

**DESARROLLO EN C# DE UN MÓDULO DE MANTENIMIENTO VEHICULAR
PARA LA APLICACIÓN FLEETSAP BASADO EN LA CREACIÓN DE UN
HISTORIAL DE MANTENIMIENTO**

SANDRA GINETH CORONADO ALDANA

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ DC
2014**

**DESARROLLO EN C# DE UN MÓDULO DE MANTENIMIENTO VEHICULAR
PARA LA APLICACIÓN FLEETSAP BASADO EN LA CREACIÓN DE UN
HISTORIAL DE MANTENIMIENTO**

**SANDRA GINETH CORONADO ALDANA
2092964**

Monografía Para Optar Al Título De Ingeniero Electrónico

**Tutor:
JAIME VITOLA OYAGA ING. MSc**

Docente de la Facultad de Ingeniería Electrónica

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ DC
2014**

RECTOR GENERAL

Padre Carlos Mario Álzate Montes, O.P

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

Padre Diego Orlando Serna Salazar, O.P.

VICERRECTOR ACADÉMICO GENERAL

Padre Eduardo González Gil, O.P.

SECRETARIO GENERAL

Doctor Héctor Fabio Jaramillo Santamaría

DECANO DIVISIÓN DE INGENIERÍAS

Padre Pedro José Díaz Camacho, O.P.

SECRETARIA DE DIVISIÓN

Yendy Elizabeth Caballero Barrera

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

Ingeniera Adriana Páez Pino.

Nota de Aceptación

Tutor Monografía

Jurado

Jurado

Bogotá, Octubre del 2014

ADVERTENCIA

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados por los autores en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

**Agradezco inmensamente a Dios,
mis padres, hermana, maestros y
amigos por todo el apoyo prestado
a lo largo de mi formación
profesional.**

AGRADECIMIENTOS

En primer momento agradezco a Dios, quien puso en mi camino a mis padres, docentes, amigos y compañeros que me brindaron siempre apoyo, consejos y contribuyeron a formarme integralmente.

En segundo lugar, a la universidad Santo Tomás de Aquino por formarme durante estos años con variados conocimientos humanistas, matemáticos, físicos y electrónicos.

En tercer lugar a mis padres, que me dieron la posibilidad de estudiar en una excelente universidad, por apoyarme económicamente y emocionalmente durante toda mi vida.

En cuarto lugar, a mis docentes que me acompañaron en mi formación, me brindaron soporte en cuanto a conocimientos, algunos de forma emocional. Muchas gracias por compartir conmigo sus conocimientos y experiencias que han sido de gran ayuda en mi vida profesional.

En quinto lugar, gracias a mis compañeros y amigos que cursaron conmigo estos cinco años, realmente compartí muy buenos momentos con ustedes además de sobresalir en los trabajos y obtener buenos resultados.

Y por último, gracias a **SCI Sur América S.A.S.** quien depositó en mí un gran voto de confianza al permitirme desarrollar mis habilidades como ingeniera electrónica. En especial al Ing. Diego Rodríguez quien me apoyó en todo momento y compartió sus conocimientos y experiencias a lo largo de la pasantía.

CONTENIDO

LISTADO DE FIGURAS	10
LISTADO DE TABLAS.....	11
GLOSARIO.....	11
INTRODUCCIÓN	15
I. DESCRIPCIÓN DEL PROYECTO.....	166
1.1 PLANTEAMIENTO DEL PROBLEMA.....	16
1.1.1 FORMULACIÓN	17
1.1.2 ANTECEDENTES	17
1.2 JUSTIFICACIÓN	19
1.3 OBJETIVOS	20
1.3.1 OBJETIVO GENERAL.....	20
1.3.2 OBJETIVOS ESPECÍFICOS.....	20
1.4 METODOLOGÍA.....	21
1.4.1 ETAPA DE ENTENDIMIENTO DEL PROBLEMA	22
1.4.2. ETAPA DE INVESTIGACIÓN Y DOCUMENTACIÓN.....	22
1.4.3. ETAPA DE ESPECIFICACIONES	23
1.4.4. ETAPA DE DESARROLLO DEL PROBLEMA.....	23
1.4.5. ETAPA DE PRUEBAS Y CORRECCIONES	24
1.4.6. ETAPA DE ENTREGA.....	24
1.5 DELIMITACIÓN DEL PROYECTO.....	25
II. LA EMPRESA	26
2.1 SCI SUR AMERICA SAS	26
2.1.1 RESEÑA	26
2.1.2 ACTIVIDADES	26

2.2 PASANTÍA EN SCI SUR AMERICA SAS	27
2.2.1 DESARROLLO DE LA PASANTÍA	27
2.2.2 FUNCIONES	27
III. MARCO TEORICO	28
3.1 VISUAL STUDIO EXPRESS	28
3.2 C#	29
3.2.1 CLASES	29
3.2.2 CONSTRUCTOR	30
3.3 SQL SERVER MANAGEMENT STUDIO	30
3.3.1 TABLAS	31
3.3.1 CONSULTAS	32
3.4 SERVICIOS WEB	33
3.5 ESTANDAR PARA ESPECIFICACIONES.....	34
IV. EL PROYECTO.....	36
4.1 DESARROLLO DEL PROYECTO.....	36
4.2 FUNDAMENTO HUMANISTA	66
V. CONCLUSIONES.....	69
VI. VALOR AGREGADO Y TRABAJOS FUTUROS.....	72
REFERENCIAS BIBLIOGRÁFICAS	73
ANEXOS.....	74

LISTADO DE FIGURAS

Figura 1: Delimitación del proyecto.	24
Figura 2: Tablas MaintenanceTypes y MaintenanceHistory.	48
Figura 3: Tablas PreventiveMaintenance y PreventiveMaintenanceType.....	49
Figura 4: Servicios web de consulta.	49
Figura 5: Servicios web de adición de registros.	50
Figura 6: Servicios web de modificación.....	50
Figura 7: Servicios web de eliminación.....	50
Figura 8: Forma de mantenimiento de flota.....	51
Figura 9: Forma nueva denominada Módulo de mantenimiento de flota.	54
Figura 10: Lista de historial de mantenimiento.....	55
Figura 11: Mantenimiento preventivo.....	56
Figura 12: Mantenimiento preventivo.....	57
Figura 13. Diagrama de Grantt 1.....	72

LISTADO DE TABLAS

Tabla 1: Tabla PreventiveMaintenanceType. 41

Tabla 2: Tabla Preventive Maintenance..... 42

Tabla 3: Tabla MaintenanceTypes. 43

Tabla 4: Tabla MaintenanceHistory..... 44

GLOSARIO

ADD:	Función de SQL que admite añadir tablas y columnas en las tablas de la base de datos.
CheckBox:	Es un control de confirmación, una caja en la que el usuario da clic y se cambia el estado de verdadero a falso y viceversa.
Clases:	Es la plantilla general de los objetos, es donde se definen las propiedades y comportamientos que va a tener el objeto en el programa.
ComboBox:	Control de programación que permite desplegar una lista de opciones para que el usuario escoja alguna opción.
Constructores:	Es una rutina inferior que se encarga de inicializar los objetos de las clases.
DELETE:	Instrucción de SQL que permite eliminar registros, campos y tablas de la base de datos.
DevExpress:	Programa especializado en mejoras gráficas, este programa ofrece muchas herramientas para mejorar visualmente la aplicación a desarrollar.
Flota:	Es un conjunto de vehículos.
FOR:	Es un bucle de programación que funciona en base a contadores y comparaciones.
FROM:	Instrucción de SQL que sirve para llamar la tabla de la cual se van a realizar las consultas.
IF:	Instrucción de Programación que permite condicional cierta parte de código y ofrece la oportunidad de ejecutar o no acciones dentro de su comparación.

INSERT:	Instrucción de SQL que permite insertar datos en alguna tabla.
Label:	Etiqueta que se utiliza en programación para escribir títulos de identificación o para entregar cadenas de texto con resultados.
Main:	Estructura principal del programa donde debe estar contemplado todas las funciones y las variables.
Objetos:	Un objeto es una instancia de una clase, en él se encuentran propiedades y métodos, se relacionan con objetos reales de nuestro entorno.
SELECT:	Instrucción de SQL que permite recopilar información de las tablas que se encuentran en la base de datos. En esta instrucción se puede realizar cálculos y operaciones sobre los campos que se deseen.
SQL:	Lenguaje de consulta estructurado, es una aplicación que permite realizar consultas a los datos que se encuentran en la base de datos, además permite modificarlas con instrucciones sencillas y operaciones básicas.
Text Box:	Herramienta de programación que le permite al usuario escribir el valor de una variable para que sea almacenada.
UPDATE:	Instrucción de SQL que permite actualizar datos conforme al tiempo que transcurre.
Visual Basic:	Es un programa que se encarga de otorgar mejores herramientas de programación, donde la configuración es más gráfica y simplificado, además de que permite tener un código estructurado y ordenado.
WHERE:	Instrucción de SQL que permite condicionar la consulta para solo extraer lo que se requiere.

WHILE:

Instrucción de programación que permite realizar un proceso repetitivamente mientras la condición que se otorgó no se cumpla.

INTRODUCCIÓN

Éste documento, realiza la presentación formal del proyecto que se llevó a cabo durante la pasantía empresarial. El cual consistió en desarrollar un módulo de mantenimiento vehicular en la aplicación FleetSAP de la compañía SCI Sur América, que permitiera al usuario programar de una manera automática, teniendo en cuenta variadas características mecánicas y eléctricas de los diferentes vehículos.

De igual forma, se desarrolló un registro y control en donde el usuario tendrá la posibilidad de administrar la cantidad de reparaciones y controles, organizándolos por fechas en cada una de las flotas.

En este orden de ideas, se inicia el desarrollo del proyecto con la formulación del mismo, teniendo en cuenta los antecedentes y por supuesto la justificación de éste. Luego, se presentan los objetivos para trazar los requerimientos y determinar la factibilidad.

Basados en la formulación, se plantea la teoría necesaria para desarrollarlo y se diseña la respectiva metodología. Enseguida, se plantea la solución más adecuada teniendo en cuenta las herramientas de desarrollo suministradas por la compañía como lo son las bases de datos, servicios web y formularios de Windows.

I. DESCRIPCIÓN DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad, las compañías requieren de una gestión y administración de flotas competente, que presente información minuciosa en cuanto a inspección, lubricación, reparación y monitoreo de vehículos. En consecuencia, es de vital importancia brindar dicha información de forma ordenada y temporizada a los usuarios, permitiendo programar a sus unidades mantenimientos que sean convenientes dependiendo de las especificaciones técnicas del fabricante, historial de fallas, cantidad de horas de trabajo y uso; de esta forma, aumenta la vida útil de la unidad y se prevén deterioros en sus diferentes sistemas.

La plataforma FleetSAP de la compañía SCI Sur América satisface las necesidades de gestión de sus clientes. Sin embargo, el mercado cada día es más competitivo y necesita de continuas innovaciones que impacten a los usuarios y mantenga a SCI Sur América como pionera en él. Buscando ese liderazgo en el mercado, se pretende modificar el actual módulo de mantenimiento que permite tres tipos de servicios (Odómetro, horas motor y consumo de combustible acumulado) por uno que sea más amplio en cuanto a cantidad de mantenimientos, formas de ejecución, especificaciones de fabricantes y otro sin número de datos llevando a su vez un registro fechado de éstos. Esto se propone debido a la dificultad de establecer un molde para todas, haciendo más sencillo el desarrollo de una sección totalmente configurable por el cliente para cumplir con sus expectativas.

1.1.1 FORMULACIÓN

¿Cómo llevar un control y registro de los mantenimientos requeridos en un vehículo de manera flexible que se ajusten a las necesidades de diferentes tipos de vehículos y de clientes?

El reunir la mayor cantidad de información posible como son datos propios de la mecánica del vehículo (odómetro, horas motor y consumo de combustible), por otro lado las especificaciones propuestas por el fabricante, las políticas de mantenimiento del cliente y el historial de fallas del vehículo ofrecen elementos de juicio pertinentes que permiten tomar decisiones a tiempo y programar servicios que aumenten la seguridad y la vida útil del vehículo.

1.1.2 ANTECEDENTES

Los seguimientos mecánicos de un vehículo que existen en el mercado, normalmente se realizan anexando documentalmente el resultado de la evaluación realizada por un experto en el momento de acudir a un centro especializado, que revise todos los sistemas del vehículo. Sin embargo, por la cantidad de dinero y de tiempo, sólo se pueden realizar cada año que es lo que exige la ley; en el caso doméstico, podría ser suficiente pero para vehículos que tienen un uso más recurrente sería importante ofrecerles un proceso más elaborado que sistematice los datos y permita tener disponible la información cuando ésta se requiera, ya sea por una reparación o por procedimientos de mantenimiento preventivo.

La empresa SCI Sur América ofrece un sistema de seguimiento vehicular denominado FleetSAP que ofrece módulos de gestión de mantenimiento poco flexibles al usuario, pues cuenta con sólo tres tipos de servicio preventivo (Odometro, horas motor y consumo de combustible acumulado). Sin embargo, el

presente proyecto busca dar flexibilidad al cliente, permitiendo la creación de nuevos tipos de mantenimientos de acuerdo a las necesidades del cliente.

1.2 JUSTIFICACIÓN

SCI Sur América provee una aplicación de gestión de flotas llamada FleetSAP a compañías que requieran administrar sus diferentes unidades de transporte. El software que desarrolla, es bastante competente e imponente frente a compañías como Novatrans o CIF-Trans (que se dedican al mismo servicio). Sin embargo, es importante ofrecer un plus frente a las demás competencias, con el fin de establecer fidelidad en los usuarios e interés en los que aún no lo son.

Frente a estas compañías, SCI desea mejorar para cumplir con las necesidades de los usuarios en su plataforma en cuanto al módulo de mantenimientos, ya que la exigencia de los clientes referente al servicio que se presta es cada vez más grande. En consecuencia, se ha proyectado un módulo totalmente configurable por el usuario, ya que establecer uno que satisfaga las necesidades de todos los usuarios es muy denso. Así, éste proyecto permite al consumidor una mejor administración de flotas llenando todas sus expectativas y compañías competencia como Novatrans y CIF-Trans quedan un paso atrás frente a SCI.

En consecuencia, se puede identificar como ventaja que el módulo de mantenimiento vehicular mantendría las flotas en excelente estado, prevendría accidentes, alargaría su vida útil e informaría al gerente de flota sobre el estado de las mismas.

Gracias a que SCI Sur América permite realizar pasantías empresariales a estudiantes de ingeniería electrónica es posible desarrollar proyectos como éste, en los que el educando tiene la posibilidad de manifestar sus habilidades y conocimientos adquiridos a lo largo de su carrera profesional, aunque es importante reconocer la importancia del continuo estudio por parte de ellos en cuanto a conceptos de plataformas de programación como C# y SQL.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Construir un módulo de software vehicular adicional al producto FleetSAP, que incorpore más información a éste, permitiendo un manejo más flexible y singularizado tanto de los mantenimientos correctivos como de los preventivos.

1.3.2 OBJETIVOS ESPECÍFICOS

- Construir una aplicación que permita administrar los tipos de mantenimiento ya sea agregando, guardando o eliminándolos dependiendo de las necesidades del usuario.
- Crear una tabla en una base de datos, que permita tener un registro de los tipos de mantenimiento vehiculares adicionados.
- Establecer un servicio web que permita obtener, adicionar modificar o eliminar los tipos de mantenimiento del usuario.
- Diseñar un módulo de mantenimiento vehicular que permita visualizar al usuario los servicios programados de manera personalizada.
- Implementar un Servicio Web que permita adquirir los datos del odómetro, horas motor, consumo del vehículo y datos personalizados que se encuentran en una base de datos de un servidor.
- Realizar un Servicio Web que permita la modificación de los datos de los vehículos que han realizado mantenimiento calculando los próximos servicios.
- Plantear el historial del módulo de mantenimiento vehicular, permitiendo al usuario visualizar e informarse acerca de los servicios de sus vehículos.

- Efectuar pruebas funcionales al módulo de mantenimiento vehicular y aplicar las respectivas correcciones.
- Construir una reflexión humanista en la que se describa el impacto social, económico y cultural.

1.4 METODOLOGÍA

Con el fin de mantener un orden, es importante seccionar la metodología por etapas que facilitan y determinan el progreso durante el proyecto.

En primer lugar, está la etapa de entendimiento del problema; luego, la etapa de investigación y documentación; en tercer lugar, la etapa de especificaciones; en cuarto lugar, la etapa de desarrollo del problema; en quinto lugar, la etapa de pruebas y correcciones y se finaliza, con la etapa de aprobación y entrega.

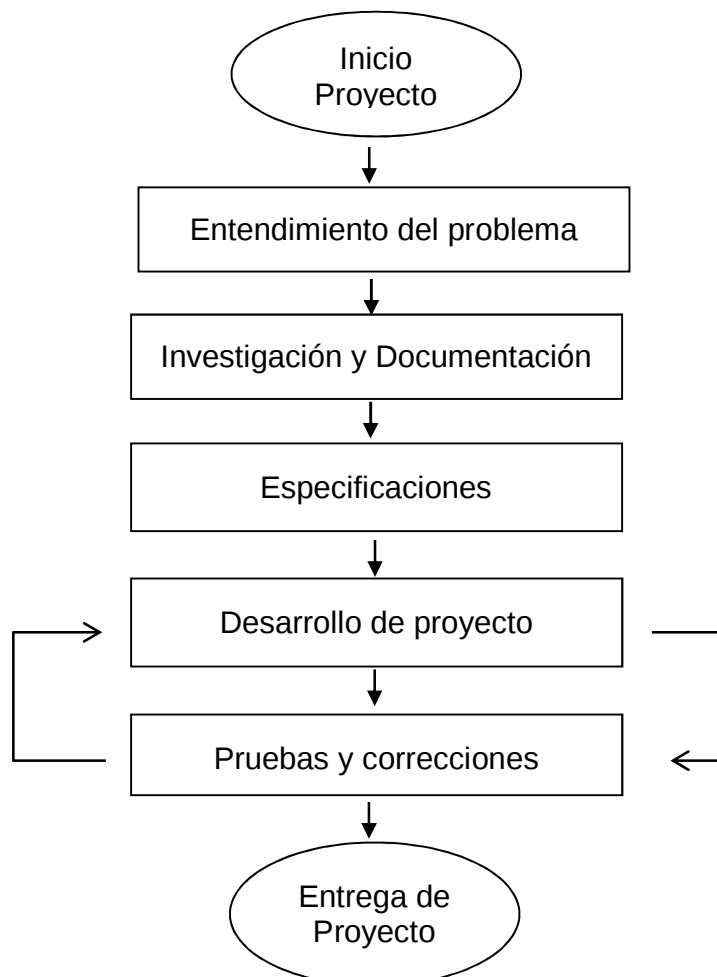


Figura 1. Etapas del proyecto

1.4.1 ETAPA DE ENTENDIMIENTO DEL PROBLEMA

Es importante conocer bien el problema, que en este caso, es el desarrollo de un módulo de mantenimiento vehicular flexible, junto con un historial de los servicios realizados a las unidades; debido a que se deben determinar que conceptos se requieren para solucionarlo. Como por ejemplo, en donde almacenar la información o como interfazarla. Gracias a que se tiene como base la aplicación FleetSAP, fue sencillo reconocer que se debe almacenar en la base de datos, manejar la información con los servicios web y mostrarla mediante una forma de Windows.

1.4.2. ETAPA DE INVESTIGACIÓN Y DOCUMENTACIÓN

En esta etapa, es importante documentarse sobre lo que se requiere para el desarrollo de la aplicación, estudiar conceptos básicos de C# en programación orientada a objetos, de bases de datos en SQL en cuanto a creación de scripts y manejo de tablas, de servicios web en cuanto a procedimientos almacenados y métodos.

1.4.3. ETAPA DE ESPECIFICACIONES

En esta etapa, se debe realizar un cuadro de especificaciones en donde de forma ordenada se entrega a la empresa un documento que lleve consignado paso a paso el desarrollo del proyecto, especificando los elementos a usar y los screenshots del diseño que finalmente quedará.

1.4.4 ETAPA DE DESARROLLO DEL PROBLEMA

En esta fase, se deben tener muy claras las etapas 1,2 y 3. Debido a que son ellas, quienes darán la base de creación del proyecto. La etapa finaliza cuando ya se tenga en completa funcionalidad las especificaciones de la etapa 3.

1.4.5 ETAPA DE PRUEBAS Y CORRECCIONES

En esta etapa, se hacen pruebas de diseño y funcionamiento. De tal forma que cumpla con los estándares de la compañía y con las especificaciones del proyecto en su totalidad. Si en la revisión se requieren corrección de errores, se debe de hacer, para luego continuar con las pruebas hasta que se encuentre en óptimas condiciones la aplicación.

1.4.6 ETAPA DE ENTREGA

Ya para terminar en esta etapa, se hacen entrega de los scripts de las tablas creadas o modificadas, las formas de Windows modificadas y creadas y los nuevos servicios web para subir la nueva versión a producción del software principal.

1.5 DELIMITACIÓN DEL PROYECTO

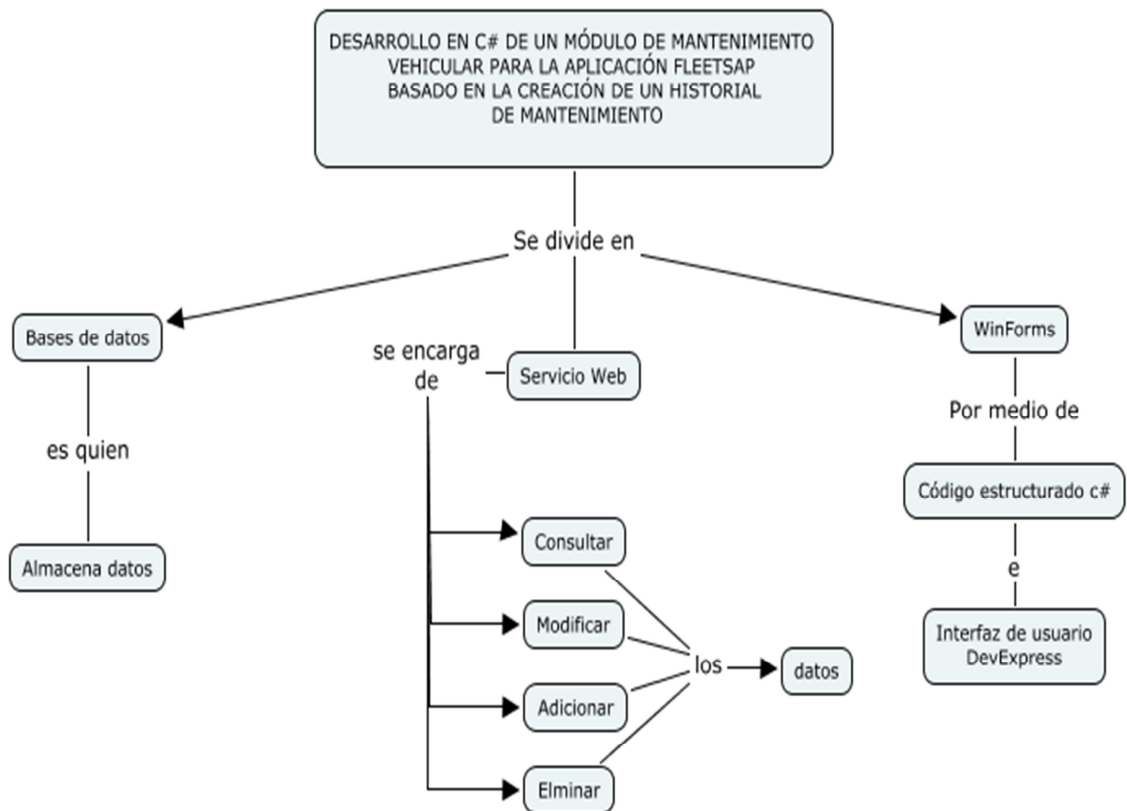


Figura 2: Delimitación del proyecto.

En la figura 2, se pueden apreciar los tres pasos básicos en los que se basa el proyecto. En primer lugar, se tiene la base de datos. Hay que determinar que se necesita implementar dentro de ella, teniendo en cuenta las normativas de SCI. En segundo lugar, los servicios web. Quienes se encargaran de manipular los registros en la base de datos. Ya sea para eliminar, adicionar, modificar o consultar los datos. Y por último, la interfaz. Es así como las formas de Windows tienen el protagonismo, debido a su facilidad de adaptación con los dos anteriores.

II. LA EMPRESA

2.1 SCI SUR AMÉRICA SAS

2.1.1 RESEÑA

Es una compañía de Colombia que se encarga de la gestión de flotas, permitiendo el control vehicular en diferentes empresas. Ésta es una subdivisión de la empresa SCI Distribution LLC de Florida, Estados Unidos.

Esta compañía ha demostrado ser pionera a la hora de brindar soluciones de este tipo, ya que siempre busca optimizar los sistemas integrados de Hardware y Software en los vehículos, innovar con dispositivos de control de unidades, además de un continuo compromiso con los clientes.

2.1.2 ACTIVIDADES

SCI SUR AMERICA SAS tiene como objeto social: La prestación de los servicios consultoría , asesoría, control, instalación y administración de herramientas y soluciones informáticas así como la construcción, desarrollo, evaluación, comercialización, distribución de todo tipo de herramienta informática, tecnológica, hardware, software y servicios afines; dentro y fuera del país. La representación, Comercialización de elementos, materiales, tecnología y servicios afines. Se realizan servicios de evaluación, preventa, pos venta e implementación de software, hardware, elementos tecnológicos que sirvan para la cobertura interna y externa de los diferentes sistemas desarrollados, el soporte técnico, comercialización y representación de elementos y materiales de tecnología y servicios afines con la función de desarrollo. Se prestan servicios de

mantenimiento, administración e instalación de sistemas de información, redes estructuradas y equipos electrónicos y de cómputo en general.¹

2.2 PASANTÍA EMPRESARIAL EN SCI SUR AMERICA SAS

2.2.1 DESARROLLO

La pasantía empresarial inició con una documentación completa sobre los productos y servicios que ofrece la compañía. Posteriormente se otorgó el proyecto de desarrollar un módulo de mantenimiento vehicular en la plataforma FleetSAP, así se fundamentó bajo un documento de especificaciones las actividades a realizar junto con un Project donde se evidenció el cronograma y duración que llevaría.

2.2.2 FUNCIONES

Diseñar el proyecto entregado, teniendo en cuenta bases de datos, servicios web y desarrollo de formas de Windows, además de realizar las debidas pruebas. Sin embargo, durante este desarrollo, se asignaban tareas de mejoramiento al software FleetSAP de diseño, implementación de idiomas extranjeros y creación de nuevos reportes de usuario.

¹ <http://www.computrabajo.com.co/bt-empd-programacion.html>

III. MARCO TEÓRICO

Para el desarrollo del proyecto se hace necesario la aplicación de una plataforma de desarrollo, que permita la implementación de aplicativos web, bases de datos y administración fácil de la información, en este orden de ideas la empresa ejecuta estas labores de diseño con la herramienta Visual Studio Express que tiene todas las cualidades necesarias para un proyecto de las características mencionadas, a continuación se hace una breve introducción a este producto.

3.1. Visual Studio Express

Visual es una colección de herramientas y servicios que permite crear una gran variedad de aplicaciones para plataformas como Microsoft. Tiene la capacidad de conectarse con varios proyectos y equipos. Su principal importancia está dada por la gran agilidad desde casi cualquier parte, independientemente de la herramienta de desarrollo, incluidas Eclipse y Xcode.

La primera versión de Visual Estudio Express fue la 2005, siendo liberada en Octubre de 2005, y la versión con SP1, fue liberada en Diciembre de 2006. Estas versiones se ejecutaban en Windows 2000 SP4 y sobre plataformas Basadas en Windows NT.

Éste lenguaje de programación se utiliza para desarrollar la aplicación FleetSAP, debido a su sencilla estructura, su orden y la posibilidad de desarrollar programación orientada a objetos que facilita la estructuración de un programa en general.

La principal razón por la que se utiliza es por brindar la posibilidad de crear aplicaciones de Windows por medio de herramientas de interacción siendo más entendible al usuario por su interfaz gráfica. Además de permitir la interacción con servicios web y bases de datos.²

3.2. C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET; su sintaxis básica deriva de C++ y utiliza el modelo de objetos de la plataforma .NET similar a Java, incluyendo por supuesto mejoras derivadas de otros lenguajes.

En el año 2000 Microsoft presenta C# diseñado por Anders Hejlsberg, éste es un híbrido entre C++ y Java que fusiona en cuanto a la capacidad de combinar operadores con la plena orientación a objetos respectivamente. La orientación a objetos es tal que el propio programa está encapsulado en una clase. Actualmente se encuentra entre los diez lenguajes más utilizados. A pesar de su corta historia ha sido aceptado por ECMA y por ISO.³

Este lenguaje de programación es preferido por la compañía SCI Sur América debido a que su tamaño es pequeño, puede realizar llamado a varias funciones, cuenta por palabras reservadas, es estructurado y tiene la posibilidad de usar apuntadores.

3.2.1. Clases

Una clase es una estructura que contiene funciones, propiedades, variables y datos en general de un programa. Hace las veces de una plantilla que permite

² <http://www.visualstudio.com/es-es/visual-studio-homepage-vs.aspx>

³ <http://jorgesaavedra.wordpress.com/2006/12/09/breve-historia-de-c-c-c/>

crear objetos. Éstos últimos hacen uso de las funciones para entregar un producto terminado.

3.2.2. Constructor

Es una función o rutina que tiene como objetivo la inicialización de variables en una clase. Normalmente, los constructores tienen el mismo nombre de la clase y en algunas ocasiones puede haber más de uno pero con diferentes parámetros de entrada.

3.3. SQL Server Management Studio

Este es un software de Microsoft que permite administrar bases de datos de forma que se pueda consultar, modificar y configurar servicios web que se vinculen directamente con la aplicación. Además permite la creación de scripts quienes se encargan de proporcionar el acceso en la aplicación.

De igual forma, este software permite la configuración de cliente-servidor de tal forma que el cliente solo puede tener la opción de visualizar los datos del servidor. En 1974 Donald Chamberlin, investigador de IBM implemento varios prototipos SEQUEL, para finalmente entre 1976 y 1977 crear lo que hoy en día conocemos como SQL.

Una de las principales ventajas de SQL es el Resource Governor (Gobernador de recursos). Ésta herramienta permite vigilar a los usuarios para que no usen todos los recursos de la base de datos con solicitudes. Al usar el servidor SQL, puede decidir qué ancho de banda pueden tomar los servidores y cuando se llegue a

cierto nivel de uso, se puedan detener automáticamente los procesos que estén tomando demasiados recursos.⁴

3.3.1. Tablas

Para la creación de una tabla se utiliza la instrucción `CREATE TABLE`, allí se definen los campos o columnas que va a contener, además de ciertos atributos como las llaves, tipo de dato, nulabilidad y cálculos.

3.3.1.1. Modificación

Una modificación permite alterar los campos que contiene, entre ellas se puede eliminar, adicionar o actualizar.

Para poder realizar esto, es importante conocer la instrucción `ALTER TABLE`, que significa que se va a modificar los campos de la tabla, posteriormente se indica el nombre de la tabla y el tipo de modificación ya sea: `ADD` si se va a adicionar un campo, `DROP` si se elimina, o `ALTER` si se modifica. Finalmente, el nombre del campo o columna.

3.3.1.2. Eliminación

La instrucción de eliminación es `DROP`. Su modo de uso es colocar la instrucción seguida del nombre de la tabla.

Las consultas, hacen parte fundamental. Ya que finalmente son los registros quienes nos interesan.

⁴ <http://www.microsoft.com/es-co/download/details.aspx?id=7593>

3.3.2. CONSULTAS

Es un llamado que hace una o varias tablas con el fin de obtener registros relevantes.

3.3.2.1. SELECT

La instrucción SELECT se utiliza para seleccionar los campos de las tablas de consulta que se requieren extraer. Su modo de uso es el siguiente:

En primer lugar se escribe la instrucción SELECT, luego los campos que se desean obtener y finalmente la instrucción FROM que es en donde se encuentran los campos, normalmente, se introduce allí el nombre de la tabla que los contiene.

3.3.2.2. WHERE

Esta instrucción es quien delimita las restricciones de la consulta con el fin de no traer todos los registros sino los necesarios. Se usan clausulas como AND, OR o ISNULL.

3.3.2.3. GROUP BY

Ésta instrucción es usada para agrupar los registros traídos. Se usa, incorporando los campos que se requieren agrupar. Es una manera de presentar mejor los resultados.

3.3.2.4. ORDER BY

Es una cláusula que se encarga de organizar de manera ascendente o descendente los registros por un campo en específico.

3.3.2.5. DELETE

Es una función que permite eliminar los registros de una tabla. Se hace uso la condición WHERE para escoger los registros erróneos o viejos.

3.3.2.6. UPDATE

Esta función permite actualizar los registros existentes, ya sea por una fecha, una descripción u otro tipo de dato que se vaya adicionando conforme pasa el tiempo.

3.3.2.7. INSERT

Esta función permite ingresar nuevos campos en las tablas involucradas. Para realizar una inserción se hace uso de la palabra reservada INSERT INTO, seguido de la tabla que llevara el nuevo registro; luego, se nombran las columnas o campos que se va a ingresar y finalmente se ingresa la palabra reservada VALUES y los nuevos datos del registro.

3.4. SERVICIOS WEB

Un servicio Web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones de software desarrolladas en lenguajes de programación diferentes, que están ejecutadas sobre cualquier plataforma a través de internet.

Como principales ventajas, se pueden mencionar los aportes por interoperabilidad entre aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen, fomentan los estándares y protocolos basados en texto, que hacen más fácil accederá su contenido y entender su

funcionamiento y permitir que servicios y software de diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.⁵

3.5. ESTANDAR PARA ESPECIFICACIONES

ERS (Especificación de requisitos de software) es la descripción del comportamiento de los sistemas a desarrollar. En él se incluye un conjunto de casos en el que se incluyen todos los requisitos funcionales y no funcionales con el usuario. Estas especificaciones se definen en el estándar IEEE 830-1998 y están dirigidas tanto al cliente como a equipo de desarrollo.

El formato que se propone básicamente tiene:

- a. Introducción: Presenta la contextualización, el alcance y dar una breve explicación o resumen del software propuesto.
- b. Objetivos: se refieren a las estrategias para cumplir el software.
- c. Características: Proponen dar una perspectiva del software, mostrar las funciones del producto, las características físicas, las restricciones, suposiciones y dependencias, en donde intervienen las interfaces externas, las funciones, los requisitos de rendimiento, las restricciones de diseño y los atributos del sistema.
- d. Conclusiones.
- e. Bibliografía.

Sin embargo, SCI Sur América plantea una versión basada en el estándar anterior de especificaciones. En donde se plantea:

- a. Situación actual: Se explica en qué situación se encuentra FleetSAP (Software principal de la compañía).

⁵ <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>

- b. Definición de negocio: Se muestra lo nuevo que llevara la aplicación FleetSAP.
- c. Objetivos: Se presentan las estrategias que se seguirán para realizar el software.
- d. Premisas: Se exponen los cambios y adecuaciones que se llevaran a cabo dentro de FleetSAP con sus respectivos pantallazos de como quedaran las formas de Windows ya existentes.
- e. Requerimientos funcionales: Se explican las nuevas formas de Windows con sus respectivos pantallazos (Diseño).
- f. Requerimientos no funcionales: Se definen las bases de datos, los servicios web y los nombres de las formas de Windows nuevas.
- g. Anexos: Por si se requiere especificar algunos de los ítems anteriores.

IV. EL PROYECTO

4.1 DESARROLLO DEL PROYECTO

De acuerdo a la propuesta de desarrollo que se presentó en el capítulo III, se inicia el proyecto con el entendimiento del problema, entendiendo los productos y servicios que se manejan en la empresa SCI Sur América.

En este orden de ideas, la segunda etapa continúa con la investigación y la documentación del proyecto, en donde se realizaron las respectivas lecturas sobre conceptos básicos de programación en c#, SQL, Servicios Web, construcción de formas de Windows; basado en la construcción de ejercicios prácticos.

Después de conocer la teoría base para la construcción del proyecto, se inicia la etapa de especificaciones en donde se realiza un cronograma y un documento de detalles que muestre las herramientas con las que se va a trabajar.

El documento de especificaciones presentado a SCI Sur América, fue nombrado como `Specification_0000_20140202_Maintenance_Module`. Consta del siguiente contenido:

- a. Situación actual: Actualmente FleetSAP ofrece la posibilidad de realizar un mantenimiento preventivo a cada uno de los vehículos que soporta cada compañía. Dicho mantenimiento se basa en los valores actuales de odómetro, horas motor y consumo de combustible acumulado o una fecha estipulada para el siguiente servicio. Del mismo modo los usuarios están en la capacidad de elegir uno de los tres criterios mencionados y recibir u obtener un listado de los vehículos próximos a mantenimiento.

- b. Definición del negocio: Aprovechando ésta funcionalidad se requiere complementar la información de mantenimiento preventivo, entregando un módulo de mantenimientos definidos por el mismo usuario como una nueva opción en la sección de Mantenimiento de vehículos.

Tanto el mantenimiento preventivo como el módulo se unificarán dentro de una misma sección llamada Módulo de mantenimientos.

Por consiguiente el usuario sólo visualizará de manera informativa data sobre odómetro, horas motor y volumen de combustible acumulado desde la sección de Mantenimiento de vehículos y podrá modificarla desde la nueva sección Módulo de mantenimientos.

c. Objetivos:

- I. Crear una nueva sección llamada Módulo de mantenimientos que le permita al usuario manejar de forma unificada el mantenimiento preventivo y el módulo de los mantenimientos realizados a cada uno de sus vehículos.
- II. Establecer una sección de tipos de mantenimiento que va ligada a la creación o modificación de los mantenimientos realizados a cada vehículo.
- III. Implantar una estructura en bases de datos que pueda soportar el módulo, los tipos de mantenimiento y el mantenimiento preventivo.
- IV. Realizar los servicios web correspondientes al módulo de mantenimiento, los tipos de mantenimiento y mantenimiento preventivo.

d. Premisas:

- I. Documentación de Mantenimiento preventivo.
- II. Adecuaciones en la forma de Vehículos.
 - ✓ Eliminar los tabs de Odometro, horas motor, consumo de combustible y fecha.

- ✓ Agregar un botón que permita visualizar el módulo de mantenimientos en donde se encuentra el Combobox de tipos de mantenimiento.
- ✓ Condensar la información de Odómetro, horas, combustible acumulado en el tab de Mantenimiento preventivo.
- ✓ Agregar un botón que abre la forma Modulo de mantenimiento.

e. Requerimientos funcionales:

- I. Forma de Tipos de mantenimiento: Esta forma permitirá al usuario agregar, modificar o eliminar los tipos de mantenimiento de manera personalizada. Esto quiere decir que el usuario podrá crear lo tipos acorde a los mantenimientos que sean realizados a sus vehículos.
 - Nombre = Tipos de Mantenimiento
 - Forma = fSys79MaintenanceTypes
 - Font = Microsoft Sans Serif, 8,25pt
 - Tamaño= 546, 500
 - Campos:
 - o ID del tipo de mantenimiento
 - o Descripción del tipo de mantenimiento
 - o Fecha de creación
 - o Fecha de actualización
 - o Actualizado
 - Botones:
 - o BindingSource
 - o Adicionar
 - o Eliminar
 - o Guardar
 - Búsqueda:
 - o ID

- o Descripción
- Banner de parámetros.
- Idiomas
 - o Español(Default)
 - o Inglés
 - o Turco
 - o Hebreo

II. Módulo de Mantenimiento:

En éste módulo el usuario tendrá la opción de ingresar los mantenimientos realizados a sus vehículos.

- Nombre = Módulo de mantenimiento
- Forma = fSys80MaintenanceModule
- Font = Microsoft Sans Serif, 8,25pt
- Tamaño = 1149, 729
- Secciones :
 - o Historial de mantenimientos
 - o Mantenimiento preventivo
- Campos:
 - ❖ Historial de mantenimientos:
 - ID del vehículo
 - Descripción del Vehículo
 - Fecha y hora del mantenimiento
 - Fecha (Control 1)
 - Hora(Control 2)
 - Tipo de mantenimiento.
 - Odómetro en el momento del mantenimiento
 - Horas Motor en el momento del mantenimiento
 - Costo

- Notas
- Fecha de creación
- Fecha de actualización
- Actualizado
- o Grilla de historial del mantenimiento
- o Botones:
 - BindingSource
 - Adicionar
 - Eliminar
 - Guardar
- ❖ Mantenimiento preventivo: El usuario podrá setear los valores actuales del odómetro, horas motor y Combustible máximo volumen. Del mismo modo podrá hacer cambios sobre los campos *Definición de usuario* 1al 5 de la descripción y sus valores actuales, delta, siguiente, fecha y si se debe informar sobre dicho mantenimiento.
 - o Odometro
 - Actual
 - Delta
 - Siguiente
 - Fecha de odómetro
 - o Horas motor
 - Actual
 - Delta
 - Siguiente
 - Fecha de Horas motor
 - o Combustible
 - Acumulado
 - Delta = 0

- Máximo
 - Fecha de combustibles
- o Arreglo de CheckBox con las tres opciones
 - Odómetro
 - Horas Motor
 - Combustible.
- o Adición de 5 campos personalizables de mantenimiento preventivo:
 - Nombre del campo(Personalizable)
 - Actual
 - Delta
 - Siguiendo
 - Fecha del campo.
- o Banner de Vehículos.
- o Idiomas
 - Español(Default)
 - Inglés
 - Turco
 - Hebreo
- o Botones:
 - Guardar

f. Requerimientos no funcionales: Las tablas creadas en la base de datos fueron:

I. PREVENTIVEMAINTENANCETYPE

La tabla 1, describe los campos utilizados para realizar la administración de los registros de los servicios personalizados vehiculares que se van a programar por compañía.

NOMBRE DEL CAMPO	TIPO	NULABILIDAD	DESCRIPCIÓN
Id	Identificador único	Not null	Identificador único de la tabla
CompanyId	Int	Not null	Id de la compañía
MaintenanceId	Int	Not null	Id del tipo de mantenimiento
Description	Int	Null	Descripción del tipo de mantenimiento

Tabla 1. Tabla PreventiveMaintenanceType

II. PREVENTIVEMAINTENANCE

La tabla 2, permite visualizar los campos detallados que contiene. Éstos forman parte de la forma de mantenimiento preventivo en donde se desea administrar los registros pertenecientes a la programación de servicios vehiculares personalizados.

NOMBRE DEL CAMPO	TIPO	NULABILIDAD	DESCRIPCIÓN
Id	Identificador único	Not null	Identificador único de la tabla
CompanyId	Int	Not null	Id de la compañía

Carnumber	Nvarchar 10	Not null	Id del vehículo
MaintenanceId	Int	Not null	Id del tipo de mantenimiento
IsMaintenance	Boolean	Null	False= no es interactivo True = es interactivo
CurrentValue	Float	Null	Valor actual del mantenimiento
Delta	Int	Null	Delta del mantenimiento
Nextvalue	Float	Null	Valor siguiente del mantenimiento
Nextdate	Datetime	Null	Fecha siguiente mantenimiento

Tabla 2. Tabla PreventiveMaintenance

III. MAINTENANCETYPES

La tabla 3, hace referencia a los tipos de mantenimiento que han sido practicados a los vehículos. A continuación se evidencian los campos.

NOMBRE DEL CAMPO	TIPO	NULABILIDAD	DESCRIPCIÓN
Id	Identificador único	Not null- autoincremental de 1-1	Identificador único de la tabla
CompanyId	Int	Not null PK	Id de la compañía
MaintenanceId	Int	Not null PK	Id del tipo de mantenimiento

MaintenanceDescription	Nvarchar 500	Null	Descripción del tipo de mantenimiento
CreatedDate	Datetime	Null	Fecha de creación
UpdatedDate	Datetime	Null	Fecha de actualización
Updated	Boolean	Null	True actualizado False sin actualizar
UserName	Nvarchar 256	Null	Auditorias
HostName	Nvarchar 30	Null	Auditorias
IpAddress	Nvarchar 30	Null	Auditorias
AppName	Nvarchar 30	Null	Auditorias
Audit	Boolean	Null	Auditorias

Tabla 3. Tabla MaintenanceTypes

IV. MAINTENANCEHISTORY

La tabla 4, muestra los campos usados en la administración de registros provenientes del historial de mantenimiento de cada compañía.

NOMBRE DEL CAMPO	TIPO	NULABILIDAD	DESCRIPCIÓN
Id	Identificador único	Not null autoincremental de 1-1	Identificador único de la tabla
CompanyId	Int	Not null -PK	Id de la compañía
CarNumber	Nvarchar 10	Not null –PK FK-cars	Identificador del vehículo.
MaintenanceId	Int	Not null FK-PK FK-	Id del tipo de mantenimiento

		MaintenanceTypes	
Date	Datetime	Null-PK	Fecha siguiente mantenimiento
Odometer	Float	Null	Valor actual del odómetro
EngineHours	Int	Null	Valor actual de las horas motor
Cost	Float	Null	Costo del mantenimiento
Notes	Nvarchar 500		Notas relacionadas al mantenimiento.
CreatedDate	Datetime	Null	Fecha de creación
UpdatedDate	Datetime	Null	Fecha de actualización
Updated	Boolean	Null	True actualizado False sin actualizar
UserName	Nvarchar 256	Null	Auditorias
HostName	Nvarchar 30	Null	Auditorias
IpAddress	Nvarchar 30	Null	Auditorias
AppName	Nvarchar 30	Null	Auditorias
Audit	Boolean	Null	Auditorias

Tabla 4. Tabla MaintenanceHistory

Los servicios web creados para la manipulación de las tablas fueron:

- ✓ GetMaintenanceTypes que cuenta con parámetros de entrada:
 - CompanyID: identificador de la compañía.
 - MaintenanceID: identificador de tipo de mantenimiento.

- MaintenanceDescription: Descripción del mantenimiento.
- DateCreated: Fecha de creación del tipo de mantenimiento.
- UpdatedDate: Fecha de actualización del tipo de mantenimiento.
- Updated: Bandera de actualizado.
- Username: Auditorías.
- Hostname: Auditorías.
- Ipaddress: Auditorías.
- Appname: Auditorías.
- Audit: Auditorías.
- out dsData: Dataset que retorna el método de consulta.
- out sErrMess: Error de salida, en caso de haber alguna excepción en la consulta.

✓ DeleteMaintenanceTypes tiene parámetros de entrada como:

- ID: Identificador interno del mantenimiento.
- CompanyID: Identificador de la compañía.
- MaintenanceID: Identificador del mantenimiento.
- MaintenanceDescription: Descripción del mantenimiento.

✓ AddMaintenanceTypes que cuenta con parámetros de entrada:

- CompanyId: Identificador de la compañía.
- MaintenanceID: Identificador del mantenimiento.

- MaintenanceDescription: Descripción del mantenimiento.
 - UserName: Auditorias.
 - HostName: Auditorias.
 - IPAddress: Auditorias.
 - AppName: Auditorias.
 - Out sErrMess: Mensaje de salida, por si falla la adición del registro.
- ✓ ModifyMaintenanceTypes: Tiene parámetros de entrada como:
- CompanyId: Identificador de la compañía.
 - MaintenanceID: Identificador del mantenimiento.
 - MaintenanceDescription: Descripción del mantenimiento.
 - UserName: Auditorias.
 - HostName: Auditorias.
 - IPAddress: Auditorias.
 - AppName: Auditorias.
 - Out sErrMess: Mensaje de salida, por si falla la modificación del registro.
- ✓ GetMaintenanceHistory: Tiene parámetros de entrada
- Id: Identificador del mantenimiento interno.
 - CarNumber: Identificación del vehículo.
 - MaintenanceId: Identificación del mantenimiento.
 - CompanyId: Identificador de la compañía.

- ✓ AddMaintenanceHistory: Tiene como parametros de entrada
 - CompanyId: Identificador de la compañía.
 - CarNumber: Identificador del vehículo.
 - Maintenceld: Identificador del mantenimiento.
 - DateMaintenance: Fecha del mantenimiento.
 - Odometer: Valor actual del odometro
 - EngineHours: Valor actual de las horas motor.
 - Cost: Costo del mantenimiento.
 - Notes: Notas del mantenimiento.
 - UserName: Auditorias.
 - HostName: Auditorias.
 - IpAddress: Auditorias.
 - AppName: Auditorias.
 - ErrMess: Devuelve un error por si falla la adición del registro.

- ✓ ModifyMaintenanceHistory: Tiene como parametros de entrada
 - CompanyId: Identificador de la compañía.
 - CarNumber: Identificador del vehículo.
 - Maintenceld: Identificador del mantenimiento.
 - DateMaintenance: Fecha del mantenimiento.
 - Cost: Costo del mantenimiento.
 - Notes: Notas del mantenimiento.
 - UserName: Auditorias.
 - HostName: Auditorias.
 - IpAddress: Auditorias.
 - AppName: Auditorias.

- ErrMess: Devuelve un error por si falla la adición del registro.
- ✓ DeleteMaintenanceHistory: En donde sus parámetros de entrada son:
- Id: Identificador único
 - CompanyId: Identificador de la compañía
 - MaintenanceId: Identificador del mantenimiento.
 - CarNumber: Identificador del vehículo.
 - sErrMess: Devuelve si se pudo o no eliminar el registro.

Para luego continuar con la etapa de desarrollo en la que realizó la creación de las tablas en la base de datos, la creación de los servicios web, la implementación de las clases y las formas de Windows en el proyecto FleetSAP, teniendo como resultado:

1. Bases de datos: En las figuras 3 y 4, encontramos la creación de las tablas mencionadas en el numeral anterior.

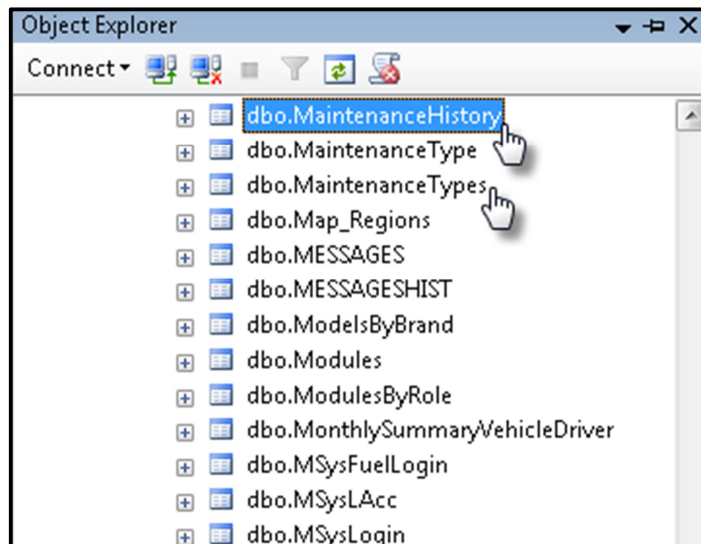


Figura 3: Tablas MaintenanceHistory y MaintenanceTypes.

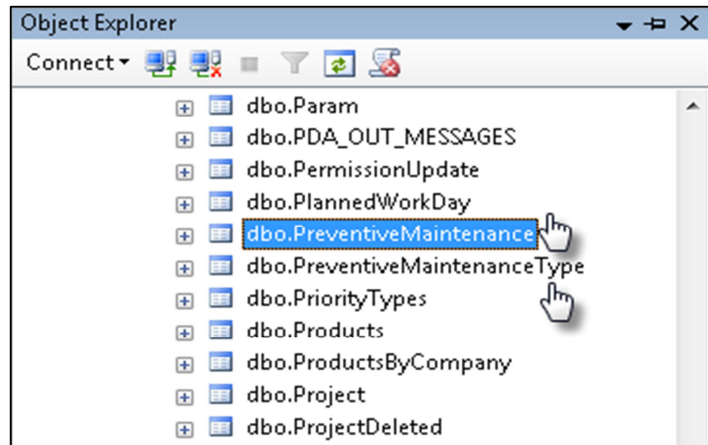


Figura 4. Tablas PreventiveMaintenance y PreventiveMaintenanceType.

Ahora, la evidencia de la creación de los servicios web, se encuentran en las figuras 5, 6, 7 y 8. Vemos en la parte superior, la dirección web a la que apunta la aplicación.

En la figura 5, GetMaintenanceHistory es el servicio web encargado de realizar la consulta de los registros referentes al historial de mantenimiento vehicular. Éste lleva a FleetSAP el dataset correspondiente a los datos buscados. En tanto que GetMaintenanceTypes, trae los registros almacenados en la tabla MaintenanceTypes referentes a los tipos de servicios.

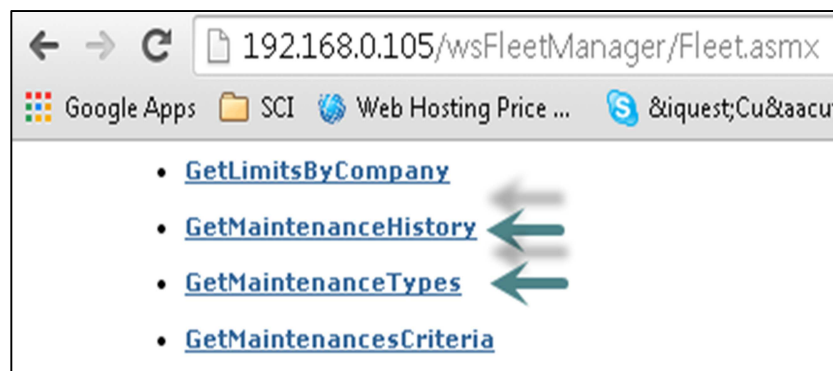


Figura 5. Servicios web de consulta.

En la figura 6, el servicio AddMaintenanceHistory captura los datos a almacenar en un registro y los lleva a la tabla MaintenanceHistory. Y el servicio AddMaintenanceTypes es quien almacena en la tabla MaintenanceTypes de la base de datos los registros pertenecientes a los tipos de mantenimiento.

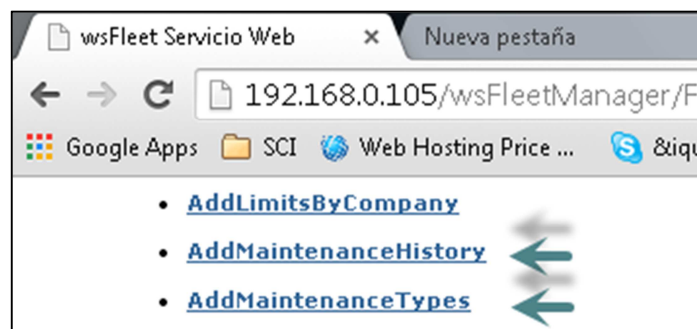


Figura 6. Servicios web de adición de registros.

En la figura 7, se muestran los servicios de modificación de registros. ModifyMaintenanceHistory, es el encargado de capturar los valores de los campos a modificar de la tabla MaintenanceHistory. En tanto que ModifyMaintenanceTypes es quien captura los campos con los datos a modificar de la tabla MaintenanceTypes de la base de datos.

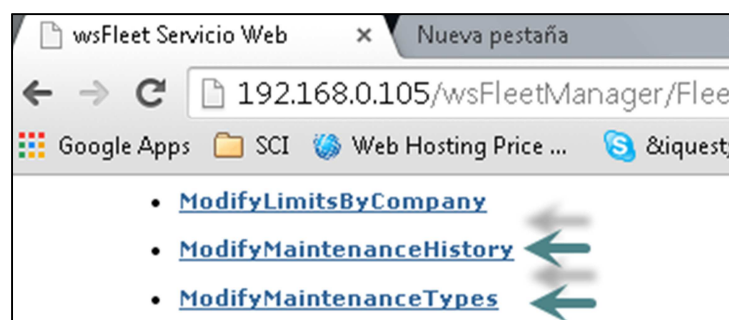


Figura 7. Servicios web de modificación de registros.

En la figura 8, se encuentran los servicios web encargados de eliminar los registros de las tablas MaintenanceHistory y MaintenanceTypes respectivamente.

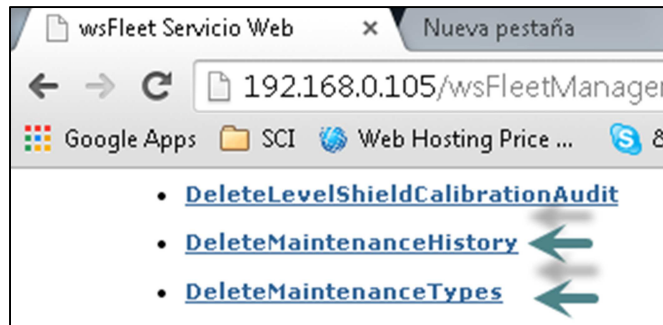


Figura 8. Servicios web de eliminación de registros.

2. La modificación de la forma de vehículos, se inicia con la eliminación del tab de mantenimiento preventivo con el fin de realizar un módulo independiente, es decir, en una forma de Windows diferente. Ahora, se adiciona un tab que recibe el nombre de *información de mantenimiento* en donde se agregan 3 campos y un botón. Los campos son respectivamente el valor actual del odómetro, horas motor y consumo de combustible acumulado, el botón se nombró como módulo de mantenimiento ya que direcciona a la nueva forma.

Figura 9: Forma de mantenimiento de flota.

En la figura 9, se muestra la distribución de los campos y del botón. Es así como se evidencia el cambio de la forma y del tab como tal.

En esta forma no se implementó ningún servicio web nuevo y tampoco se crearon campos en las tablas que ya existían referentes a Cars.

3. El módulo de mantenimiento al que se da acceso consta de dos partes principalmente presentadas como tabs: El historial de mantenimiento y el mantenimiento preventivo.

Dentro del tab “historial de mantenimiento”, se crean dos tabs que permiten visualizar en una grilla los registros creados y manipular los registros ya sea para

crear, modificar o eliminarlos (AddMaintenanceHistory, ModifyMaintenanceHistory, DeleteMaintenanceHistory). Dentro del primer tab encontramos los siguientes campos:

2.2. Información del vehículo:

2.2.1. ID de vehículo

2.2.2. Descripción del vehículo.

2.3. Información del mantenimiento

2.3.1. Tipo de mantenimiento

2.3.2. Fecha

2.3.3. Hora

2.3.4. Odómetro

2.3.5. Horas motor

2.3.6. Costo

2.3.7. Notas

2.4. Log

2.4.1. Fecha de creación

2.4.2. Fecha de actualización

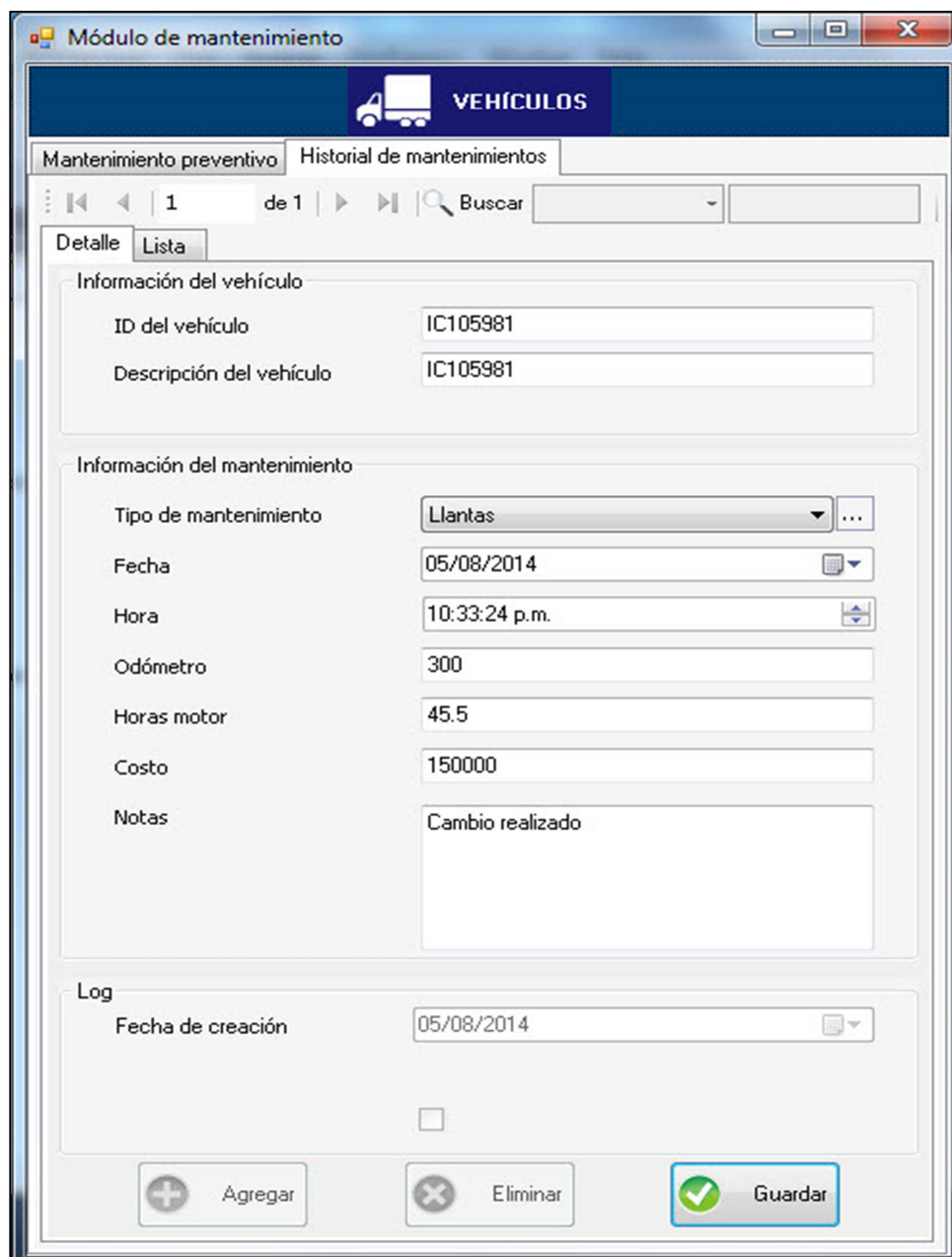
2.4.3. Actualizado.

En este orden de ideas, los campos del Groupbox “Información del vehículo” son quienes presentan la identificación del vehículo y el nombre del vehículo.

El siguiente Groupbox presenta informativamente el valor del odómetro y de las horas motor del vehículo al momento del mantenimiento. El campo de fecha y hora permiten seleccionar el tiempo en el que se hizo éste. Los campos de costo y

notas permiten digitar como su nombre lo indica el costo del servicio y el segundo digitar algunas especificaciones que se requieran.

Por último el Groupbox “Log” permite visualizar la fecha de creación del registro, la fecha en la que se actualizó y un Checkbox que muestra si está actualizado o no.



Módulo de mantenimiento

VEHÍCULOS

Mantenimiento preventivo | Historial de mantenimientos

1 de 1 | Buscar

Detalle | Lista

Información del vehículo

ID del vehículo: IC105981

Descripción del vehículo: IC105981

Información del mantenimiento

Tipo de mantenimiento: Llantas

Fecha: 05/08/2014

Hora: 10:33:24 p.m.

Odómetro: 300

Horas motor: 45.5

Costo: 150000

Notas: Cambio realizado

Log

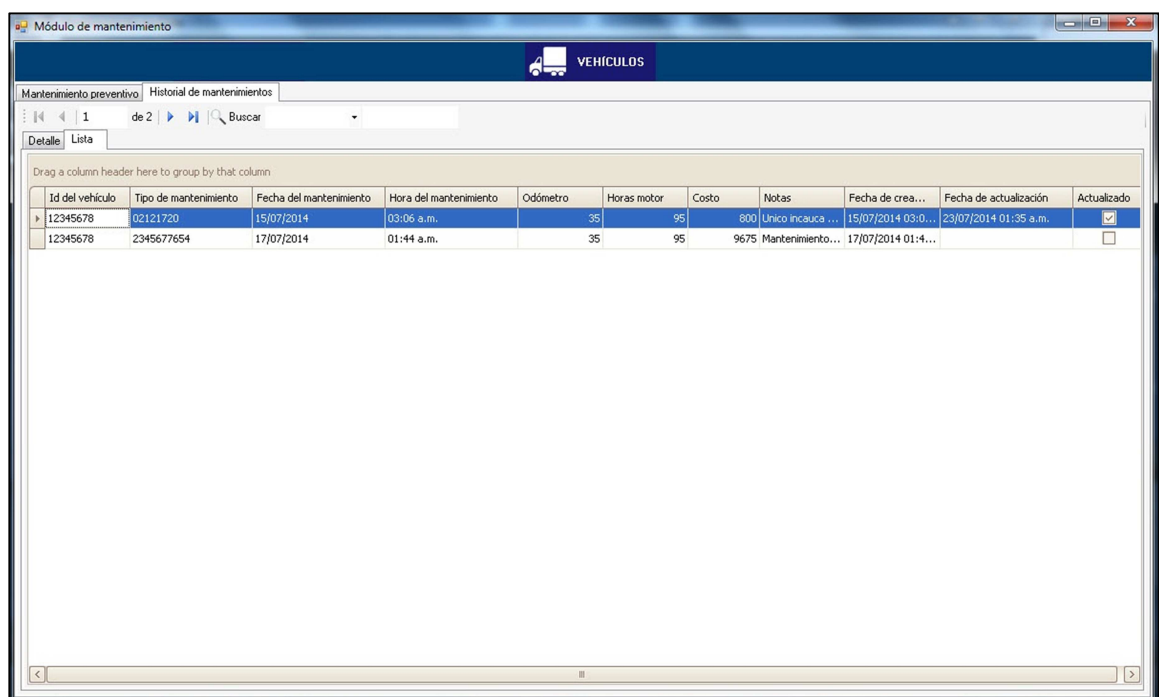
Fecha de creación: 05/08/2014

☐

+ Agregar - Eliminar ✓ Guardar

Figura 10: Forma nueva denominada módulo de mantenimiento

En la figura 10, se evidencia la distribución de los campos con sus respectivos labels indicativos. Este historial, permite administrar todos los mantenimientos realizados a la unidad, pero si se requiere sólo visualización de todos los mantenimientos registrados se puede visualizar en el tab lista como se muestra en la figura 10 (Hace uso del servicio web GetMaintenanceHistory).



Id del vehículo	Tipo de mantenimiento	Fecha del mantenimiento	Hora del mantenimiento	Odómetro	Horas motor	Costo	Notas	Fecha de crea...	Fecha de actualización	Actualizado
12345678	02121720	15/07/2014	03:06 a.m.	35	95	800	Unico incauca...	15/07/2014 03:0...	23/07/2014 01:35 a.m.	☒
12345678	2345677654	17/07/2014	01:44 a.m.	35	95	9675	Mantenimiento...	17/07/2014 01:4...		☐

Figura 10: Lista de historial de mantenimiento.

Ahora, en el tab de mantenimiento preventivo se creó una grilla llamada Mantenimiento preventivo en la que se muestran los mantenimientos que se pueden programar además de dar la opción de editar los campos de definición de usuarios dependiendo de los nuevos mantenimientos que se requieran.

La grilla posee las siguientes columnas:

- 2.3. Tipo de mantenimiento
- 2.4. Valor actual
- 2.5. Delta
- 2.6. Siguiete valor
- 2.7. Fecha de mantenimiento
- 2.8. Mantenimiento programado

The screenshot shows a web application window titled 'Módulo de mantenimiento'. At the top, there's a header with a truck icon and the word 'VEHICULOS'. Below the header, there are two tabs: 'Mantenimiento preventivo' (selected) and 'Historial de mantenimientos'. Under the 'Mantenimiento preventivo' tab, there's a form for vehicle information with fields for 'ID del vehículo' (IC050052) and 'Descripción del vehículo' (IC050052). Below this is a table of preventive maintenance records. The table has columns: 'Nombre del mantenimiento', 'Valor Actual', 'Fecha Actual', 'Delta', 'Siguiete valor', 'Siguiete fecha', 'Programado', and 'Editar'. There are three rows of data: 'Odómetro', 'Horas motor', and 'Llantas'. Each row has an 'Editar' link in the last column. A green box highlights the 'Editar' link for the 'Llantas' row, and a green arrow points from it to a modal dialog box titled 'Mantenimiento preventivo'. The dialog box has a checkbox for 'Formato fecha' and three input fields: 'Nombre mantenimiento' (Odómetro), 'Actual' (962), 'Delta' (5505), and 'Siguiete' (6467). There are 'Aceptar' and 'Cancelar' buttons at the bottom of the dialog. At the bottom right of the main window, there are two buttons: 'Adicionar' (with a plus icon) and 'Guardar' (with a checkmark icon).

Nombre del mantenimiento	Valor Actual	Fecha Actual	Delta	Siguiete valor	Siguiete fecha	Programado	Editar
Odómetro	962		5505	6467		☒	Editar
Horas motor	12		43	55		☒	Editar
Llantas		31/10/2014	100		08/02/2015	☒	Editar

Figura 11: Mantenimiento preventivo

En la figura 11, se muestran las columnas de la grilla que son editables. Al deseleccionar de los mantenimientos programados personalizados el CheckBox de mantenimiento, se elimina automáticamente.

Es importante aclarar que no utilizó ningún servicio web nuevo. Sólo se manipularon los registros.

Luego de tener listo el diseño, se procede a realizar las pruebas funcionales y de diseño en donde se revisa que no hallan excepciones pendientes por tratar.

Seguidamente, se realiza la traducción a los diferentes idiomas que maneja la compañía como lo son el Hebreo, Turco, e Inglés como se visualiza en la figura 12.

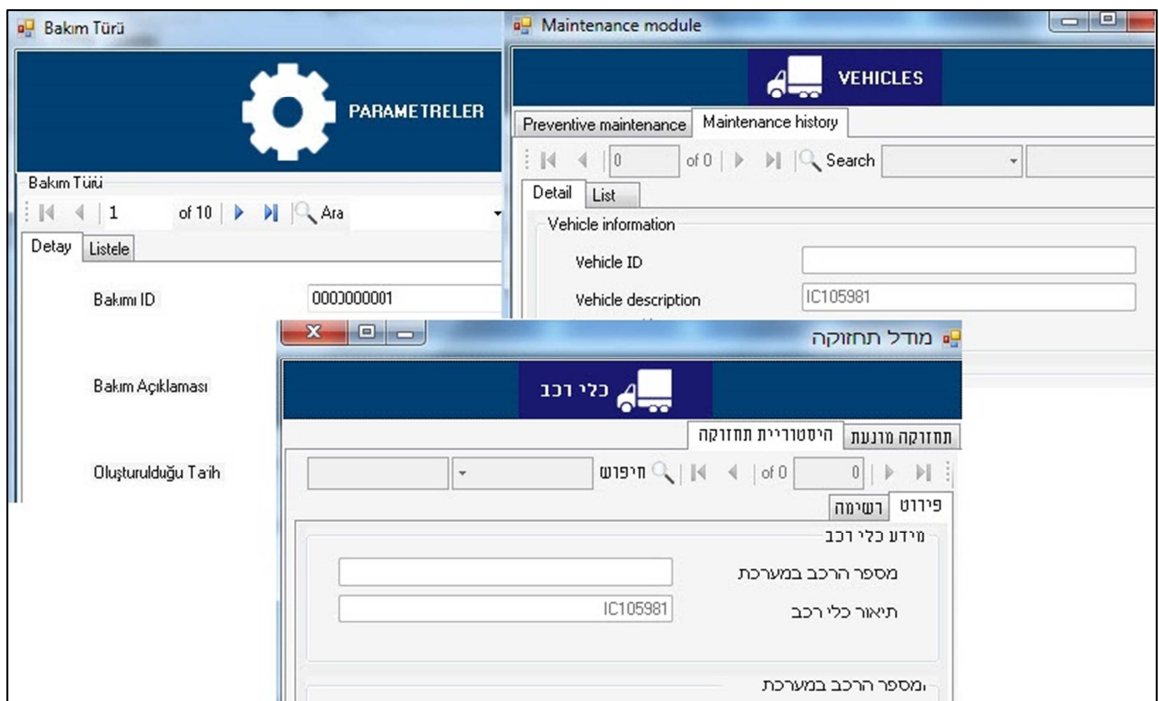


Figura 12: Formas en los idiomas de la compañía

En conclusión, se realizaron tres funcionalidades que se describieron con anterioridad. Una de ellas fue la realización de la forma de tipos de mantenimiento, en la que el usuario tiene la posibilidad de manipular adicionando, eliminando o modificando datos con respecto a los tipos de mantenimiento que ejerce normalmente la compañía a los distintos vehículos. El esquema se presenta en el

diagrama 2. A continuación una breve explicación de lo que se presenta en el diagrama.

- a. Obtención de datos: GetMaintenanceTypes es el servicio web que se ejecuta al correr el método Load de la forma de Windows. Ésta llena los campos con la data que retorna éste método.
- b. Muestrario de datos: Se puede evidenciar de dos formas, en el llenado de los campos o en la grilla de la forma de Windows
- c. Desea adicionar uno nuevo?: Es un botón de nuevo tipo de mantenimiento.
- d. Borrar campos: Al dar click sobre ese botón se borran los campos y se activa sólo el botón de guardar.
- e. Escribir campos: Se escriben los nuevos datos sobre los campos.
- f. Guardar: Al dar click, guarda y muestra una alerta indicando que ya se agregó con éxito el registro. Luego se ejecuta de nuevo el método de obtención de datos para refrescar la grilla y los campos.
- g. Modificar: Al dar click sobre los campos se pueden modificar.
- h. Guardar: Guarda las modificaciones ejecutando el servicio web ModifyMaintenanceTypes.
- i. Desea Eliminar?: Al dar click sobre el botón eliminar se elimina el registro, confirmando por una alerta emergente.
- j. Cerrar?: Al dar click en el botón de cerrar la forma, se cierra.

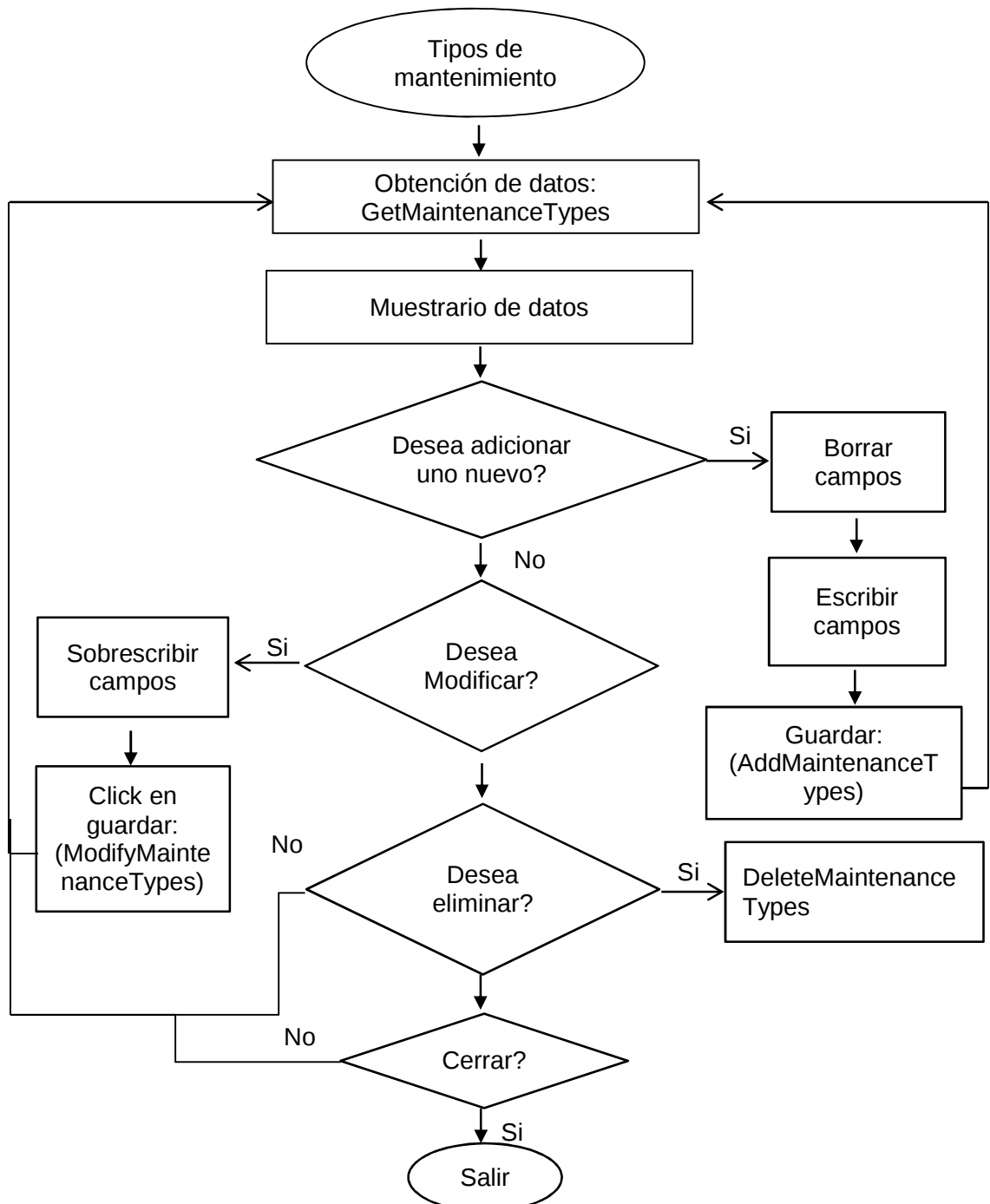


Figura 13. Tipos de mantenimiento

Ahora bien, la segunda funcionalidad consistió en la creación de un historial de mantenimiento, en donde el usuario tenga la posibilidad de administrar los servicios vehiculares. El diagrama 3, muestra a modo de resumen la funcionalidad. A continuación una breve explicación del contenido del diagrama.

- a. Obtención de datos: GetMaintenanceHistory es el servicio web que se ejecuta al correr el método Load de la forma de Windows. Ésta llena los campos con la data que retorna éste método.
- b. Muestrario de datos: Se puede evidenciar de dos formas, en el llenado de los campos o en la grilla de la forma de Windows
- c. Desea adicionar uno nuevo?: Botón indicador de adición de un mantenimiento.
- d. Borrar campos: Al dar click sobre ese botón se borran los campos y se activa sólo el botón de guardar.
- e. Escribir campos no inicializados: Se escriben los datos sobre los campos.
- f. Guardar: Al dar click, guarda al ejecutar el método de AddMaintenanceHistory y muestra una alerta indicando que ya se agregó con éxito el registro. Luego se ejecuta de nuevo el método de obtención de datos para refrescar la grilla y los campos.
- g. Modificar: Al dar click sobre los campos se pueden modificar.
- h. Guardar: Guarda las modificaciones ejecutando el servicio web ModifyMaintenanceHistory.
- i. Desea Eliminar?: Al dar click sobre el botón eliminar se elimina el registro, al ejecutar el servicio web DeleteMaintenanceHistory confirmando por una alerta emergente.
- j. Cerrar?: Al dar click en el botón de cerrar la forma, se cierra.

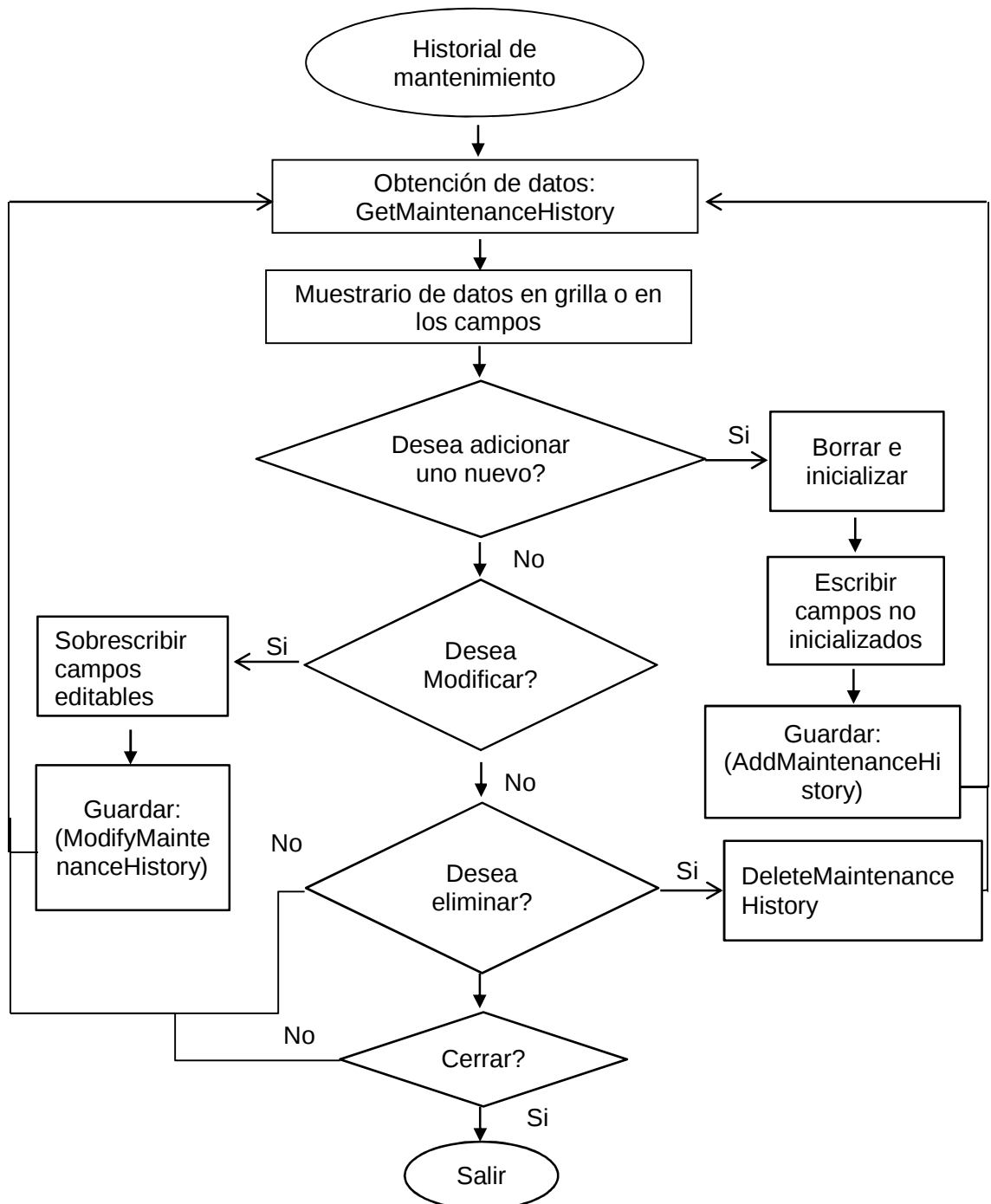


Figura 14. Historial de mantenimiento

Por último, el diagrama 4 muestra la funcionalidad del mantenimiento preventivo vehicular. En donde el usuario podrá programar los próximos servicios de sus flotas flexiblemente. A continuación una breve explicación del contenido del diagrama.

- a. Obtención de datos: GetMaintenancePreventive es el servicio web que se ejecuta cuando se corre el Load de la forma.
- b. Muestrario de datos: Son los registros traídos por el anterior servicio web que se muestran directamente sobre la grilla.
- c. Desea adicionar uno nuevo: Dando Click sobre la grilla directamente.
- d. Editar sobre grilla: Se sobrescriben los datos.
- e. Guardar: Bajo la ejecución del servicio web, ModifyMaintenancePreventive se guardan las modificaciones realizadas.
- f. Modificar?: Si se desea modificar, se edita sobre la grilla y se sobrescriben los campos.
- g. Guardar: Bajo el servicio ModifyMaintenancePreventive se modifica en la base de datos de acuerdo a los datos ingresados sobre la grilla.
- h. Desea eliminar?: Se ejecuta cuando se da click sobre el CheckBox de mantenimiento.
- i. Delete MaintenancePreventive: Es el método que elimina los datos restaurando a ceros la fila no seleccionada.
- j. Cerrar?: Es el botón de cerrar de la forma en la parte superior derecha que indica que desea salir.

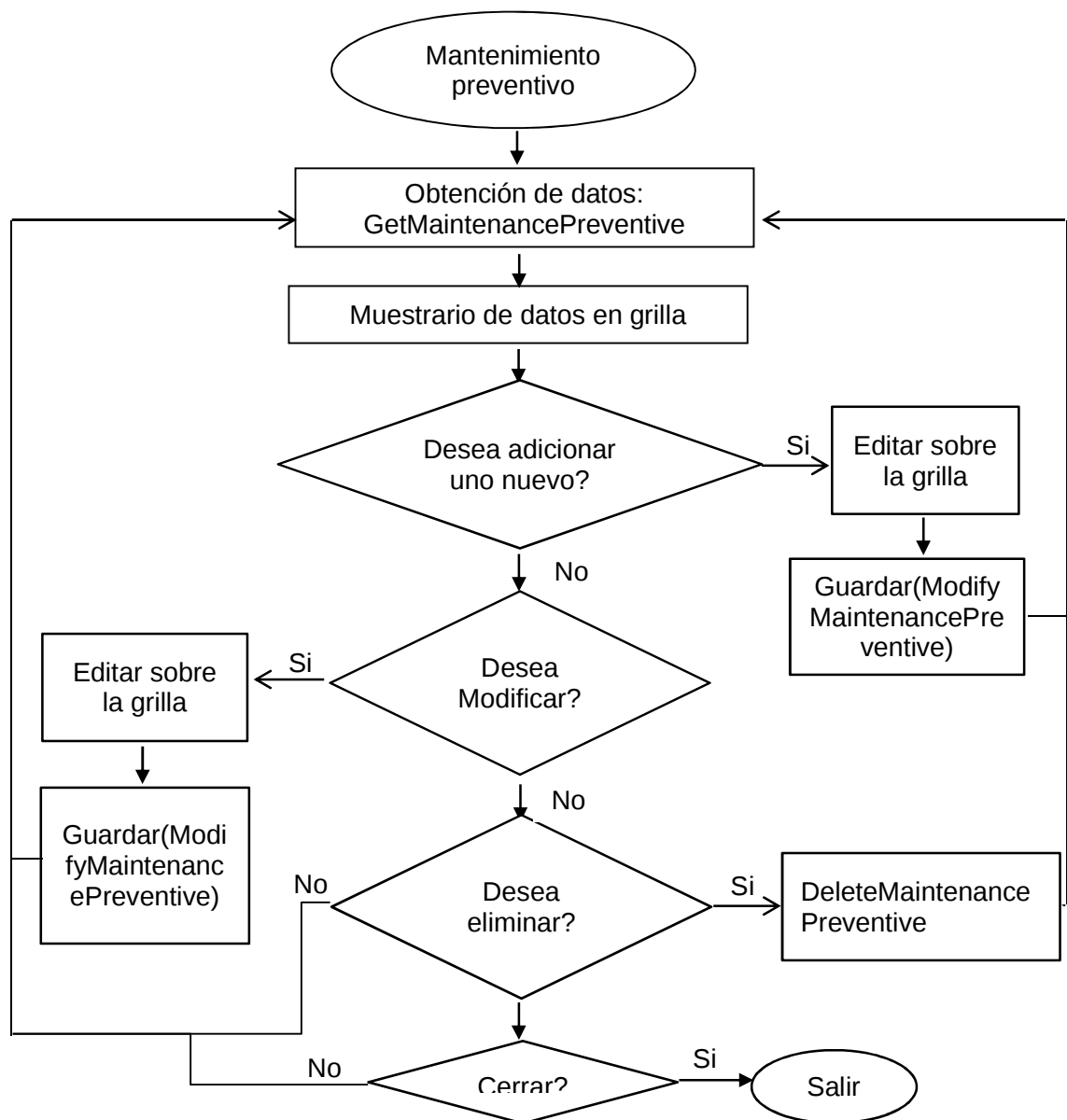


Figura 15. Mantenimiento preventivo.

Antes de finalizar con la entrega del proyecto, se realizaron pruebas al módulo de mantenimiento en las que se incluyen:

- a. Resolución de pantalla: La aplicación debe tener en todas sus formas una resolución mínima de pantalla de 1360x768.
- b. Pruebas de Anchor y Dock: Las formas al maximizar, deben quedar ordenadas. Estas propiedades en las formas de Windows son las encargadas de posicionar a los controles a la izquierda, derecha, arriba y/o abajo.
- c. Pruebas de diseño: Las formas deben quedar intuitivas y agradables al usuario.
- d. Pruebas de ortografía: Las formas deben tener tildes, buen uso de mayúsculas y signos de puntuación adecuados.
- e. Pruebas de idiomas: Las formas deben quedar totalmente traducidas a los idiomas que maneja la compañía como lo son: Español, Inglés, Hebreo y Turco.
- f. Prueba de RightToLeft: Las formas deben tener la posibilidad de leerse de derecha a izquierda para el caso del idioma Hebreo.
- g. Prueba de funcionalidad: Las formas deben tener todo tipo de mensajes de errores o de alertas en caso de requerirse. Además debe considerar excepciones al haber algún error en base de datos o servicios web.

En este caso, se manejó con detalle el módulo, arrojando sólo un error en los idiomas. Se tuvo que corregir la traducción al turco con ayuda de un traductor de Turquía.

Ya como última etapa, se hace la entrega formal de los scripts de las bases de datos (creación de tablas y adición de campos en la base de datos), los servicios

web creados o modificados. Las tablas que se entregaron en la base de datos fueron:

1. PreventiveMaintenanceType: Esta tabla administra los mantenimientos personalizados por el usuario.
2. PreventiveMaintenance: Esta tabla administra los mantenimientos programados personalizados.
3. MaintenanceTypes: Esta tabla administra los registros pertinentes a los tipos de mantenimiento de los vehículos.
4. MaintenanceHistory: esta tabla administra los registros del historial de mantenimiento del vehículo.

Los servicios web entregados fueron:

1. GetMaintenanceTypes: Obtiene los tipos de mantenimiento por compañía de la base de datos.
2. AddMaintenanceTypes: Agrega los tipos de mantenimiento por compañía a la base de datos.
3. ModifyMaintenanceTypes: Modifica los registros pertinentes a los tipos de mantenimiento de la base de datos.
4. DeleteMaintenanceTypes: Elimina los registros de la base de datos de la tabla de tipos de mantenimiento.
5. GetMaintenanceHistory: Obtiene los registros de la tabla MaintenanceHistory de la base de datos.
6. AddMaintenanceHistory: Agrega registros a la tabla MaintenanceHistory de la base de datos.
7. ModifyMaintenanceHistory: Modifica los registros pertinentes al historial de mantenimiento de la base de datos.

8. DeleteMaintenanceHistory: Elimina los registros de la base de datos de la tabla MaintenanceHistory.

Y finalmente, se sube versión a producción teniendo en cuenta la numeración de versionamiento estipulado por la empresa.

4.2 FUNDAMENTO HUMANISTA

La Facultad de Ingeniería Electrónica de la Universidad Santo Tomas, orienta su actividad formativa con una visión humano cristiana, inspirada en los planteamientos de Santo Tomás de Aquino. Formación que se dirige a cada uno de los componentes de la estructura integral del ser humano y que mediante acciones pedagógicas, investigativas y de conocimiento de la realidad social, propenden por la participación del egresado, en la creación de una sociedad más justa, con valores éticos arraigados, capaz de establecer una actitud crítica de sí misma y creativa en la búsqueda de soluciones; de tal forma, que cumpla con las exigencias del diario vivir al atender las más prominentes necesidades y problemáticas de la misma. Esta propuesta de la universidad la considero bastante acertada. Pues en Colombia, se han venido evidenciando ciertas posturas sociales, prácticas y pensamientos que van en contra de la moral humana y de la ética profesional. Los egresados Tomasinos hemos tenido la oportunidad, con la política curricular, de formarnos en la orientación humano – tomista, por lo tanto estamos en el deber de participar en nuestra sociedad con los principios de la estructuración de nuestro pensamiento, el “saber hacer” y “el saber obrar”.

De acuerdo a lo anterior, la fundamentación humanista del proyecto se funda desde un enfoque ambiental, ético y social.

En donde el enfoque ambiental, se sustenta tras realizar un adecuado mantenimiento a los vehículos, seguimiento y prevención de fallas. El mayor índice de contaminación atmosférica en el mundo está dado por unidades vehiculares que emiten gases tóxicos de forma descontrolada, que a lo largo del tiempo ocasionan variados tipos de patologías pulmonares, estrés, daños circulatorios, lluvia ácida, polución de aire y calentamiento global. El enfoque ético en el proyecto da lugar a la concienciación de gerentes de flota, quienes deben estar

atentos y prevenir de la mejor manera accidentes. Gracias al módulo de mantenimiento, se puede llevar un registro de los servicios realizados, permitiendo un seguimiento de las fallas que puede tener una unidad continuamente.

Y el enfoque social, se refiere a la calidad de vida de los seres humanos, porque se ha visto que la falta de control vehicular ha ocasionado múltiples desastres y accidentes que involucran muchas vidas. Ya sea por la irresponsabilidad de los gerentes o por el mal planteamiento de servicios. En Colombia el 8% de los accidentes de tránsito son causados por fallas mecánicas siendo éste un problema de salud pública. En consecuencia, el desarrollo del módulo, no sólo permite brindar seguridad a los conductores sino que contribuye a una armonía usuario-vehículo-vía previniendo daños personales y comunales.

V. CONCLUSIONES

Se crearon de forma exitosa las tablas en la base de datos, haciendo pruebas de inserción, modificación y eliminación de registros, El planteamiento de las tablas se basó en un estándar que lleva la empresa para hacerlo de una manera ordenada y de fácil acceso.

Se crearon los servicios web para la consulta, adición, modificación y eliminación de registros de la base de datos. Permitiendo así una interacción dinámica entre el software FleetSAP y la base de datos. De igual forma, se llevaron estándares en cuanto a los nombres y procesos de cada uno.

Se crearon los nuevos formularios de Windows con sus respectivos campos, ordenados por Groupbox y tabs. De igual forma, se agregaron las funcionalidades que acogen a los servicios web mediante los controles. De igual forma se comprobó con la base de datos que los registros estuviesen completos y con el formato adecuado.

Se adicionaron con éxito las hojas de recursos pertinentes a los idiomas manejados por la compañía como lo son Hebreo, Turco, Español e Inglés. Con el fin de mostrar al usuario el software con las etiquetas, botones y títulos en general con el idioma correspondiente.

Se integró cabalmente al proyecto FleetSAP el módulo de mantenimiento de flota, verificando el versionamiento y los protocolos de la compañía, los idiomas, el diseño y la funcionalidad.

El culminar el proyecto me llenó de satisfacción ya que se trabajó bastante duro en su implementación en la pasantía empresarial al eliminar de alguna manera las

malas costumbres de programación como lo son el nombrar variables aleatoriamente. Es por eso que se llevó a cabo un seguimiento de los estándares de programación de la compañía. Además de sobrepasar varias dificultades al carecer de todos los conceptos como lo son las propiedades, los métodos y como tal las clases. Sin embargo, me queda como reto personal poder desarrollar este mismo módulo como aplicación web. Es por eso, que decidí iniciar el reto de estudiar MVC e iniciar un nuevo proyecto consistente en desarrollar la plataforma FleetSAP de escritorio como aplicación web.

El desarrollo del módulo de mantenimiento vehicular me deja mucho conocimiento en cuanto a programación orientada a objetos, bases de datos y servicios web que al implementarse en conjunto se pueden convertir en múltiples aplicaciones desarrolladas en .Net de forma sencilla e intuitiva.

VI. VALOR AGREGADO Y TRABAJOS FUTUROS

Método que invierte los controles en la forma.

Se realizó un método en la clase cUtilities que pertenece al proyecto FleetSAP en donde se colocan de izquierda a derecha los controles de todos los formularios para el idioma Hebreo.

Estudios MVC

En algún futuro se piensa incorporar a FleetSAP como aplicación web ya que en la actualidad encontramos muchas herramientas útiles online y no se dependería de un programa instalado para poder controlar las flotas. Es así como se han venido realizando fuertes entrenamientos orientados a este tipo de programación basados en ejemplos y conceptos.

REFERENCIAS BIBLIOGRÁFICAS

1. Patrick. Tim. Programación con Visual Basic 2008. España: McGraw-Hill España, 2010. Páginas 782.
2. Joyanes Aguilar, Luis Sánchez García, Lucas. Programación en C++; un enfoque práctico. España: McGraw-Hill España, 2006. Páginas 418.
3. Gustavo Alonso. Fabio Casati. Web Services: Concepts, architectures and applications. Berlin [u.a.]: Springer, 2004. Páginas 354.
4. Joan Ribas Lequerica. Web Services. Madrid: Anaya Multimedia, 2003. Páginas 336
5. Durán Rodríguez, Luis. Bases de datos con Visual Basic. España: Marcombo, 2008. Páginas 305.
6. Joyanes Aguilar, Luis Zahonero Martínez, Ignacio. Programación en C: Metodología, algoritmos y estructuras de datos (2a. ed.). España: McGraw-Hill, 2005. Páginas 754.
7. Tapia Montes de Oca, Cristina. Aprenda ya SQL server 2005 técnicas aplicadas. México: McGraw-Hill Professional Publishing, 2010. Páginas 365.
8. Paul T. Kimmel. Professional DevExpress Asp.net Controls. John Wiley & Sons Inc., 2010. Páginas 672.
9. Organización Mundial de la Salud. <http://www.who.int>. [Online].; 2011 [Cited 2012 septiembre 19. Available. from:<http://www.who.int/mediacentre/factsheets/fs358/es/index.html>.

ANEXOS

1	[-] Maintenance Module	130 días	mar 14-01-21	lun 14-07-21
2	[-] Tipos de mantenimiento	53.5 días	mar 14-01-21	vie 14-04-04
3	[-] Bases de datos	42 días	mar 14-01-21	mié 14-03-19
4	Script de creación de la tabla MaintenanceTypes	8 días	mar 14-01-21	jue 14-01-30
5	Script del trigger MaintenanceTypes_Insert_Update_Delete	2 días	lun 14-03-17	mié 14-03-19
6	Restricciones Date created -Updated	2 días	mar 14-03-18	mié 14-03-19
7	Crear Primary Key tabla Maintenance type	0 días	vie 14-02-21	vie 14-02-21
8	[-] Web Services	26.5 días	vie 14-02-21	lun 14-03-31
9	GetMaintenanceTypes	8 días	vie 14-02-21	mar 14-03-04
10	AddMaintenanceTypes	8 días	mié 14-03-19	vie 14-03-28
11	ModifyMaintenanceTypes	8 días	mié 14-03-19	lun 14-03-31
12	DeleteMaintenanceTypes	2 días	jue 14-03-20	vie 14-03-21
13	[-] Form	15.5 días	vie 14-03-14	vie 14-04-04
14	[-] Diseño	11 días	jue 14-03-20	vie 14-04-04
15	Agregar forma fsys79MaintenanceTypes	11 días	jue 14-03-20	vie 14-04-04
16	Dock-Anchor-Minimizado-Maximizado-Size-Banner-	2 días	vie 14-03-21	lun 14-03-24
17	Crear grilla-Lista	2 días	vie 14-03-21	mar 14-03-25
18	Class Implementation	2 días	vie 14-03-21	mar 14-03-25
19	[-] Idioma	2 días	vie 14-03-14	lun 14-03-17
20	Configurar Idiomas	2 días	vie 14-03-14	lun 14-03-17
21	[-] Botones	2 días	lun 14-03-24	mié 14-03-26
22	Add	1 día	lun 14-03-24	mar 14-03-25
23	Delete	1 día	mar 14-03-25	mar 14-03-25
24	Save	1 día	mar 14-03-25	mié 14-03-26
25	Bindingsource	1 día	mar 14-03-25	mié 14-03-26
26	[-] Pruebas Tipos de mantenimiento	1.5 días	mié 14-03-26	jue 14-03-27
33	[-] Maintenance Module	92 días	vie 14-03-14	lun 14-07-21
34	[-] Form	20 días	vie 14-03-14	jue 14-04-10
35	Add Fsys80MaintenanceHistory	20 días	vie 14-03-14	jue 14-04-10
36	[-] Maintenance History	44 días	vie 14-03-14	mié 14-05-14
37	[-] Form	44 días	vie 14-03-14	mié 14-05-14
38	[-] Design	15 días	vie 14-03-14	jue 14-04-03
39	Add new tab with MaintenanceHistory information	15 días	vie 14-03-14	jue 14-04-03
40	Dock-Anchor-Minimizado-Maximizado-Size-Banner-	4 días	vie 14-03-14	mié 14-03-19
41	Add grid-List	2 días	vie 14-03-14	lun 14-03-17
42	Add Maintenance type button.	2 días	vie 14-03-14	lun 14-03-17
43	[-] Language	2 días	mar 14-05-06	mié 14-05-07
44	Traduccion a todos los idiomas	2 días	mar 14-05-06	mié 14-05-07
45	[-] Buttons	5.5 días	mié 14-05-07	mié 14-05-14
46	Add	0.5 días	mié 14-05-07	mié 14-05-07
47	Delete	0.5 días	mié 14-05-07	mié 14-05-07
48	Save	0.5 días	mié 14-05-14	mié 14-05-14
49	Bindingsource	0.5 días	mié 14-05-14	mié 14-05-14
50	[-] Test Maintenance history	2 días	vie 14-03-14	lun 14-03-17
51	Add	2 días	vie 14-03-14	lun 14-03-17
52	Modify	2 días	vie 14-03-14	lun 14-03-17
53	Delete	2 días	vie 14-03-14	lun 14-03-17
54	move Binding	2 días	vie 14-03-14	lun 14-03-17
55	Search criteria	2 días	vie 14-03-14	lun 14-03-17
56	Grid Data	2 días	vie 14-03-14	lun 14-03-17
57	[-] Preventive Maintenance	92 días	vie 14-03-14	lun 14-07-21
58	[-] WebService	2 días	vie 14-03-14	lun 14-03-17
59	ModifyCarsPreventiveMaintenance	2 días	vie 14-03-14	lun 14-03-17
60	[-] Form	5 días	vie 14-03-14	jue 14-03-20
61	Visualizar la informacion proveniente de vehiculos referente a Mantenimiento preventivo	4 días	vie 14-03-14	mié 14-03-19
62	Boton Save	5 días	vie 14-03-14	jue 14-03-20
63	Test Preventive Maintenance	2 días	vie 14-07-18	lun 14-07-21

Figura 14. Diagrama de Grantt