las sondas normalmente por el mensaje que solicita el diagnóstico, de esta manera se espera que las sondas respondan con la dicha información.

Las sondas que manejan comunicación TCP-IP no están configuradas directamente en el servicio por lo que se desconoce el identificador de estas. Como para enviar la solicitud de diagnóstico se debe conocer este parámetro, se crea inicialmente una rutina que identifique cada una de las sondas conectadas vía TCP-IP. A continuación, se crea un método que se ejecuta cada cierto tiempo (el mismo que para el caso de las sondas con comunicación serial RS232), dicho método realiza la solicitud de diagnóstico a cada una de las sondas y recupera la respuesta de cada una de ellas.

Finalmente se creó un método para procesar la información de diagnóstico y publicarla en la base de datos, dicho método verifica el checksum del mensaje que se recibió, organiza la información recibida y la publica en la base de datos si es una sonda sin diagnostico o sobrescribe la información si ya existe un diagnostico pasado.

11.2. Creación de la herramienta de visualización de datos en tiempo real

La herramienta de visualización de datos en tiempo real se va a adicionar a la ventana de configuración del servicio (figura 5), será una forma basada en formularios windows form, accesible solo cuando el servicio este detenido y permitirá al usuario realizar pruebas locales sin necesidad de ningún componente externo.

Port	Transmission Mode	Probe Serial Number
COM10	Push	0000
COM10	Pulling	11111

Figura 5. Ventana de configuración del servicio

Se establece que la herramienta debe estar en la capacidad de obtener datos de las sondas en los tres modos de comunicación que estas pueden poseer (Serial RS232 tipo pull, serial RS232 tipo push y TCP-IP tipo pull), teniendo en cuenta esto se define crear tres pestañas diferentes, una por cada modo de comunicación, dentro de las cuales se manejarán los diversos tipos de mensaje que soporten las sondas.

Teniendo en cuenta que el objetivo de la herramienta es mostrar la información recuperada de las sondas de una manera comprensible para el usuario, se decide crear inicialmente un método que interprete cualquiera de los mensajes recibidos de las sondas. El método identifica el mensaje recibido, verifica su integridad y organiza la información dándole un formato adecuado para mostrarla al usuario. Este método será reutilizado en todos los procesos en los que se obtengan datos de las sondas en la herramienta.

Posteriormente se procede a crear cada una de las pestañas empezando por la interfaz gráfica y continuando con el código que establece el funcionamiento de la misma.

11.2.1. Pestaña para la comunicación serial RS232 tipo pull

La primera pestaña que se crea es la pestaña para la comunicación serial RS232 tipo pull. Para esta, se define como requerimiento, tener la posibilidad de recuperar información de nivel y diagnóstico de una o varias sondas, así como poder consultar en todos los puertos COM por las sondas disponibles, aun sin conocer su identificador o el puerto al que están asociadas. Basado en los requerimientos se diseña la interfaz que se muestra en la figura 6.

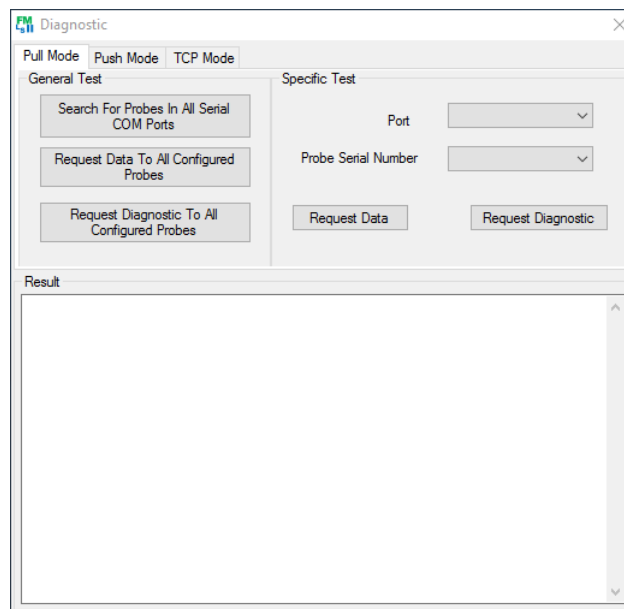


Figura 6. Interfaz gráfica pestaña para la comunicación serial RS232 tipo pull

Luego de tener el diseño de la interfaz, se adiciona el código para añadir la funcionalidad a los componentes de la pestaña. Primero, se llenan las listas desplegables, la lista "Port" se llena con los puertos COM asociados a las sondas tipo pull configuradas en el servicio y la lista "Probe Serial Number" se llena con los identificadores de las sondas configuradas según el puerto seleccionado.

Para finalizar el desarrollo de la pestaña, se agrega el código que define el proceso que se debe realizar cuando se presiona cada uno de los botones. A continuación se explica la funcionalidad de cada botón y se expone el caso de uso en el cual se basó el desarrollo:

11.2.1.1 Botón “Search For Probes In All Serial Com Ports”

Al presionar este botón, el programa debe preguntar en cada uno de los puertos COM que posea el equipo por sondas disponibles y en caso de obtener respuesta debe mostrarla en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 3.

Tabla 3. Caso de uso botón “Search For Probes In All Serial Com Ports”

Numeral	Entidad	Caso Uso
1	Usuario	Hace Clic en el botón “Search For Probes In All Serial COM Ports”
2	Sistema	Envía el comando que consulta por sondas disponibles (C + chr[13]) a cada uno de los puertos seriales que tenga disponibles el equipo.
3	Sistema	Espera respuesta en cada puerto.
4	Sistema	Si obtiene respuesta muestra las sondas disponibles en el puerto en cuestión en el cuadro “Result”, de lo contrario muestra el error presentado.
5	Fin	Fin.

11.2.1.2 Botón “Request Data To All Configured Probes”

Al presionar este botón, el programa debe preguntar a cada una de las sondas configuradas por datos de nivel y en caso de obtener respuesta debe mostrarla en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 4.

Tabla 4. Caso de uso botón “Request Data To All Configured Probes”

Numeral	Entidad	Caso Uso
1	Usuario	Hace Clic en el botón “Request Data To All Configured Probes”
2	Sistema	Envía el comando que consulta por información de nivel (MXXXXX+chr[13] Siendo XXXXX el id de la sonda) a cada una de las sondas configuradas en el servicio (ventana “Settings”).
3	Sistema	Espera respuesta de cada sonda.
4	Sistema	Si obtiene respuesta muestra la información de nivel que reporta cada sonda en el cuadro “Result”, de lo contrario muestra el error presentado.
5	Fin	Fin.

11.2.1.3 Botón “Request Diagnostic To All Configured Probes”

Al presionar este botón, el programa debe preguntar a cada una de las sondas configuradas por datos de diagnóstico y en caso de obtener respuesta debe mostrarla en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 5.

Tabla 5. Caso de uso botón “Request Diagnostic To All Configured Probes”

Numeral	Entidad	Caso Uso
1	Usuario	Hace Clic en el botón “Request Diagnostic To All Configured Probes”
2	Sistema	Envía el comando que consulta por información de diagnóstico (DXXXXX+chr[13] Siendo XXXXX el id de la sonda) a cada una de las sondas configuradas en el servicio (ventana “Settings”).
3	Sistema	Espera respuesta de cada sonda.
4	Sistema	Si obtiene respuesta muestra la información de diagnóstico que reporta cada sonda en el cuadro “Result”, de lo contrario muestra el error presentado.
5	Fin	Fin.

11.2.1.4 Botón “Request Data”

Para que este botón funcione, el usuario debe haber escogido antes un puerto y/o un número de serial de sonda en las listas desplegables. Al presionar este botón, el programa debe preguntar a cada una de las sondas configuradas en el puerto seleccionado o a la sonda en específico (si se seleccionó una) por datos de nivel y en caso de obtener respuesta debe mostrarla en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 6.

Tabla 6. Caso de uso botón “Request Data”

Numeral	Entidad	Caso Uso
1	Usuario	Selecciona un puerto y/o una sonda usando las listas desplegables “Port” y “Probe Serial Number” (se visualizan sondas configuradas en la ventana Settings).
2	Usuario	Hace Clic en el botón “Request Data”
3	Sistema	Envía el comando que consulta por información de nivel (MXXXXX+chr[13] Siendo XXXXX el id de la sonda) a cada una de las sondas configuradas en el puerto seleccionado o a la sonda seleccionada.
4	Sistema	Espera respuesta de la(s) sonda(s).
5	Sistema	Si obtiene respuesta muestra la información de nivel que reporta la(s) sonda(s) en el cuadro “Result”, de lo contrario muestra el error presentado.
6	Fin	Fin.

11.2.1.5 Botón “Request Diagnostic”

Para que este botón funcione, el usuario debe haber escogido antes un puerto y/o un número de serial de sonda en las listas desplegables. Al presionar este botón, el programa debe preguntar a cada una de las sondas configuradas en el puerto seleccionado o a la sonda en específico (si se seleccionó una) por datos de diagnóstico y en caso de obtener respuesta debe mostrarla en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 7.

Tabla 7. Caso de uso botón “Request Diagnostic”

Numeral	Entidad	Caso Uso
1	Usuario	Selecciona un puerto y/o una sonda usando las listas desplegables “Port” y “Probe Serial Number” (se visualizan sondas configuradas en la ventana Settings).
2	Usuario	Hace Clic en el botón “Request Diagnostic”
3	Sistema	Envía el comando que consulta por información de diagnóstico (DXXXXX+chr[13] Siendo XXXXX el id de la sonda) a cada una de las sondas configuradas en el puerto seleccionado o a la sonda seleccionada.
4	Sistema	Espera respuesta de la(s) sonda(s).
5	Sistema	Si obtiene respuesta muestra la información de diagnóstico que reporta la(s) sonda(s) en el cuadro “Result”, de lo contrario muestra el error presentado.
6	Fin	Fin.

11.2.2. Pestaña para la comunicación serial RS232 tipo push

La segunda pestaña que se crea es la pestaña para la comunicación serial RS232 tipo push. Las sondas que manejan este modo de comunicación son las más básicas por lo que se define solo un requerimiento, recuperar información de nivel desde cualquiera de los puertos COM asociados a sondas tipo push configuradas en el servicio. Basado en este requerimiento se diseña la interfaz que se muestra en la figura 7.

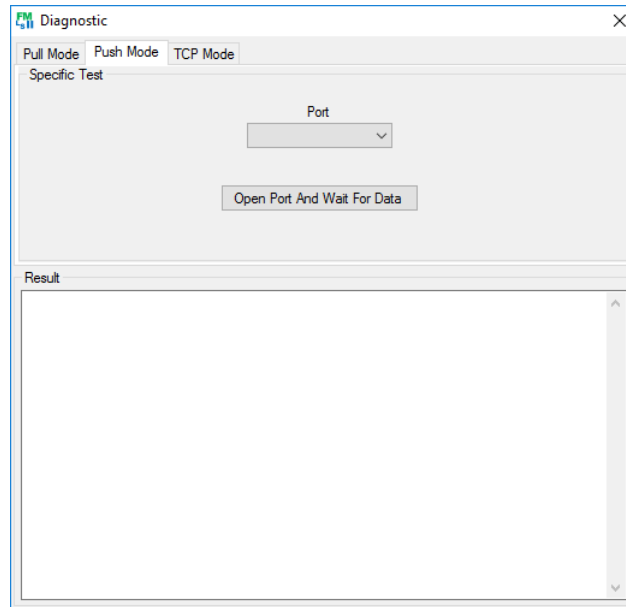


Figura 7. Interfaz gráfica pestaña para la comunicación serial RS232 tipo push

Luego de tener el diseño de la interfaz, se procede a adicionar el código para añadir la funcionalidad a los componentes de la pestaña. Primero, se llena la lista desplegable “Port” con los puertos COM asociados a las sondas tipo push configuradas en el servicio.

Posteriormente se añade el código que define el proceso que se debe realizar cuando se presiona el único botón que posee la pestaña. A continuación se explica la funcionalidad del mismo y se expone el caso de uso en el cual se basó el desarrollo:

11.2.2.1 Botón “Open Port And Wait For Data”

Para que este botón funcione, el usuario debe haber escogido antes un puerto en la lista desplegable. Al presionar este botón, el programa debe abrir el puerto anteriormente seleccionado durante sesenta segundos y esperar por datos de nivel de las sondas asociadas a este, luego de este tiempo debe mostrar la información que recibió en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 8.

Tabla 8. Caso de uso botón “Open Port And Wait For Data”

Numeral	Entidad	Caso Uso
1	Usuario	Selecciona un puerto usando la lista desplegable “Port” (se visualizan puertos configurados en la ventana Settings).
2	Usuario	Hace Clic en el botón “Open Port And Wait For Data”
3	Sistema	Abre el puerto seleccionado durante un minuto y espera por datos de nivel de sondas tipo push.
4	Sistema	Cierra el puerto y si obtiene respuesta muestra la información de nivel que reportan la(s) sonda(s) tipo push en el cuadro “Result”, de lo contrario muestra el error presentado.
5	Fin	Fin.

11.2.3. Pestaña para la comunicación TCP-IP tipo pull

La tercera y última pestaña que se crea es la pestaña para la comunicación TCP-IP tipo pull. El requerimiento general para esta es recuperar información de nivel y diagnóstico de las sondas que se están conectando al servidor vía TCP-IP. Basado en este requerimiento se diseña la interfaz que se muestra en la figura 8.

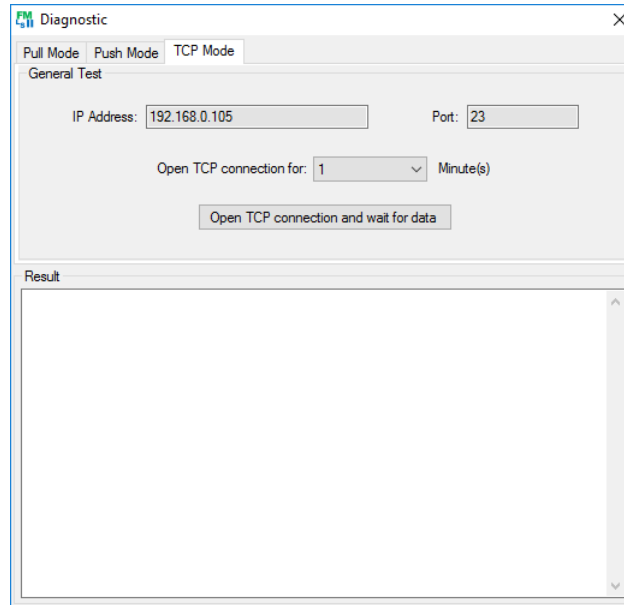


Figura 8. Interfaz gráfica pestaña para la comunicación TCP-IP tipo pull

Luego de tener el diseño de la interfaz, se procede a adicionar el código para añadir la funcionalidad a los componentes de la pestaña. Primero, se llenan los campos "IP Address" y "Port" con la dirección IP y el puerto (definidos en la configuración del servicio) en el cual el servicio va a recibir los clientes TCP con sondas asociadas. De igual manera se llena la lista desplegable con tres valores posibles para que el usuario escoja, uno, dos y tres minutos, durante los cuales la conexión va a estar activa para recibir clientes y recuperar datos de nivel.

Finalmente se adiciona el código que define el proceso que se debe realizar cuando se presiona el único botón que posee la pestaña. A continuación se explica la funcionalidad del mismo y se expone el caso de uso en el cual se basó el desarrollo:

11.2.3.1 Botón “Open TCP Connection And Wait For Data”

Al presionar este botón, el programa abre una conexión TCP en la dirección IP y el puerto que se observa en los campos “IP Address” y “Port” durante el tiempo que el usuario determine en la lista desplegable. Si se recibe una conexión de un cliente, el programa realiza la consulta de datos de nivel y diagnóstico y si recibe datos muestra la información en la ventana de la herramienta. El caso de uso en el que se basó el desarrollo del código para este botón se expone en la tabla 9.

Tabla 9. Caso de uso botón “Open TCP Connection And Wait For Data”

Numeral	Entidad	Caso Uso
1	Sistema	Carga la dirección IP y el puerto configurados en el servicio para el modo TCP (ventana “Settings”) y los coloca en las casillas “IP Address” y “Port”.
2	Usuario	Selecciona el tiempo que desea mantener abierta la conexión TCP usando la lista desplegable “Open TCP connection for”.
3	Usuario	Hace Clic en el botón “Open TCP connection and wait for data”.
4	Sistema	Abre una conexión TCP en la dirección IP y puerto configurados y espera a que se conecten dispositivos remotos como clientes.
5	Sistema	Cuando recibe un nuevo cliente envía el comando que consulta por sondas disponibles (C + chr[13]) y espera por respuesta.
6	Sistema	Cuando pasa el tiempo seleccionado, cierra la conexión TCP y si obtiene respuesta muestra la información de nivel que reportan la(s) sonda(s) en el cuadro “Result”, de lo contrario muestra el error presentado.
7	Fin	Fin.

11.3. Análisis y mejora en el rendimiento del servicio

Luego de finalizadas las tareas principales que componían el proyecto, se tomó el reto de realizar un análisis de rendimiento al servicio, con el fin de reducir la carga de procesamiento que se estaba generando en el servidor donde se aloja el mismo, ya que esta era exagerada teniendo en cuenta que el servicio no realiza tareas complejas que lleven a esta consecuencia.

Para realizar esta tarea se usaron dos herramientas, la primera fue uno de los documentos revisados en el entrenamiento que se realizó al comenzar la pasantía[12], que habla sobre buenas practicas a la hora de escribir código y la segunda fue el aplicativo de análisis de rendimiento y diagnostico que ofrece Visual Studio 2013.

Inicialmente se realizó una revisión general del código buscando malas prácticas de programación, en esta revisión se encontraron dos situaciones que presuntamente podían llevar a elevar la carga de procesamiento que tiene el servicio sobre la máquina. En la tabla 10 se enuncian las situaciones encontradas, la causa por la cual estas podrían elevar el procesamiento y la medida que se tomó para corregir dicha mala práctica.

Tabla 10. Resultados del análisis al código del servicio

Mala práctica encontrada	Causa por la que podría elevar el procesamiento	Acción tomada
Múltiples instancias a web service en los diferentes métodos y clases del servicio	Cada que se realiza una instancia a web service el servicio debe hacer una petición para recuperar todos los métodos de este y almacenarlos en un objeto. Al ser esta una tarea dependiente de una petición en la red, toma tiempo y procesamiento por lo que no es conveniente hacerla varias veces en el mismo programa.	Se instancia al web service una sola vez cuando el servicio inicia y luego se reutiliza el objeto en el que se almacenaron los métodos de este.
Múltiples consultas a base de datos en los métodos de web service utilizados	Dentro de algunos de los métodos de web service que se usaban en el servicio, se realizaban varias consultas a base de datos que traían información común, esto es ineficiente pues se debe acceder a la base de datos varias veces para recuperar información que se puede obtener con una sola consulta.	Se modificaron los métodos de web service en los que se encontró esta mala práctica y se redujeron todos a una consulta por método haciendo uso de sentencias como JOIN.

Luego de la revisión de código, se realizó una prueba de rendimiento al servicio haciendo uso de la herramienta de análisis de rendimiento y diagnostico que ofrece Visual Studio 2013. Para poder usar esta herramienta, se revisó la documentación que provee Microsoft [13] con el fin de entender su funcionamiento y aprender a analizar los resultados que esta arroja.

Para la prueba se usó un método de análisis que incluye la herramienta llamado muestreo de CPU. Este método captura durante intervalos de tiempo determinados, información de las funciones que se usan en el servicio e informa al usuario, cuáles de ellas realizan la mayor parte del trabajo sobre la CPU.

Los resultados del análisis al servicio se muestran en la figura 9.



Figura 9. Resultados de análisis al código del servicio

Debido a la experiencia con otros servicios, se sabe que al iniciarlo se produce una subida de procesamiento, pues se debe instanciar cada una de las clases que lo componen y se debe instanciar el web service, por ello se ignora la primera parte del informe que muestra alto procesamiento.

Se procede a revisar las otras subidas que se dan periódicamente (cada que el servicio pregunta por información a las sondas). El análisis específico de una de estas zonas se muestra en la figura 10.

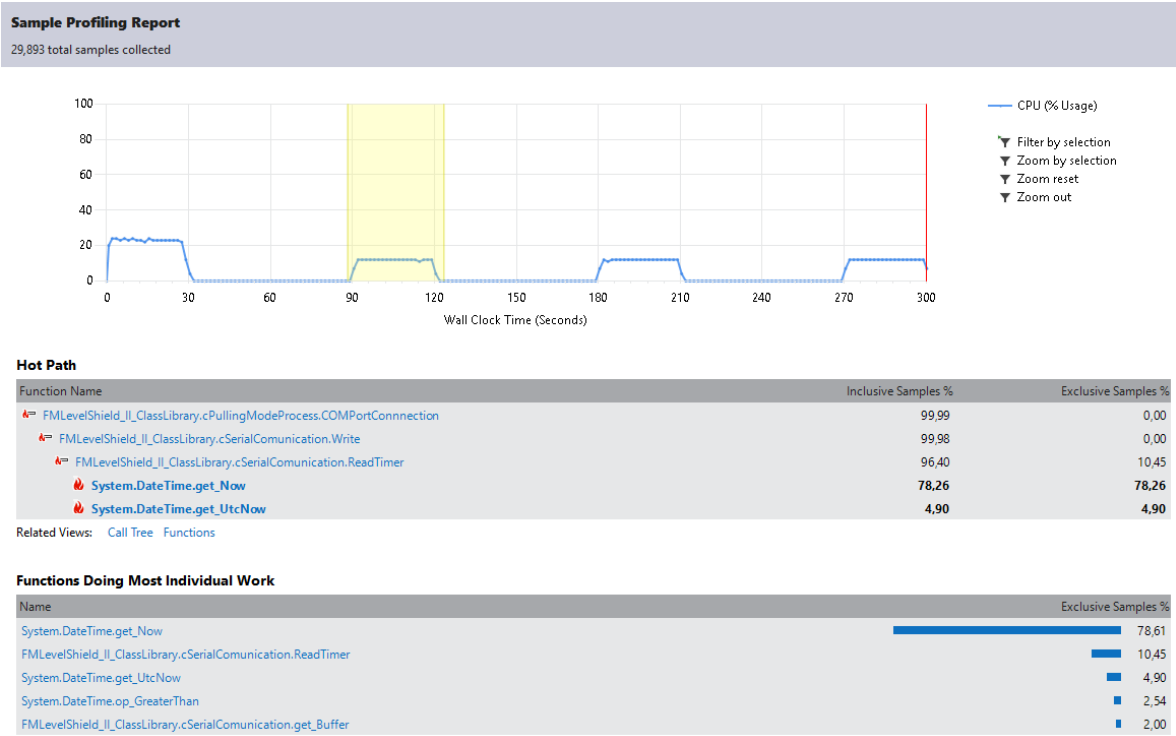


Figura 10. Resultados de análisis específico

En la zona de alto procesamiento, se identifica que todo el trabajo del servicio recae sobre una sola función, utilizada dentro de la clase que maneja la comunicación serial en el servicio. Se revisa el código y se descubre un mal uso de dicha función, por lo que se opta por realizar las modificaciones pertinentes para eliminar el problema. Se modifica el código que tenía dicha clase en el servicio y se repite el análisis manteniendo la misma configuración que se había usado en la prueba anterior (figura 11).

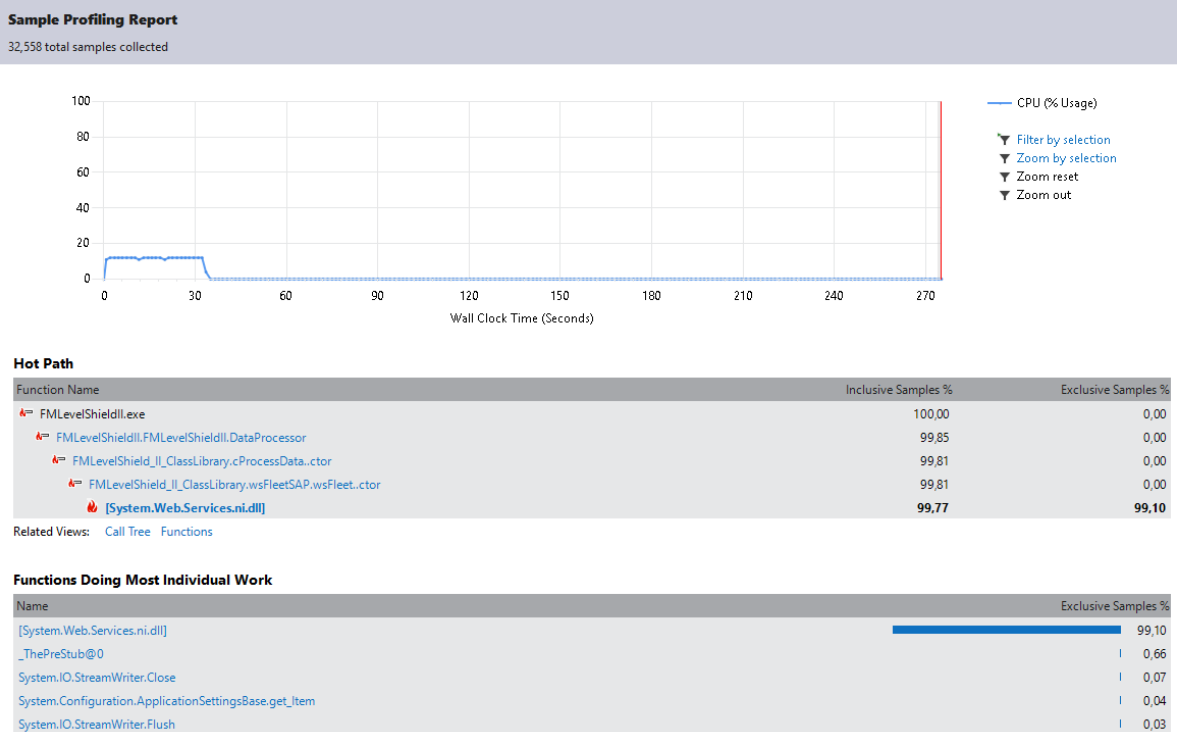


Figura 11. Resultados de análisis luego de modificaciones

Se observa que el procesamiento del servicio se redujo a cero prácticamente luego de que inicializa, con esto se confirma que el problema se solucionó y ahora el servicio tiene un rendimiento óptimo.

Se genera el instalador y a liberar la versión del servicio junto con los respectivos manuales para que así, el personal de soporte haga la actualización en el servidor de la compañía.

12. RESULTADOS

Se ha logrado integrar la solicitud de diagnóstico al servicio, ahora es posible conocer los detalles de diagnóstico de una sonda.

Luego de la realización de este cambio, el equipo de desarrollo web integró esta información a la plataforma de la compañía (FleetSAP) para que fuera accesible para los clientes. En la figura 12 se observa el diseño actual de la información que se decidió mostrar basados en la relevancia de la misma. Para tener acceso a todos los valores se debe consultar directamente en la base de datos.

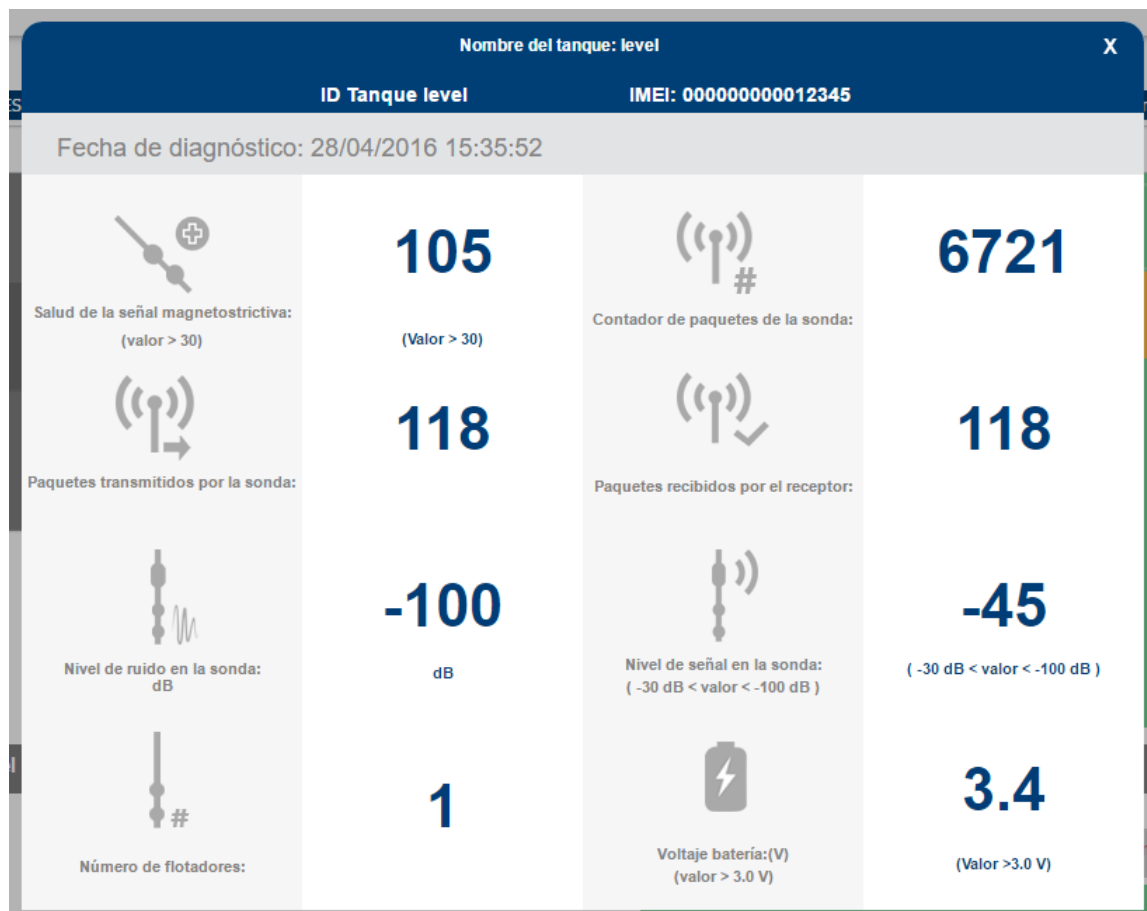


Figura 12. Valores de diagnóstico mostrados en FleetSAP [13]

También se entregó la herramienta de visualización de datos en tiempo real integrada a la consola de configuración del servicio, otorgando la posibilidad de visualizar los datos de las sondas sin necesitar herramientas ajenas al servicio (web service, base de datos). En la figura 13 se muestra la manera en que se visualizan datos de nivel de las sondas y en la figura 14 se ilustra la manera en que se visualizan los datos de diagnóstico en la herramienta.

```

----- Port COM3 - Probe Serial Number 20031 -----
Bytes received: 32-30-30-33-31-4E-30-3D-2B-31-39-34-3D-30-31-37-36-31-2E-36-35-3D-30-30-30-31-
35-2E-37-34-3D-30-39-34-0A-0D
Message received: 20031N0=+194=01761.65=00015.74=094
DATA RECEIVED
    Probe Address: 20031
    Probe Status: OK (0)
    Temperature: 19,4 °C
    Product Level: 1761,65 mm
    Water Level: 15,74 mm

```

Figura 13. Datos de nivel visualizados en la herramienta

```

----- Port COM3 - Probe Serial Number 04276 -----
Bytes received: 30-34-32-37-36-44-30-31-39-3D-30-30-30-30-30-30-37-3D-30-30-30-35-39-31-30-
30-3D-30-30-30-30-30-30-37-3D-30-30-32-3D-30-30-35-3D-32-30-30-3D-30-30-31-3D-30-37-32-
3D-31-31-37-3D-30-37-36-3D-30-38-37-3D-31-31-37-3D-30-37-31-3D-32-30-34-0A-0D
Message received:
04276D019=00000007=00059100=00000007=002=005=200=001=072=117=076=087=117=071=204
DIAGNOSE RECEIVED
    Probe Address: 04276
    Window Signal: 19
    Packets Transmitted: 7
    Counter Of Packets: 59100
    Packets Received: 7
    Number Of Floats: 2
    Transmission Channel: 5
    Signal Power: 200
    Operating Mode: 1
    Level Local Signal: -72 dB
    Level Local Noise: -117 dB
    Receiver Tension A: 3,6 V
    Level Probe Signal: -87 dB
    Level Probe Noise: -117 dB
    Receiver Tension B: 3,4 V

```

Figura 14. Datos de diagnóstico visualizados en la herramienta

Finalmente se comprobó que se había mejorado el rendimiento del servicio con los cambios realizados. Se realizó una verificación local con una sola sonda pues por políticas de la compañía no está permitido obtener datos de los servidores oficiales. En la figura 15 se observa el monitor de recursos de Windows con el servicio antes de los cambios.

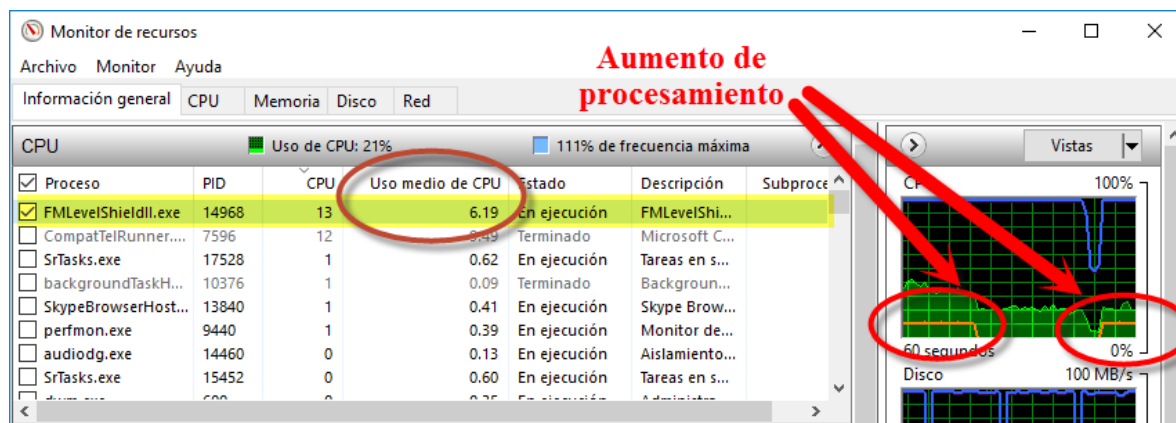


Figura 15. Datos de diagnóstico visualizados en la herramienta

Se observa que el uso medio de CPU es del 6% y que cada que el servicio pregunta por información a la sonda se da una subida en el procesamiento (Línea naranja de la gráfica). La figura 15 ilustra el monitor de recursos de Windows con el servicio luego de los cambios.

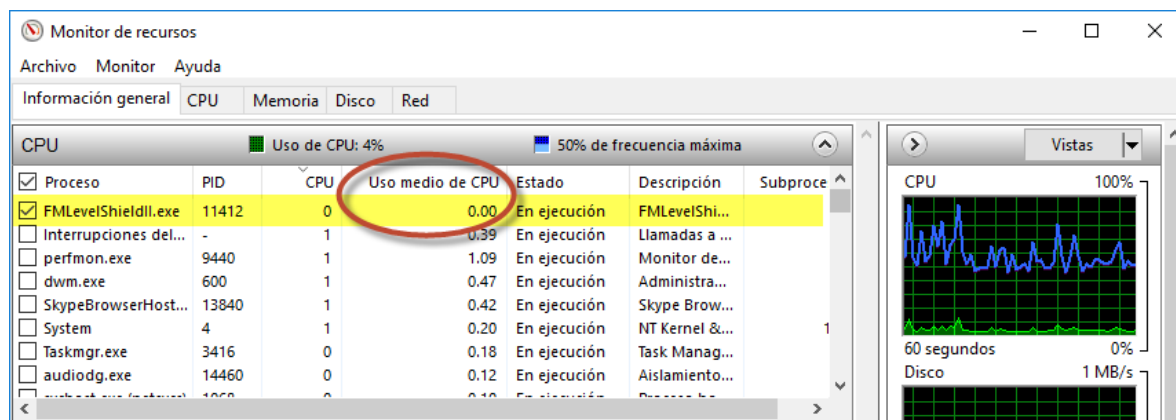


Figura 16. Datos de diagnóstico visualizados en la herramienta

Se observa que ahora el uso medio de CPU es del 0% y en la gráfica no se aprecian subidas de procesamiento cuando el servicio pregunta a las sondas.

Cabe aclarar que el resultado final en el servidor, fue pasar de un uso medio de CPU del 60% a un uso medio de CPU del 0.2%, esto teniendo en cuenta que allí se manejan el 90% de las sondas asociadas a la compañía (las demás son instalaciones *standalone*).

Luego de un tiempo de la liberación de los cambios, se verificó el estado de los casos relacionados con el servicio reportados por los clientes al departamento de soporte. Las cifras presentadas a continuación fueron entregadas de manera verbal por este departamento, el registro formal de los casos no puede ser divulgado fuera de esta área.

El mes anterior a la liberación de los cambios se reportaron doce (12) casos relacionados con problemas de sondas Level Shield™, nueve (9) de estos estuvieron asociados a fallas en la transmisión por bajos niveles de batería. De haber estado disponible el desarrollo realizado en este proyecto se habrían evitado dichos reportes, pues dentro de la información de diagnóstico se tiene un campo relacionado al nivel de batería, por lo que se habría podido realizar el cambio de la misma cuando se observara un voltaje bajo en esta.

El mes posterior a la liberación de los cambios solo se obtuvieron cuatro (4) casos relacionados con problemas de sondas Level Shield™, adicionalmente, durante este mes se realizaron tres cambios de batería gracias a la información obtenida en el diagnóstico, estos cambios hubiesen podido desencadenar en tres casos reportados de no haber estado el desarrollo.

Aparte de la reducción de casos reportados, se redujo el tiempo de respuesta a la hora de la atención de los mismos. Antes de la liberación del desarrollo, cuando se presentaba un reporte de falla, era necesario contactar al encargado de base de datos para verificar si había datos de las sondas en la misma, con ello se definía si el problema era de la aplicación web (FleetSAP) o del hardware, este proceso podía tardar entre uno y dos días. Ahora gracias a la herramienta de visualización de datos en tiempo real, es posible establecer si el hardware está funcionando correctamente en menos de dos minutos.

Finalmente, como con la mejora de rendimiento del servicio se logró reducir el uso de recursos del servidor, el departamento de tecnología decidió instalar en este equipo un web service adicional que ha permitido realizar contingencias y reducir la carga del servidor que tenía el web service original.

13. CONCLUSIONES

- Al finalizar todo el trabajo, se ha entregado una nueva versión del servicio FMLevelShieldII. Dicha versión contiene todos los requerimientos definidos inicialmente como objetivos para este trabajo, a saber, la adición de solicitud de diagnóstico para sondas Level Shield™, una herramienta para visualización de datos de las sondas en tiempo real y una mejora en el rendimiento y uso de recursos de la máquina.
- Con el desarrollo entregado se logran aprovechar todas las capacidades de la sonda, de este modo se saca el máximo beneficio de la inversión realizada y se pueden ofrecer al cliente, funcionalidades adicionales que dan un valor agregado al producto o servicio en cuestión.
- Gracias a las nuevas funcionalidades que se incluyen en el servicio, se logró facilitar el trabajo del personal de la compañía que tiene contacto directo con los clientes. Con ello se contribuye al mejoramiento de los procesos en otras áreas, se mantiene la fidelidad y confianza del cliente pues recibe un soporte más rápido y se aumenta la productividad interna.
- Este desarrollo confirma la importancia de usar un estándar al escribir código que permita entender fácilmente el programa, de esta manera se asegura que este podrá ser versátil, que se le podrá dar mantenimiento fácilmente y que se podrá hacer crecer metódicamente sin necesidad de modificar, lo ya realizado.
- El trabajo realizado para mejorar el rendimiento del programa, hace comprender el especial cuidado que se debe poner sobre este aspecto en una aplicación. Se debe controlar y monitorear el uso de recursos de la misma y se debe actuar ante cualquier falla por pequeña que sea, pues al no pensar en escala, se puede dar un uso inadecuado de los equipos cuando esta empieza a crecer.
- La práctica profesional realizada en la compañía SCI, permitió al estudiante poner en práctica muchos de los conocimientos adquiridos en la carrera, así como potencializar diversas habilidades que en conjunto permitieron llevar a buen término el presente proyecto. De igual forma se mejoraron las habilidades comunicativas, pues se tuvo la posibilidad de tratar con clientes y proveedores durante el desarrollo del trabajo.

14. REFERENCIAS Y BIBLIOGRAFIA

- [1] Creación propia, basado en: Manual de instalación y configuración servicio FMLevelShieldII, 2016.
- [2] COMPUTRABAJO, Información empresa SCI Sur América SAS [en línea]. <www.computrabajo.com.co/empresas/ofertas-de-trabajo-de-sci-sur-america-sas-6E08B506D43AD8A6> [Revisado en 17 de mayo de 2016]
- [3] SCIUFM, Sitio web de una de las filiales de la compañía [en línea]. <<http://www.sciufm.com/>> [Revisado en 17 de mayo de 2016]
- [4] DEITEL, Harvey - DEITEL, Paul. C# for programmers. Crawfordsville: Pearson Education, 2005. P 3-4
- [5] SELLS, Chris. Windows Forms Programming in C#. Boston: Pearson Education, 2003. P xxxvii
- [6] DEITEL, Harvey - DEITEL, Paul. C# for programmers. Crawfordsville: Pearson Education, 2005. P 964-965
- [7] AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey D. Compilers, Principles, Techniques. Boston: Addison wesley, 1986.
- [8] SCIWW, Level Shield™ Overview [en línea]. <<http://sciww.com/level-shield/>> [Revisado en 17 de mayo de 2016]
- [9] Creación propia, basado en: Manual de instalación y configuración servicio FMLevelShieldII, 2016.
- [10] Departamento de tecnología SCI, Manual de instalación y configuración servicio FMLevelShieldII, 2016.
- [11] Creación propia.
- [12] Departamento de tecnología SCI, ST01StandardCSharp, 2016.

[13] MICROSOFT DEVELOPER NETWORK, Analyzing application performance by using profiling tools [en línea]. <[https://msdn.microsoft.com/en-us/library/z9z62c29\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/z9z62c29(v=vs.120).aspx)> [Revisado en 6 de enero de 2017]

[14] APLICATIVO FLEETSAP, Ventana de diagnóstico LevelShield [en línea]. <<https://web.fleetsap.com/FleetSAP>> (Compañía de pruebas) [Revisado en 6 enero de 2017]

15. ANEXOS

Anexo A. Diagrama de Gantt (Gantt project)

