

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**SISTEMA PARA EL MEJORAMIENTO DE LAS ACTIVIDADES DE
LOGÍSTICA DE TRANSPORTE TERRESTRE UTILIZANDO ARQUITECTURA DE
SOFTWARE EMPRESARIAL EDRA PARA EL GAD MUNICIPAL DE MONTÚFAR**

Trabajo de grado previo a la obtención del título de Ingeniera en Sistemas
Computacionales.

Autora:

Flores López Joselyn Katherine

Director:

MSc. Xavier Mauricio Rea Peñafiel

Ibarra -2019



UNIVERSIDAD TÉCNICA DEL NORTE

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CEDULA DE IDENTIDAD		1003779244	
APELLIDOS Y NOMBRES		FLORES LÓPEZ JOSELYN KATEHRINE	
DIRECCIÓN		LA ESPERANZA BARRIO RUMIPAMBA	
EMAIL		kathysolp100@gmail.com	
TELÉFONO FIJO	062660039	TELÉFONO MOVIL	0989046885

DATOS DE LA OBRA	
TÍTULO	SISTEMA PARA EL MEJORAMIENTO DE LAS ACTIVIDADES DE LOGÍSTICA DE TRANSPORTE TERRESTRE UTILIZANDO ARQUITECTURA DE SOFTWARE EMPRESARIAL EDRA PARA EL GAD MUNICIPAL DE MONTÚFAR
AUTOR	JOSELYN KATHERINE FLORES LÓPEZ
FECHA	25 DE DICIEMBRE DE 2017
PROGRAMA	PREGRADO
TÍTULO	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	Msc, MAURICIO REA

2. CONSTANCIAS

La autora manifiesta que la obra objeto de la presente autorización es original y se la desarrollo, sin violar derechos del autor de terceros, por lo tanto la obra es original y que es la titular de los derechos patrimoniales, por lo que asumen la responsabilidad sobre el contenido de la misma y saldrán en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra a los 7 días de febrero del 2019

LA AUTORA:

A handwritten signature in blue ink, appearing to read "Joselyn Katherine Flores López", written over a horizontal dotted line.

JOSELYN KATHERINE FLORES LÓPEZ



UNIVERSIDAD TÉCNICA DEL NORTE

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE GRADO

Certifico que la Tesis previa a la obtención del título de Ingeniería en Sistemas Computacionales con el tema **“SISTEMA PARA EL MEJORAMIENTO DE LAS ACTIVIDADES DE LOGÍSTICA DE TRANSPORTE TERRESTRE UTILIZANDO ARQUITECTURA DE SOFTWARE EMPRESARIAL EDRA PARA EL GAD MUNICIPAL DE MONTÚFAR”** ha sido desarrollada y terminada en su totalidad por la señorita: Joselyn Katherine Flores López portadora de la cédula de identidad número: 1003779244 bajo mi supervisión para lo cual firmo en constancia.

Atentamente,

Msc. Mauricio Rea

DIRECTOR DE TESIS

AGRADECIMIENTO

Agradezco a Dios por poner en mi camino una familia maravillosa y a personas muy valiosas que me han dado mucha felicidad.

A mi familia que en toda mi carrera estuvieron pendientes de mí, dándome un apoyo invaluable siempre.

A Santiago Chauca por ser un gran amigo, sobre todo un gran apoyo y compartir conmigo muchos momentos inolvidables en el aula de clases.

A la Universidad Técnica del Norte a la Carrera de Ingeniería en Sistemas Computacionales y a todos mis profesores forjadores de mi carrera profesional.

Al Msc. Diego Trejo amigo y profesor, gracias por todo lo que me enseñó, por las oportunidades que me brindó y por los consejos siempre sinceros que hicieron que crezca mucho como profesional.

A mi director de Tesis Msc. Mauricio Rea por ser un excelente profesor y amigo habiéndome guiado a lo largo del desarrollo de este trabajo

Al Msc. Marco Revelo e Ing. Marco Sánchez quienes aportaron con sus conocimientos para el desarrollo de un sistema de calidad que cumplió con su objetivo.

A mi auspiciante el GAD municipal de Montúfar que me dio todas las facilidades necesarias para cumplir con este trabajo.

.Joselyn Katherine Flores López

DEDICATORIA

Este trabajo está dedicado a mi familia que son las personas más importantes en mi vida las cuales me dieron su apoyo incondicional a lo largo de toda mi carrera.

A **Nicolás Flores** y **Rosa López** los mejores padres del mundo quienes me brindaron su amor, respeto y apoyo siempre, que con su ejemplo y esfuerzo han hecho de su hija una gran persona y excelente profesional.

A mi hermano **Fernando Flores** por brindarme su compañía y cariño siempre que lo necesite.

A mis abuelitos, tíos, primos y sobrina quienes siempre han estado cerca de mí, ayudándome en todo lo que he requerido e incentivándome para salir adelante.

Joselyn Katherine Flores López

Tabla de Contenido

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR.....	II
CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE GRADO.....	IV
AGRADECIMIENTO	V
DEDICATORIA	VI
Tabla de Contenido.....	VII
Índice De Figuras.....	X
Índice De Cuadros	XII
RESUMEN	XIV
ABSTRACT.....	XV
INTRODUCCIÓN.....	1
a. Identificación de la problemática.....	1
i. Antecedentes	1
ii. Situación Actual.....	2
iii. Planteamiento del Problema	2
i. Prospectiva.....	2
b. Objetivos	3
i. Objetivo General.....	3
ii. Objetivos específicos	3
c. Alcance	3
d. Justificación	5
i. Justificación del Proyecto	5
ii. Justificación de la Arquitectura Metodología de desarrollo	5
iii. Justificación de las Herramientas.....	6
CAPÍTULO 1.....	9
Marco teórico.....	9
1.1. Sistemas de Información Web.....	9
1.1.1. Sistemas de Información.....	9
1.1.2. Sistemas Web.....	11
1.1.3. Sistemas de Logística.....	13
1.2. Arquitectura EDRA	14

1.2.1.	MBI 3.O & EDRA	16
1.2.2.	Servidor Web IIS	17
1.2.3.	Plataforma Microsoft.NET	18
1.2.4.	Lenguaje de Programación C#.....	20
1.2.5.	Base de Datos SQL Server.....	22
1.3.	Metodología SCRUM.....	23
1.3.1.	Roles	25
1.3.2.	Componentes.....	26
1.3.3.	Fases del Proyecto.....	29
1.3.4.	Ventajas.....	31
1.3.5.	Desventajas	31
•	Framework Bootstrap.....	31
•	SpringToMeter.....	32
	CAPÍTULO 2.....	33
	Desarrollo.....	33
2.1.	Preparación del proyecto	33
2.1.1.	Dimensión del Proyecto	33
2.1.2.	Roles del Sistema.....	34
2.1.3.	Pila de Producto	34
2.1.4.	Historias de Usuario.....	35
2.1.5.	Pila de Tareas	35
2.2.	Planificación del Sprint	38
2.2.1.	Sprints	39
2.3.	Desarrollo de los Sprint	43
2.3.1.	Sprint 1 Seguridad y Administración General	43
2.3.2.	Sprint 2 Equipos y Planes de Mantenimiento	47
2.3.3.	Sprint 3 Rutas y Kilometrajes	49

2.3.4.	Sprint 4 Mantenimientos no Planificados y Revisiones Vehiculares	51
2.3.5.	Sprint 5 Notificaciones Alertas y Reportes.....	54
	CAPÍTULO 3.....	57
	Resultados.....	57
3.1.	Resultados Obtenidos	57
3.2.	Análisis de Impactos.....	58
3.2.1.	Impacto Tecnológico	58
3.2.2.	Impacto Económico	58
3.2.3.	Impacto Ambiental.....	59
	CONCLUSIONES	61
	RECOMENDACIONES.....	62
	REFERENCIAS.....	63
	ANEXOS	67
	ANEXO A: Historias de Usuario.....	68
	ANEXO B: Interfaces del Sistema	75
	ANEXO C: CERTIFICADO	77

Índice De Figuras

Figura 1. Módulos del Sistema.....	4
Figura 2. Partes de los Sistemas de Información	10
Figura 3, Evolución de la Web	12
Figura 4. Administración de la logística.....	14
Figura 5. Modelo Conceptual EDRA.....	15
Figura 6. Evolución de MBI	16
Figura 7. Características de IIS	18
Figura 8. Ejecución de un programa en .NET	19
Figura 9. Partes de un programa.....	20
Figura 10. Estructura de un Sprint	24
Figura 11. Ciclo de la metodología SCRUM.....	25
Figura 12. Historia de Usuario	27
Figura 13. Pila de Producto	28
Figura 14. Lista de Tareas.....	28
Figura 15. Diagrama Detallado de las fases de Scrum	30
Figura 16. Planificación del Sprint.....	39
Figura 17. Estructura del Sprint	39
Figura 18. SPRINT 1: Seguridad y Administración General	40
Figura 19. SPRINT 2: Equipos y Planes de Mantenimiento	41
Figura 20. SPRINT 3: Rutas y Kilometrajes	41
Figura 21. SPRINT 4: No Planificados y Revisiones.....	42
Figura 22. SPRINT 5: Notificaciones, Alertas y Reportes	42
Figura 23. Modelo de Datos	44
Figura 24. Pantalla Login	45
Figura 25. Administración General del Sistema.....	46

Figura 26. Administración de Parametros.....	46
Figura 27. Administracion de Equipos.....	48
Figura 28. Plantillas y Asignación de Planes de Mantenimiento.....	48
Figura 29. Asignacion de Rutas.....	50
Figura 30. Administracion de Repuestos.....	50
Figura 31. Asignacion de Accidentes.....	52
Figura 32. Asignacion de Revisiones.....	53
Figura 33. Administración de Evidencias.....	53
Figura 34. Aprobación de peticiones.....	55
Figura 35. Actualizacion Clave.....	75
Figura 36. Actualizacion Clave.....	75
Figura 37. Perfil de Usuario.....	76
Figura 38. Administración de Usuarios.....	76

Índice De Cuadros

Tabla 1. Roles de Scrum.....	25
Tabla 2. Roles del Equipo de Trabajo.....	34
Tabla 3. Pila de Producto.....	34
Tabla 4. Pila de Tareas H1: Plataforma Tecnológica.....	35
Tabla 5. Pila de Tareas H2: Seguridad: Usuarios y Roles	35
Tabla 6. Pila de Tareas H3: Administración General del Sistema	36
Tabla 7. Pila de Tareas H3: Administración de Parámetros	36
Tabla 8. Pila de Tareas H3: Gestión de Equipos.....	36
Tabla 9. Pila de Tareas H6: Gestión de Planes de Mantenimiento	37
Tabla 10. PILA DE TAREAS H7: Gestión de Mantenimientos no Planificados	37
Tabla 11. PILA DE TAREAS H8: Gestión de Revisiones Vehiculares.....	37
Tabla 12. PILA DE TAREAS H9: Gestión de Rutas y Kilometrajes.....	37
Tabla 13. PILA DE TAREAS H10: Aprobación de Peticiones	38
Tabla 14. PILA DE TAREAS H11: Inventario de Herramientas	38
Tabla 15. PILA DE TAREAS H12: Notificaciones y Reportes	38
Tabla 16. Roles o Tipos de Usuario.....	43
Tabla 17. Historia de Usuario: Aprobación de Rutas	54
Tabla 18. Historia de Usuario: Plataforma Tecnológica	68
Tabla 19. Historia de Usuario: Seguridad Usuarios y Roles.....	68
Tabla 20. Historia de Usuario: Administración General del Sistema.....	69
Tabla 21 . Historia de Usuario: Administración de Parámetros.....	69
Tabla 22. Historia de Usuario: Gestión de Equipos.....	70
Tabla 23. Historia de Usuario: Gestión de Planes de Mantenimiento.....	71
Tabla 24. Historia de Usuario: Gestión de Mantenimientos no Planificados.....	71
Tabla 25. Historia de Usuario: Gestión de Revisiones Vehiculares	72

Tabla 26. Historia de Usuario: Rutas y Kilometrajes	72
Tabla 27. Historia de Usuario: Aprobación de Peticiones	73
Tabla 28. Historia de Usuario: Inventario de Herramientas	73
Tabla 29. Historia de Usuario: Notificaciones y Reportes	74

RESUMEN

El presente proyecto se fundamenta en el “SISTEMA PARA EL MEJORAMIENTO DE LAS ACTIVIDADES DE LOGÍSTICA DE TRANSPORTE TERRESTRE PARA EL GAD MUNICIPAL DE MONTÚFAR”, utilizando la arquitectura de software empresarial EDRA y la metodología de desarrollo ágil SCRUM.

El documento consta de una introducción y tres capítulos en el que desglosan de manera clara, paso a paso el desarrollo de las diferentes etapas hasta la culminación del proyecto.

En la Introducción se describe Antecedentes, Situación Actual, Objetivo General, Objetivos Específicos, la Justificación del Proyecto, las Herramientas y Metodología de estudio que se utilizara para desarrollar el sistema.

En el Capítulo 1 se detalla de manera teórica todas las herramientas y conceptos necesarios para el desarrollo como: Sistemas de Información web, Sistemas de logística, Arquitectura EDRA, Plataforma Microsoft .NET, Lenguaje de Programación C#, Base de Datos SQL Server, Metodología SCRUM, Framework Bootstrap y SpringToMeter.

En el Capítulo 2 se desarrolla la construcción del sistema, se describe los roles del sistema, pila de productos, pila de tareas, planificación del proyecto y las iteraciones. Finalmente en el Capítulo 3 se presentan las conclusiones, recomendaciones obtenidas en la realización del Proyecto de Titulación además el impacto que generará al utilizar la aplicación. Cabe destacar que este trabajo se lo realizó en conjunto con la carrera de Ingeniería Automotriz, la construcción y desarrollo de SISMANCAR se desarrolla en esta tesis mientras que el proceso de implementación y constatación de resultados el cual es posterior al desarrollo, se encuentra documentado en la tesis “IMPLEMENTACIÓN DE UN SISTEMA DE MANTENIMIENTO TOTAL PRODUCTIVO EN EL GAD MUNICIPAL DE SAN GABRIEL” del autor Marco Sánchez.

Palabras Claves: Sistema, web, EDRA, Scrum, mantenimiento, logística, transporte, mejoramiento, desarrollo, ágil, parque automotor, desarrollo.

ABSTRACT

This project is based on "SYSTEM FOR THE IMPROVEMENT OF LAND TRANSPORTATION LOGISTICS ACTIVITIES FOR THE MUNICIPAL GAD OF MONTÚFAR", using the business software architecture EDRA and the production of agile development SCRUM.

The document consists of an introduction and three chapters in which the development of the different stages up to the completion of the project is broken down in a clear manner, step by step.

The Introduction describes Background, Current Situation, General Objective, Specific Objectives, and Justification of the Project, Tools and Study Methodology that is used to develop the system.

In Chapter 1 the theoretical way is detailed, all the tools and concepts are used for the development as: Web information systems, Logistics Systems, EDRA Architecture, Microsoft .NET Platform, C # Programming Language, SQL Server Database , SCRUM Methodology, Bootstrap Framework and SpringToMeter.

In Chapter 2 the construction of the system is developed, the system roles, product stack, task stack, project planning and iterations are described. Finally, Chapter 3 presents the conclusions, the recommendations in the completion of the Titling Project, as well as the impact it will generate when using the application. It should be noted that this work is what is done in conjunction with the career of Automotive Engineering, the construction and development of SISMANCAR is developed in this thesis while the process of implementation and verifies results which is after the development, is documented in the thesis "IMPLEMENTATION OF A TOTAL PRODUCTIVE MAINTENANCE SYSTEM IN THE MUNICIPAL GAD OF SAN GABRIEL" by the author Marco Sánchez.

Keywords: System, web, EDRA, Scrum, maintenance, logistics, transport, improvement, development, Agile, automotive park, development.

INTRODUCCIÓN

a. Identificación de la problemática

i. Antecedentes

A lo largo del tiempo la información que se maneja en las empresas y en todos los escenarios de la sociedad se ha convertido en el eje promotor de cambios sociales, económicos y culturales que ha crecido exponencialmente haciendo que su manejo, administración y depuración sea cada vez más complejo, para solventar este problema y gracias al “auge de las telecomunicaciones se ha producido una transformación de las tecnologías de la información y de la comunicación, cuyo impacto ha afectado a todos los sectores de la economía y la sociedad” (Estrada, 2014, pág. 18).

En la década de 1980 el ritmo de la innovación en informática y comunicaciones de hardware y software se aceleró. Los microordenadores¹ tuvieron un gran impacto en la logística. Uno de los métodos que se recomienda es la administración por objetivos (APO) que se define como: “un sistema administrativo integral en el que se combinan en forma sistemática muchas actividades administrativas básicas y que persigue deliberadamente el cumplimiento eficaz y eficiente de los objetivos organizacionales” (Baca, Acosta, & Solares, 2014, pág. 2).

La adopción de un software de gestión de logística es un claro ejemplo de un método de administración por objetivos.

El impacto que genera la adopción de un sistema web en la gestión de la logística ha hecho que la adquisición de dicho software se convierta en una necesidad en la mayoría por no mencionar todas las empresas ya sean privadas o públicas, ya que representa recursos que permiten «la interacción directa entre los usuarios facilitando la construcción colectiva de contenidos, intercambiando información de valor o simplemente vinculando intereses o proyectos comunes. (Gallego, Álvarez, & Rosanigo, 2015, pág. 43)

El GAD² Municipal del Cantón Montúfar de la Ciudad de San Gabriel, Provincia del Carchi fundada el 27 de septiembre de 1905, se ve en la necesidad de adoptar un sistema web capaz de gestionar la logística de los vehículos que utiliza, ya que en el Ecuador

¹ Computadoras que ocupan espacios físicos pequeños, comparadas a sus predecesoras que utiliza un microprocesador como unidad central de procesamiento.

² Gobierno Autónomo Descentralizado: entidades que pertenecen al sector público no financiero del país.

especialmente en el sector público se evidencia la falta de un software que brinde las facilidades para llevar a cabo la organización y cumplimiento de objetivos en el sector automotriz automotor específicamente.

ii. Situación Actual

No se cuenta con la tecnología para la gestión de la logística que reúna procesos tanto de administración de procesos y de planificación y seguimiento de mantenimientos vehiculares en el parque automotor del GAD.

Las actividades se gestionan a través de mecanismos obsoletos e ineficaces. Proceso que se vienen realizando a través de registros en hojas electrónicas de Excel e incluso registros a mano, un ejemplo claro el proceso tedioso e ineficiente que se realiza siempre que la persona encargada de los vehículos frecuentemente tenga que actualizar los kilometrajes a mano, parando las actividades de todos los vehículos, registrando uno por uno cada kilometraje y luego actualizando esta información en una hoja de Excel. Estos métodos ocasionan problemas como: la inexistencia de planificación y seguimiento de procesos, desorganización y falta de control en la asignación de actividades al personal.

También se observa una inadecuada programación y realización de los mantenimientos vehiculares, esto se produce debido a que se debe revisar más de 30 planes de mantenimiento diferentes por cada vehículo y verificar que debe hacerse, aunque la mayoría de veces se realizan lo mantenimientos preventivos básicos, mientras que otros se descuidan y en vez de prevenir posibles daños, se producen, y se debe realizar reparaciones, por lo que representa un desgaste en las piezas y partes, generando el paro de vehículos o maquinarias, así como pérdidas económicas debido a gastos injustificados en los mantenimientos y en el combustible.

iii. Planteamiento del Problema

¿Cómo lograr el mejoramiento de las actividades de logística de transporte terrestre utilizando arquitectura de software empresarial EDRA³ para el GAD Municipal de Montúfar?

i. Prospectiva

Se propone un sistema web para la automatización y el mejoramiento de las actividades, donde dicha propuesta se convierte en una herramienta de seguimiento y control de todos los recursos del parque automotor del GAD Municipal de Montúfar.

³Arquitectura de referencia de desarrollo empresarial (Enterprise Development Reference Architecture) o también conocida como Shadowfax.

Se podrá verificar y justificar los gastos realizados así como llevar un seguimiento y aplicar medidas correctivas que aportarán a la toma de decisiones, en el departamento. Así se evitará el incumplimiento de los procesos y la ineficiencia en la ejecución de los mismos, también la pérdida de tiempo de trabajo, las averías fortuitas y disminuirá el riesgo de que un vehículo sufra una avería grave.

b. Objetivos

i. Objetivo General

Desarrollar un sistema para el mejoramiento de las actividades de logística de transporte terrestre utilizando arquitectura de software empresarial EDRA para el GAD Municipal de Montúfar.

ii. Objetivos específicos

1. Investigar y recolectar información pertinente a todos los actores y elementos involucrados en los procesos de logística de transporte GAD Municipal de Montúfar.
2. Investigar la arquitectura de software empresarial EDRA
3. Realizar las pruebas de funcionamiento de la solución informática.
4. Utilizar la metodología de desarrollo SCRUM⁴ para asegurar niveles de calidad en el producto

c. Alcance

Se desarrollará un Sistema Web con 4 Roles de Usuario; Secretaria, Técnico, Supervisor, Administrador y 5 Módulos.

En la Figura 1 se detallan los módulos que forman el sistema y la funcionalidad que cada uno tiene, en total son 8 módulos, la construcción del sistema será en 2 fases bien definidas; la primera en la cual se debe completar al 100% los 6 primeros módulos ya que estos son necesarios para implementar en el GAD y llenar datos necesarios para proceder a la segunda fase, la cual tiene que ver con el procesamiento de la información en el módulo de kilometrajes y rutas para al final generar la funcionalidad de reportes.

⁴ Forma de trabajo de la metodología Ágil que consistente en prácticas colaborativas que minimizan riesgos en la elaboración de un proyecto.



Figura 1. Módulos del Sistema

Fuente. Propia

d. Justificación

i. Justificación del Proyecto

La implementación del programa para el mejoramiento de las actividades de logística de transporte terrestre, es de gran importancia para el GAD Municipal de Montúfar, ya que permitirá tener en óptimas condiciones el buen funcionamiento a los vehículos livianos y maquinaria pesada, mediante parámetros de organización, planificación y ejecución programada, los mismos que serán utilizados en los trabajos de mantenimiento y viabilidad de calles de la ciudad, parroquias, barrios y comunidades.

De esta manera se quiere sustituir la documentación impresa por la digital para resolver las pérdidas de tiempo y dinero, tener acceso a una información veraz, ordenada, clasificada y actualizada, por lo tanto, el sistema de mantenimiento aportará con cambios positivos para la institución.

Este sistema mejorará el desempeño de todo el personal que labora en la entidad pública, informará sobre los trabajos de mantenimientos preventivos, permitirá evitar fallas antes de que ocurran, se podrá documentar toda la información de las rutinas de mantenimiento de los equipos.

ii. Justificación de la Arquitectura Metodología de desarrollo

“La arquitectura de software se hace necesaria cuando el tamaño y complejidad de los sistemas crecen” (Arias & Durango, 2016, pág. 202), por ello EDRA tiene como principal característica ser empresarial, lo que permite estandarizar y resolver los problemas que se presentan en el desarrollo de sistemas distribuidos, también se basa en principios apoyados en una fuerte feedback⁵ de la comunidad y la industria. Otro motivo por el cual se escogió este marco de trabajo es porque en el GAD piensan a futuro implementar un ERP⁶ en su entidad por lo que se hace necesario la creación de servicios web, para ser consumidos por futuras soluciones de software.

“SCRUM permite el desarrollo de software ágil es una metodología que tiene como objetivos fundamentales: planes de adaptación, respuesta rápida y flexible al cambio, desarrollo evolutivo y entrega rápida” (Blokehead, 2016). Estos objetivos permiten generar más y mejor software en menor tiempo, ya que se prioriza módulos, funcionalidades y

⁵ Obtener mucha información y sugerencias sobre los resultados o desempeño de un proceso o artículo.

⁶ Enterprise Resource Planning: programas se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos.

características que agregan más valor al producto final y a diferencia de metodologías de desarrollo tradicionales esta fomenta la transparencia, la constante integración y permite a todos los colaboradores del equipo y usuarios finales estar siempre al tanto de los avances, objetivos y tiempos del proyecto.

iii. Justificación de las Herramientas

La mayoría de herramientas que se utiliza para la implementación de este proyecto son tecnologías de Microsoft ya que la arquitectura EDRA incorpora como recursos a la plataforma ASP.NET⁷ y el framework .NET; que permiten maximizar la productividad en el desarrollo y brinda una seguridad bastante alta para el sistema.

“La plataforma .NET apareció para dar respuesta al problema de complejidad de las APIs⁸. Las unifica y las moderniza, permitiendo su uso con cualquier lenguaje que posea una implementación compatible” (Putier, 2015, pág. 17), no tienen ningún valor ya que se utilizan versiones del framework .NET Community disponible en el link:

<https://www.visualstudio.com/vs/community/>

También se utiliza la versión de SQL Server 2014 Express disponible en el link: <https://www.microsoft.com/es-es/download/details.aspx?id=42299>

“El lenguaje de programación C# orientado a objetos simple, moderno y de propósito general y tiene la capacidad para desarrollar componentes de software que se puedan usar en ambientes distribuidos” (Guerin, 2016, pág. 88).

“El servidor de base de datos SQL Server contiene un conjunto completo de tecnologías y herramientas que ayudan a obtener el máximo rendimiento a la información” (Medina, 2015, pág. 8). Actualmente es uno de los más utilizados en el mercado debido a su motor de base de datos relacional.

Internet Information Server (IIS) “es el servidor de páginas web avanzado que posee certificados centralizadas, restricciones dinámicas por IP⁹, restricción de acceso incorrectos por FTP¹⁰, escalabilidad NUMA¹¹, etc.” (Villada, 2014).

⁷ Es un modelo de desarrollo Web unificado creado por Microsoft para el desarrollo de sitios y aplicaciones web dinámicas con un mínimo de código.

⁸ Interfaces de programación de aplicaciones (Application Programming Interface).

⁹ Protocolo de Internet: un estándar que se emplea para el envío y recepción de información mediante una red.

¹⁰ Protocolo de Transferencia de Ficheros para transferir grandes bloques de datos por la red

¹¹ Acceso a memoria no uniforme (Non-Uniform Memory Access): un diseño de memoria utilizado en multiprocesamiento.

“Bootstrap es un enfoque de diseño web que proporciona una visualización óptima para una experiencia de navegación fácil y con un mínimo de cambio de tamaño, paneo y desplazamiento a través de una amplia gama de dispositivos” (Miguel, 2014, pág. 1)

CAPÍTULO 1

Marco teórico

1.1. Sistemas de Información Web

Los Sistemas de Información Web son indispensables para el funcionamiento óptimo de una empresa ya que se consideran como la mejor solución a muchos errores e interrupciones en el flujo de procesos, causados al ser realizado en forma manual.

1.1.1. Sistemas de Información

Su definición es la siguiente: “es una infraestructura iterativa de personas y procedimientos que hacen que la información relevante dentro de una organización esté disponible para planear, controlar e implementar con más facilidad cualquier tipo de innovación” (Baca, 2015, pág. 23). Y esto hace que la entrega del producto o servicio sea muy riguroso en cuanto a calidad, tiempo de entrega, disposición, entre otros. Valores agregados que crean lealtad del cliente.

Desde el punto de vista de su funcionamiento, un sistema de información realiza cuatro operaciones básicas: recepción de datos, almacenamiento, procesamiento y salida o entrega de información. Desde el punto de vista de sus objetivos, son tres los principales: automatiza el flujo de datos en cualquier proceso, proporciona información que sirve de base para tomar mejores decisiones y siempre proporciona mayor valor y ventajas competitivas a la empresa. (Baca, 2015, pág. 24)

Estos sistemas generalmente logran ser implementados a través de diferentes proyectos creados por los departamentos, principalmente por el de TI¹². Esta tarea sigue lineamientos y metodologías especificadas, no se puede simplemente escoger el área donde surgió el problema ya que primero el director de TI o encargado, debe identificar el área en el que existe un problema asesorándose del CEO¹³ o persona con mayor jerarquía. La siguiente etapa consiste en seleccionar el área donde se presenta el problema en el flujo de información, hacer un estudio técnico y el rediseño del proceso, en esta etapa por lo regular es necesaria la adquisición de nuevo TI y capacitaciones a los participantes, después se realiza el análisis económico, es decir cuánto costara la inversión que se va a realizar, los beneficios tangibles e intangibles, flujos netos de efectivo incrementales para

¹² Tecnología de la información: puede ser un departamento o equipos.

¹³ (Chief Executive Officer) persona con más alta responsabilidad de una organización o corporación.

finalmente pasar a la última etapa que es la evaluación económica, el análisis de riesgo y las conclusiones generales.

En la Figura 2 se detallan las partes que conforman la metodología de un proyecto en sistemas de información.



Figura 2. Partes de los Sistemas de Información

Fuente. (Baca, 2015, pág. 17)

Los datos manejados en la administración y gestión del parque automotor del GAD de Montufar es información bastante extensa y delicada que al paso y con mecanismos obsoletos hace imposible la administración de esta. La entidad al ser pública y manejar presupuestos del gobierno debe asegurar la calidad en cuanto al servicio que brindan y un sistema de información logra mayor valor al servicio.

El tipo de sistema de información empleado por una empresa determina la eficiencia y la competitividad en el mercado. Un manejo óptimo de la información hace que una empresa pueda disminuir costos de distribución del producto, lo que por esta sola acción hace más eficiente a la empresa en términos de costos, y si además el sistema de información permite a la empresa entregar el producto en el lugar adecuado y en cantidades demandadas por el mercado harán más competitiva a la empresa. (Baca, 2015, pág. 25)

“Por lo general, los proyectos en sistemas de información requieren de la adquisición de TI. En una estrategia de se define cual información, cual sistema de información y que arquitectura se refiere para apoyar a una empresa en la obtención de los mencionados beneficios tangibles e intangibles” (Baca, 2015, pág. 14).

Se construye un software a medida para el GAD ya que proporciona muchas facilidades al momento de la construcción y la implementación. Se escoge un sistema web porque es una solución más eficaz que se obtiene en el menor tiempo posible, también porque al utilizar software libre no significa ningún costo para la entidad, por otro la implementación es sencilla ya que se tiene a disposición toda la tecnología y recursos necesarios.

1.1.2. Sistemas Web

Aquel tipo de software diseñado como una herramienta que permita al usuario realizar una tarea específica, compilado o interpretado escrito en algún lenguaje de programación que necesita de un sistema operativo soporte para poder funcionar y requiere unas condiciones determinadas para su instalación y ejecución, y que hacen referencia a espacio en disco, procesador y memoria RAM¹⁴. (Moreno & González, 2014, pág. 64)

La condición primordial para su funcionamiento son las comunicaciones en red o también conocida como la web, que consiste “en una forma de organizar la información usando como medio físico de comunicación la red internet y el protocolo HTTP¹⁵” (Ramos & María, 2014, pág. 3) por consiguiente es utilizado para realizar peticiones a los servidores web y recibir las respuestas por medio de un navegador como el caso de Chrome. La WWW permite de manera sencilla, visualizar información alojada en ordenadores remotos a través de un conjunto de páginas web.

(Moreno & González, 2014) Afirma: los programas que sirven para la conexión a Internet o que usan servicios de esta red son:

- a) **Navegadores:** Se usan para acceder a multitud de servicios de Internet a través del servicio Word Wide Web o www, mostrando contenidos en HTML¹⁶ u otros lenguajes. Destacan Internet Explorer, Firefox, Opera.
- b) **Cientes de correo electrónico:** Empleados para recibir y enviar correos electrónicos a través de un servidor de correo. Destacan MS Outlook Express, Eudora, Thunderbird.
- c) **Programas de chat o Mensajería Instantánea:** Sirven para tener conversaciones en tiempo real con los contactos registrados. Destacan MNS Messenger, Mirc.
- d) **Reproductores multimedia de Internet:** Permiten ver contenidos multimedia transmitidos por Internet. Destacan Real Player, Windows Media Player, Flash Player.

¹⁴ Memoria de acceso aleatorio (Random Access Memory): es donde el computador guarda los datos que está utilizando en el momento presente.

¹⁵ Protocolo de transferencia de hipertexto: sistema mediante el cual se transfiere información entre los servidores y los clientes (por ejemplo los navegadores).

¹⁶ Lenguaje de marcas de hipertexto (HyperText Markup Language) para la elaboración de páginas web.

- e) **Aplicaciones P2P:** Permiten compartir archivos en Internet con otros usuarios. Destacan Emule, BitTorrent o Panda.
- f) **Software Firewall:** Son aplicaciones que impiden el intrusismo en nuestro equipo a través de Internet, además de establecer restricciones en nuestro propio equipo al usar Internet. Destacan Zone Alarm, Norton Internet Security.

En la Figura 3 se presenta una tabla con las etapas de la WEB, el nacimiento de esta tecnología se basa en la Web 1.0 que “se concibió como una colección de documentos estáticos enlazados para su consulta o descarga, donde los usuarios no podían interactuar con ellos, teniendo que hacer un consumo pull, es decir, se iba a por el recurso siendo la actualización unidireccional” (Lerma, Murcia, & Mifsud, 2013, pág. 9). Se fue popularizando cada vez más en consecuencia aumentaron sus prestaciones y se hizo necesario desarrollar métodos que permitiesen la creación de páginas con contenidos dinámicos .


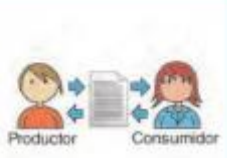



Web 1.0	Web 1.5	Web 2.0	Social media	Web semántica
		 El C/P es el Prosumidor		
estática	dinámica	colaborativa	social	inteligente
1989 – 1997	1997 – 2003	2003 – 2008	2008 – Actual	En un futuro cercano

Figura 3, Evolución de la Web

Fuente. (Lerma et al. 2013)

En la Web 1.5 se realizaban transacciones en función a las peticiones, en esta etapa aparecen lenguajes de programación y arquitecturas tanto del lado del servidor como del lado del cliente, en el lado del servidor lenguajes de programación como PHP, ASP, JSP y sistemas de ejecución de módulos, en el lado del cliente lenguajes de script como JavaScript o VBScript, applets, etc. Todo esto para construir páginas web con acceso a los SGBD¹⁷ y totalmente dinámicas e interactivas.

Ya en el año 2003 empieza a ponderar la era de la colaboración y se utilizan Tecnologías RSS¹⁸ al usuario ahora le llegan los nuevos contenidos sin tener que ir a buscarlos incluso la actualización del contenido pasa a ser bidireccional. Los recursos son más fáciles de producir

¹⁷ Sistema gestor de base de datos: conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.

¹⁸ Really Simple Syndication: un formato XML para distribuir contenido en la web.

y consumir. Accediendo a través de la red mediante un navegador u otras aplicaciones específicas. Estas general de manera dinámica una serie de páginas con tecnología AJAX¹⁹ que interpretara y presentara el cliente. (Lerma et al. 2013).

Finalmente en la era actual en la que vivimos la de la socialización, la donde las redes sociales y el internet móvil se apoderaron de la sociedad definiéndose así la etapa Social media.

Se hace una proyección a futuro, aunque ya en varios países del mundo es una realidad; la web semántica que hace la inclusión de metadatos y la interacción de cosas o maquinas, un ejemplo claro es la inteligencia artificial o el internet de las cosas.

1.1.3. Sistemas de Logística

Como se mencionó anteriormente un sistema de información se puede clasificar según diferentes consideraciones, ya se definió que será implementado un sistema web que desde el punto de vista de sus objetivos, que es un software que automatiza el flujo de datos de los procesos pertinentes a la logística de un parque automotor.

“Logística es el proceso de planear, implementar y controlar efectiva y eficientemente el flujo y almacenamiento de bienes, servicios e información relacionada del punto de origen al punto consumo con el propósito de cumplir los requisitos del cliente” (Ramírez, 2015, pág. 3). Esta función no depende del tamaño ni de la misión de la entidad, sino de la necesidad de la empresa de implementar una forma adecuada de gestión en la administración de los recursos y procesos, para así ofrecer una respuesta eficiente en cuanto al tiempo y calidad del servicio.

En la Figura 4 se muestra las estrategias o características que debe poseer un sistema logístico, y entre los más importantes se encuentra la existencia de una organización logística formal. Ramírez (2015) afirma:

En los últimos años, la gestión logística se ha facilitado para las pequeñas y grandes empresas gracias a la utilización de softwares que permiten que las diferentes áreas estén comunicadas entre sí. Por ejemplo, una compañía recibe un pedido vía Internet, este llega al área de compras, allí es re direccionado al almacén, donde se verifica si los productos están disponibles, y, luego, si se comprueba que hay existencias, es empaquetado y despachado para su transporte. Todo esto se realiza a través de una red computarizada, sin necesidad de papeleo y movilización de personal como se hacía cuando no existían estos programas. (pág. 10).

¹⁹ Asynchronous JavaScript And XML; es una técnica de desarrollo web para crear aplicaciones interactivas.



Fuente: Elaboración propia, con base en la visión del Georgia Institute of Technology.

Figura 4. Administración de la logística

Fuente. (Ramírez, 2015)

1.2. Arquitectura EDRA

EDRA (Enterprise Development Reference Architecture), también llamada “shadowfax” es una guía de arquitectura para estandarizar y resolver los problemas que se presentan en el desarrollo de sistemas distribuidos; nos provee de “quickstars”²⁰ y plantillas de aplicaciones, y su visión es percibir los beneficios de SOA²¹ con .NET sin experimentar grandes complicaciones en su implementación. (Fagas, 2007)

EDRA es una evolución de la arquitectura referencial de desarrollo empresarial Cliente/Servidor en la cual todo su proceso se resume básicamente en que el cliente realiza peticiones al servidor request/response. Es su transformación ya que este lineamiento viene a solucionar problemas que se presentaban como por ejemplo la entrega de mensajes, el manejo de errores y soporte de transacciones.

En la Figura 5 se muestra la evolución antes mencionada, integrando al modelo del cliente/ servidor los servicios web y su interfaz.

²⁰ Es un método de carga utilizado por varias aplicaciones de software diferentes, diseñado para acelerar el tiempo de carga de su software.

²¹ Arquitectura Orientada a Servicios: que construye la app con un conjunto de servicios que son una representación lógica de una actividad de negocio.

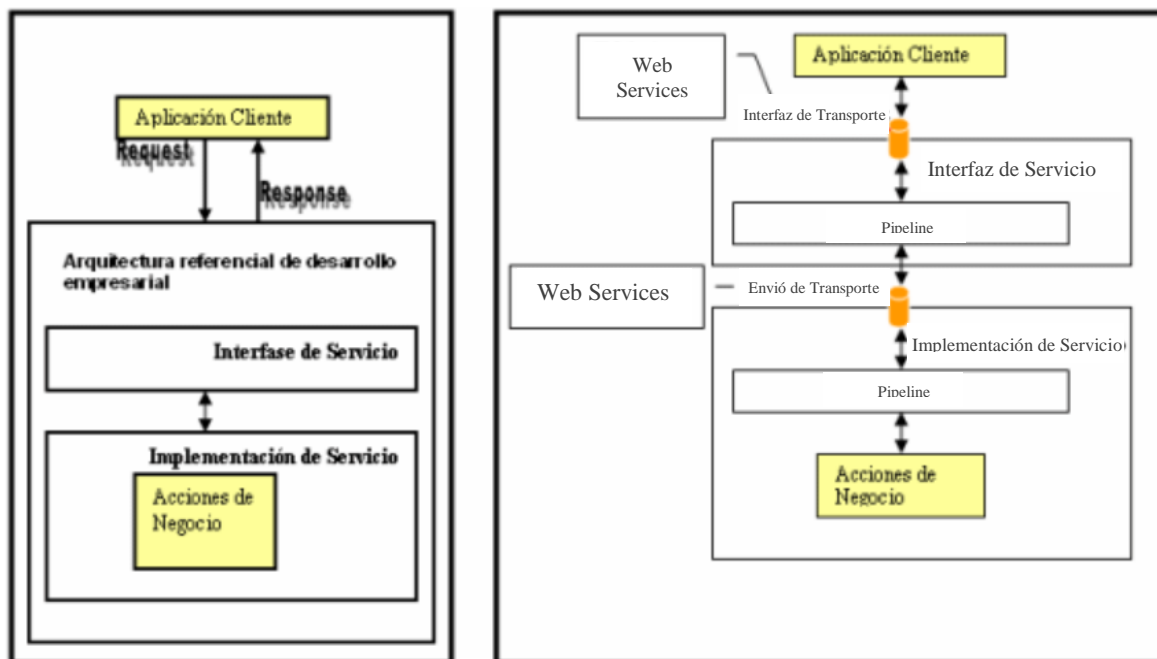


Figura 5. Modelo Conceptual EDRA

Fuente.

(Fagas, 2007) Afirma que: EDRA utiliza cuatro principios orientadores que ayudan a aliviar la transición de objetos distribuidos altamente acoplados a servicios basados en estándares y con bajo acoplamiento y son:

- a) Separación de la interfaz del servicio y la implementación interna del servicio para permitir escenarios de instalación que estén optimizados para tener escalabilidad, fiabilidad, seguridad, rendimiento y disponibilidad.
- b) Separación de la lógica de negocio y las preocupaciones de corte cruzado tales como auditoría, monitoreo o disparo de eventos de negocio. (Una preocupación de corte cruzado [cross-cutting concern] es un tipo de funcionalidad que puede aplicarse a múltiples clases o aplicaciones).
- c) Separación de la lógica de negocios y el transporte subyacente de manera que múltiples transportes se puedan usar para acceder a una única implementación de un servicio.
- d) Desarrollo de interfaces de servicio estables para asegurar la robustez de los servicios instalados.

Los principios en los que se basa este modelo hacen mucho énfasis en la separación de la lógica del negocio con las demás capas de la aplicación, todo esto para construir componentes de manera rápida y reutilizable.

Una vez analizados los objetivos propuestos por Microsoft para esta arquitectura se considera lo siguientes: construir un servicio y luego exponerlo, mantener la lógica de servicio separada de la lógica de auditoría o supervisión o cualquier otra secundaria a la lógica del negocio, lograr una interfaz robusta y estable.

Para su uso EDRA provee un aplicación de iniciación la cual posee una plantilla basada en la referencia de un banco ficticio, Global Bank en un escenario de banca en línea, esta app fue una iniciativa apoyada en un fuerte feedback de la comunidad y la industria.

Después de que se implementara EDRA con esta plantilla y debido a que solo se enfocaba en un modelo de negocio, surgió la implementación prescriptiva de Shadowfax denominada MBI 3.0, que fue una integración entre las arquitecturas MBI con EDRA.

1.2.1. MBI 3.0 & EDRA

MBI.1.x comenzó con un objetivo claro proveer un marco integrado que permita facilitar la elaboración de aplicaciones con el uso de herramientas de Microsoft Corporation, las cuales están basadas en un modelo de procesamiento distribuido y definida como:

MBI (Microsoft Business Integrator) es una arquitectura de referencia para la construcción de aplicaciones, cuya arquitectura de integración común a todos los “portfolios” de soluciones utiliza las mejores prácticas de Microsoft Corporation; en otras palabras, MBI es un framework para crear, ejecutar y mantener aplicaciones corporativas basadas en la plataforma Microsoft .NET, desarrollado para resolver escenarios recurrentes de una empresa corporativa y la reducción de costos de desarrollo y mantenimiento. (Fagas, 2007)

Pero a medida que las necesidades surgían, iba evolucionando, en la figura 6 se muestra como en el 2002 se estandarizo como MBI 1.x y en el 2004 MBI 3.0 se implementa como componente central a Shadowfax.

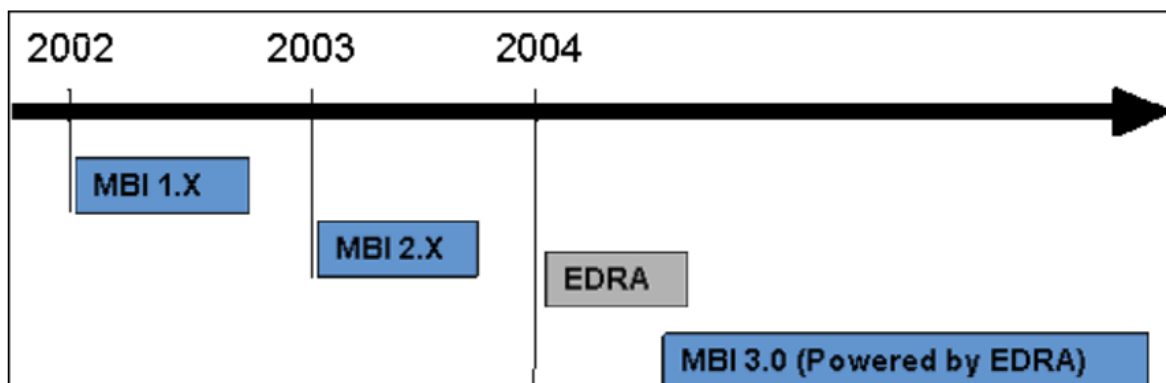


Figura 6. Evolución de MBI

Fuente. (Fagas, 2007)

MBI.3.0 profundiza a EDRA, a través de una arquitectura mucho más detallada, prescriptiva y refinada basada en la experiencia que ha acumulado Microsoft a través de sus servicios de consultoría y soporte. Lo que busca es proveer una plataforma tecnológica robusta, confiable, económica y ágil y abarca todos los componentes de los sistemas de información, sean estos técnicos u organizacionales.

(Fagas, 2007) Afirma que MBI 3.0 implementa como componente central a EDRA y lo complementa en escenarios que éste aún no resuelve. Entre los beneficios que nos provee el uso de MBI & EDRA destaco los siguientes:

- a) Menos código que mantener
- b) Código de base mejorado
- c) Mayor esfuerzo en horas de pruebas
- d) Modelo más flexible y extensible
- e) Integración con los Application Blocks²² corporativos (EIF, CMAB, etc.)
- f) Maximización del uso de la plataforma

Sobre menos código que mantener, se refiere a que el programador no debe invertir mucho tiempo en el mantenimiento del sistema eso se logra con: parametrización, usuarios, entre otros; el código de base mejorado hace referencia a la separación de la lógica del negocio; el mayor esfuerzo en horas de prueba sirve para solventar rápidamente y totalmente errores o validaciones que se presentan en los procesos de negocio reales ; el modelo más flexible y extensible tiene que ver con la implementación de servicios web; la integración con los Application Blocks resalta la biblioteca de bloques de propósito general que posee el framework, también la capacidad de crear pequeños bloques reutilizables; en cuanto a la maximización del uso de la plataforma, la arquitectura recomienda construir una arquitectura técnica común, por ello se construye el sistema con herramientas de Microsoft para una integración fácil y más segura.

1.2.2. Servidor Web IIS

IIS 8(Internet Information Services) es la última versión del servidor Web de Microsoft. Se incluye de forma completa con Windows Server 2012, y proporciona una plataforma segura y fácil de administrar para albergar servicios Web y aplicaciones Web enriquecidas. Con Windows Server 2012 R2, se proporciona la versión 8.5 de la aplicación IIS 8 es una plataforma Web unificada que integra IIS, ASP.NET, PHP, servicios FTP y Windows Communication Foundation (WCF: el modelo de programación de WCF es una capa de abstracción que unifica y simplifica el mecanismo de integración de los servicios Web, .NET

²² Los bloques de aplicación es una iniciativa que implica la construcción de pequeños bloques de aplicación bajo ciertas especificaciones.

Remoting, Microsoft Transaction Server y Microsoft Message Queuing). (Deman, Elmaleh, Neil, & Van Jones, 2014, pág. 553)

En la Figura 5 se detallan las principales características del servidor IIS, entre sus puntos fuertes están la facilidad de instalación, configuración y administración; esto permite que IIS sea un servidor Web intuitivo y fácil de instalar, centralizado con su configuración y administración a través de la consola.

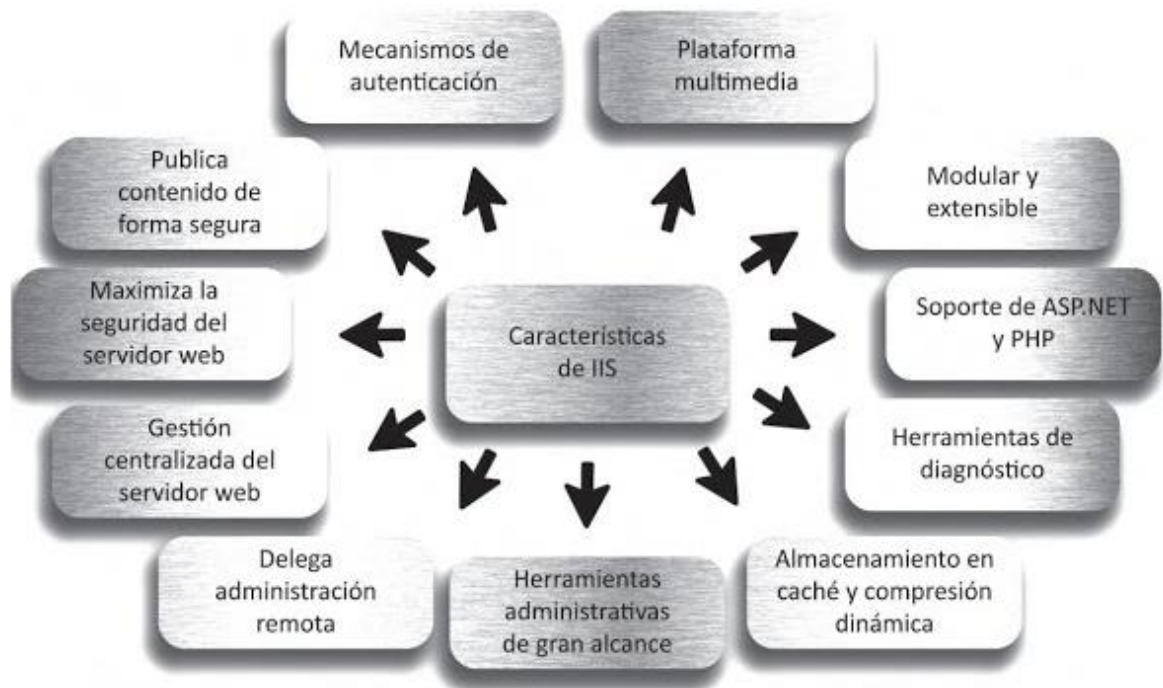


Figura 7. Características de IIS

Fuente. (Autores, 2016)

1.2.3. Plataforma Microsoft.NET

Corresponde a un framework o IDE de desarrollo de aplicaciones a partir de una estructura básica de código fuente prediseñadas, además provee a los desarrolladores potentes herramientas integradas para facilitar el ciclo de desarrollo de software.

“Framework es la estructura compuesta por módulos que pueden ser personalizados e intercambiables para el desarrollo de aplicaciones web, este conjunto de herramientas puede ser considerado como una aplicación configurable a la que se le puede añadir módulos y componentes necesarios para desarrollar aplicaciones concretas” (Gutierrez, 2013, pág. 17).

Utilizar un framework facilita mucho la creación de un proyecto de software, el mantenimiento y la escalabilidad con muchas ventajas como; el uso de patrones de diseño, una estructura predefinida y estandarizada dependiendo de la tecnología a implementar, organización de archivos y código, utilidades y librerías, alto testeo de código, soporte en una amplia comunidad, etc.

Se trata de un entorno de desarrollo multilenguaje diseñado por Microsoft para simplificar la construcción, distribución y ejecución de aplicaciones para Internet. Tiene fundamentalmente tres componentes: una máquina virtual (CLR: Common Language Runtime) que procesa código escrito en un lenguaje intermedio (MSIL: Microsoft Intermediate Language), una biblioteca de clases (biblioteca .NET) y ASP.NET que proporciona los servicios necesarios para crear aplicaciones Web. (Ceballos, 2017, pág. 10)

En la Figura 6 se muestra el flujo de procesos al ejecutar un programa escrito sobre el paquete .NET. El programa escrito en Visual Basic o algún lenguaje de programación, pasa por el compilador que es un programa traductor que produce un código MSIL que es un lenguaje intermedio común a todos los sistemas operativos que soporten .net framework el cual la máquina virtual ejecutará, finalmente termina el proceso.



Figura 8. Ejecución de un programa en .NET

Fuente. (Ceballos, 2017)

(López, 2013) Afirma: A continuación se detallan las funciones y características de los componentes del Framework .NET entre los que se destacan:

a) **Common Language Runtime (CLR):** Es el componente principal de la plataforma .NET que se encarga de administrar la ejecución de las aplicaciones desarrolladas. También, una máquina virtual en la que funcionan las aplicaciones .NET, esto es, que teniendo el CLR correspondiente es posible ejecutar cualquier aplicación diseñada en la plataforma .NET. De igual manera, el CLR controla la estandarización de tipos, es decir, la plataforma .NET comparte los tipos de datos, de ahí que es posible utilizarlos de la, misma forma en diferentes lenguajes.

b) **Librerías de clases** Contiene gran cantidad de clases prediseñadas que el programador utiliza para crear sus clases. La librería de clase base (BCL) es una librería del Framework.NET que contiene los tipos de datos y una gran cantidad de funciones para el

desarrollo de sistemas. Por la cantidad de clases del Framework.NET, es necesario organizarlas en espacios de nombres (namespaces), de los que mencionaremos algunos:

- System: Contiene los tipos de datos básicos, excepciones, control de entrada-salida en consola, etcétera.
- System.Collections: Son estructuras de datos, entre las cuales las más comunes son pilas, colas, listas, diccionarios, etcétera.

c) **Microsoft Intermediate Language (MSIL):** Los compiladores de la plataforma .NET no generan código para un microprocesador en particular, sino que lo hacen en un lenguaje intermedio llamado MSIL, lo que provoca a que el CLR solamente reconoce e interpreta el MSIL durante la ejecución de las aplicaciones creadas en la plataforma .NET, por eso los diferentes lenguajes de la plataforma .NET generan código compatible, ya que sus aplicaciones están convertidas a MSIL :

d) **Just In Time (JIT):** Es un componente del CLR que convierte el código MSIL a código nativo durante la ejecución de una aplicación desarrollada en código .NET.

1.2.4. Lenguaje de Programación C#

Entendemos por programación al proceso por el cual se escribe en un lenguaje la solución a un problema resultando de ello lo que se conoce como software. Para poder llevar a cabo dichos programas se emplea la lógica de programación, que incluye estructuras de decisión, de repetición o de asignación. (Ollero, 2015, pág. 15)

En la Figura 7 se definen las partes imprescindibles que conforman la escritura de cualquier programa los cuales son: cabecera, una zona de definición de variables y el cuerpo del programa.



Figura 9. Partes de un programa

Fuente. (Ollero, 2015)

La cabecera es donde debemos explicar resumidamente qué hace el programa, podemos incluir el nombre del programador y cualquier información que pueda ser relevante para un posterior mantenimiento. La siguiente parte de un programa es la zona de definición de variables; en muchos lenguajes podemos declarar una variable en cualquier zona, es muy recomendable agruparlas todas al principio del programa, de esta manera sabemos dónde acudir y donde buscar cuando necesitamos saber aquello que está definido y cómo lo está. Si desperdigamos nuestras variables, definiéndolas allí donde va surgiendo su necesidad solo lograremos que nuestro programa sea muy difícil de seguir y de mantener, incluso para nosotros mismos. Por último nos queda el cuerpo del programa, en esta parte es donde irá el contenido propiamente dicho del programa. (Ollero, 2015, pág. 19)

C# es un lenguaje de programación orientado a objetos que admite los conceptos de encapsulación²³, herencia²⁴ y polimorfismo²⁵, capaz de crear aplicaciones que se ejecutaran en .NET CLR. “Es una evolución de los lenguajes C y C++ y ha sido creado específicamente por Microsoft para trabajar con la plataforma .NET. C# ha sido diseñado para incorporar muchas de las mejores características de otros lenguajes, al tiempo que reduce sus problemas” (Perkins, Vibe, & Reid, 2016, pág. 9).

Permite crear aplicaciones cliente de Windows, servicios web XML²⁶, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas más cosas. Visual C# proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje C# y .NET Framework.

Desarrollar aplicaciones usando C# es más simple que usar C++, porque la sintaxis del lenguaje es más simple. Aun así, C# es un lenguaje poderoso, y hay muy poco que quieras hacer en C++ que no puedas hacer en C#. Habiendo dicho eso, esas características de C# que son paralelas a las características más avanzadas de C++, como acceder y manipular directamente la memoria del sistema. (Perkins, Vibe, & Reid, 2016, pág. 9)

Aprender C# es bastante sencillo ya que se parece mucho al lenguaje de programación Java²⁷, el cual es el más utilizado y difundido en el medio, por tanto si aprende a programar en Java no será nada nuevo emplear C#.

²³ Ocultamiento del estado de los datos miembro de un objeto.

²⁴ Permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares.

²⁵ Permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares.

²⁶ Lenguaje de Marcas Extensible (eXtensible Markup Language): meta-lenguaje que permite definir lenguajes de marcas

²⁷ Lenguaje de programación diseñado específicamente para tener pocas dependencias de implementación, es uno de los más populares en uso para aplicaciones cliente/servidor.

1.2.5. Base de Datos SQL Server

Antes de que las bases de datos aparecieran la información se gestionaba a través de sistemas de gestión de archivos, donde si la estructura de los datos cambiaba se debía modificar todos los programas que los manejaban, y este gran inconveniente era muy difícil de administrar en grandes volúmenes de información debido a esto surgen los SGBD donde se puede modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan.

Los sistemas de gestión de bases de datos o SGBD (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. En concreto, definimos un Sistema Gestor de Bases de Datos como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina base de datos. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. (Valderrey, 2014, pág. 37)

Es un sistema de administración de bases de datos de clase empresarial (DBMS) que es capaz de ejecutar cualquier cosa desde una base de datos personal de solo unos pocos megabytes en un dispositivo W Mobile de mano hasta un sistema de base de datos multiservidor administrando terabytes²⁸ de información. Sin embargo, SQL Server 2014 es mucho más que solo el motor de base de datos. Se compone de varios componentes. (Rankins, Bertucci, Gallell, & Silverstein, 2015)

(Gabillaud, 2015) Afirma: Las diferentes piezas de software que proporciona SQL Server se articulan siempre en torno al motor de base de datos relacionales, que trata de manera eficiente la información almacenada en formato relacional y formato XML:

- a) **SQL Server Integration Services (SSIS):** es una herramienta de importación y exportación de datos fácil de utilizar y muy parametrizable
- b) La replicación de los datos en diferentes instancias permite colocar los datos lo más cerca posible de los usuarios y reducir los tiempos de procesamiento.
- c) **Service Broker** permite el trabajo en modo asíncrono y facilita la gestión de los picos de fuerte actividad, almacenando las peticiones de trabajos antes de tratarlas.
- d) La integración de CLR en SQL Server permite desarrollar procedimientos y funciones utilizando los lenguajes VB.NET y C#. La integración de CLR no pretende sustituir a Transact SQL, aunque se presenta como un complemento con el objetivo de poder realizar

²⁸ 1.000 gigabytes cantidad de datos almacenados en el disco duro de una computadora.

una codificación simple y de buen rendimiento para el conjunto de las funcionalidades que deben estar presentes en el servidor.

(Gabillaud, 2015) Afirma: SQL Server puede gestionar dos tipos de bases de datos diferentes:

a) Las bases OLTP (OnLine Transactional Processing), que corresponden a las bases de datos en las cuales la información se almacena de manera directa, con el objetivo de reutilizarla más tarde tal y como fue almacenada.

b) Las bases OLAP (Online Analytical Processing), que contienen información estadística con el objetivo de poder extraer información en forma de cubos multidimensionales para ayudar en la toma de decisiones, por ejemplo. Las estadísticas contenidas en las bases OLAP están basadas en la información incluida en una base OLTP.

1.3. Metodología SCRUM

Esta metodología es una de las más populares no solo en el Ecuador sino en todas partes del mundo, se dejó atrás lineamientos rígidos, donde se perdía demasiado tiempo en cantidades exageradas de documentación y no había la retroalimentación y participación directa de los usuarios y equipo de trabajo, esto sucede con metodologías no ágiles, pero SCRUM llega como una propuesta donde se plantea desde una estructura y procesos sencillos y bien definidos donde ese prioriza el trabajo en equipo, la adecuación de los cambios que puedan surgir y sobre todo la producción de software ideal para el usuario y en menor tiempo.

“Es un marco de trabajo dentro del cual se puede emplear varias técnicas y procesos para entregar productos de máximo valor productivamente y creativamente. Se basa en la teoría de control de procesos empírica o empirismo²⁹” (ScrumInc, Scrum.Org, 2014).

Al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves o periodos definidos de tiempo para el desarrollo, que comúnmente se las llama iteraciones o “SPRINTS”, mientras que los INCREMENTOS representan requisitos que se han completado en una iteración y que son perfectamente operativos. Según los resultados sé que obtenga, el cliente puede ir haciendo los cambios necesarios.

- a) Es una metodología de desarrollo muy simple que requiere trabajo duro.
- b) Orientado a personas más que a procesos
- c) Emplea la estructura de desarrollo ágil

²⁹ Aprender y mejorar de los errores pasados.

d) Modo de desarrollo de carácter adaptable

“Sprint es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto. Los Sprint contienen; Reunión de Planificación del Sprint, Scrums Diarios, trabajo de desarrollo, revisión del Sprint y retrospectiva del sprint” (ScrumInc, Scrum.Org, 2014).

El enfoque de este marco de trabajo es dividir las funcionalidades e ir las trabajando en cada periodo de tiempo donde se necesita un equipo consolidado, no solo del parte del grupo de desarrolladores sino también por parte de los interesados y usuarios finales donde en todo el proceso se realizan una serie de reuniones para garantizar que el sprint cumpla con los objetivos planteados.

En la Figura 10 se muestra las reuniones necesarias al desarrollar un sprint: reunión de planificación, reuniones diarias y la reunión final, donde se presenta el incremento, se realiza la retroalimentación y se estima el nuevo sprint si es necesario.

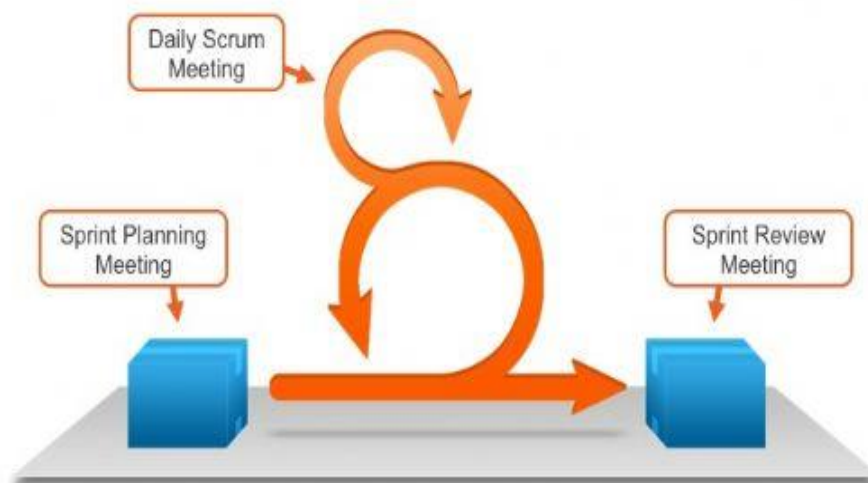


Figura 10. Estructura de un Sprint

Fuente. (Trigas, 2012)

La Figura 11 detalla el ciclo de vida de la metodología que se basa en el ciclo fundamental del desarrollo del software: Análisis, Diseño, Codificación, Pruebas e Integración, la diferencia está en que este proceso se cumple en cada iteración o sprint y se controla bajo reuniones de trabajo diarias, potenciando así el desarrollo y calidad en tiempos de entrega y funcionalidad de software.



Figura 11. Ciclo de la metodología SCRUM

Fuente. (Trigas, 2012)

1.3.1. Roles

Los roles son las responsabilidades y derechos que tiene un participante interesado en la construcción del software, estos se asignan para que cada integrante sepa las actividades que debe llevar a cabo y los objetivos específicos que debe sacar adelante y propiciar su culminación.

(ScrumInc, Scrum.Org, 2014) Afirma que existen dos grupos bien definidos a diferenciar, los comprometidos y los involucrados. La Tabla 1 muestra los roles que pertenecen a estos grupos y a continuación se explica cada uno.

TABLA 1.1
GRUPOS DE ROLES DE SCRUM

COMPROMETIDOS	INVOLUCRADOS
Product Owner	Usuarios
Scrum Master	StakeHolders
Scrum Team	Managers

Tabla 1. Roles de Scrum

Fuente. (Trigas, 2012)

Product Owner: Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Decide en última instancia cómo será el resultado final. Conoce el plan del producto, sus posibilidades y el plan de inversión.

Scrum Master: Es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores. Es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer, garantiza que el equipo de trabajo no tenga impedimentos u obstáculos para abordar sus tareas dentro del proyecto.

Scrum Team: suele ser un equipo de personas que tienen la autoridad para organizar y tomar decisiones para conseguir su objetivo. Es un equipo multifuncional. Se marca diferencias entre grupo de trabajo y equipo comparten el objetivo y la responsabilidad de logro todos participan en las decisiones conocen el modelo SCRUM.

Usuarios o Cliente: son los beneficiarios finales de la aplicación a desarrollar.

SteakHolder: las personas a las que el proyecto les producirán un beneficio. Participan durante las revisiones del Proyecto

Manager: toman las decisiones finales participando en la selección de los objetivos y de los requisitos, las acciones tienen relación directa con los actores las acciones de Scrum forman parte de un ciclo iterativo repetitivo y tienen como objetivo minimizar el esfuerzo y maximizar el rendimiento en el desarrollo.

1.3.2. Componentes

En Scrum no existe una documentación extensa como en otros marcos de desarrollo típicos, a continuación se detallan cada uno de los documentos necesarios:

Historias de Usuario: son los requerimientos funcionales se detallan pequeñas plantillas. La figura 13 muestra el formato que esta debe llevar y toda la información necesaria, en el ANVERSO se encuentra:

- a) **ID:** muestra el número de requerimiento o un identificador único por cada historia.
- b) **TITULO:** una descripción general de la funcionalidad a implementar.
- c) **DESCRIPCION:** se detalla los procesos que se deben llevar a cabo para este requerimiento de una manera más específica.
- d) **ESTIMACIÓN:** cuánto tiempo se va a destinar a este requisito, el cual se puede estimar en horas e incluso días, las unidades de estimación se definen antes de elaborar las historias de usuario aunque generalmente se mide en horas.

e) **PRIORIDAD:** el grado de importancia que posee una historia de usuario con relación a las demás.

f) **DEPENDENCIAS:** que otros requerimientos dependes de este.

Mientras que en el reverso se encuentran las pruebas de aceptación de cada proceso, es decir las pruebas funcionales que se deben realizar para establecer como realizada y finalizada esa historia de usuario.

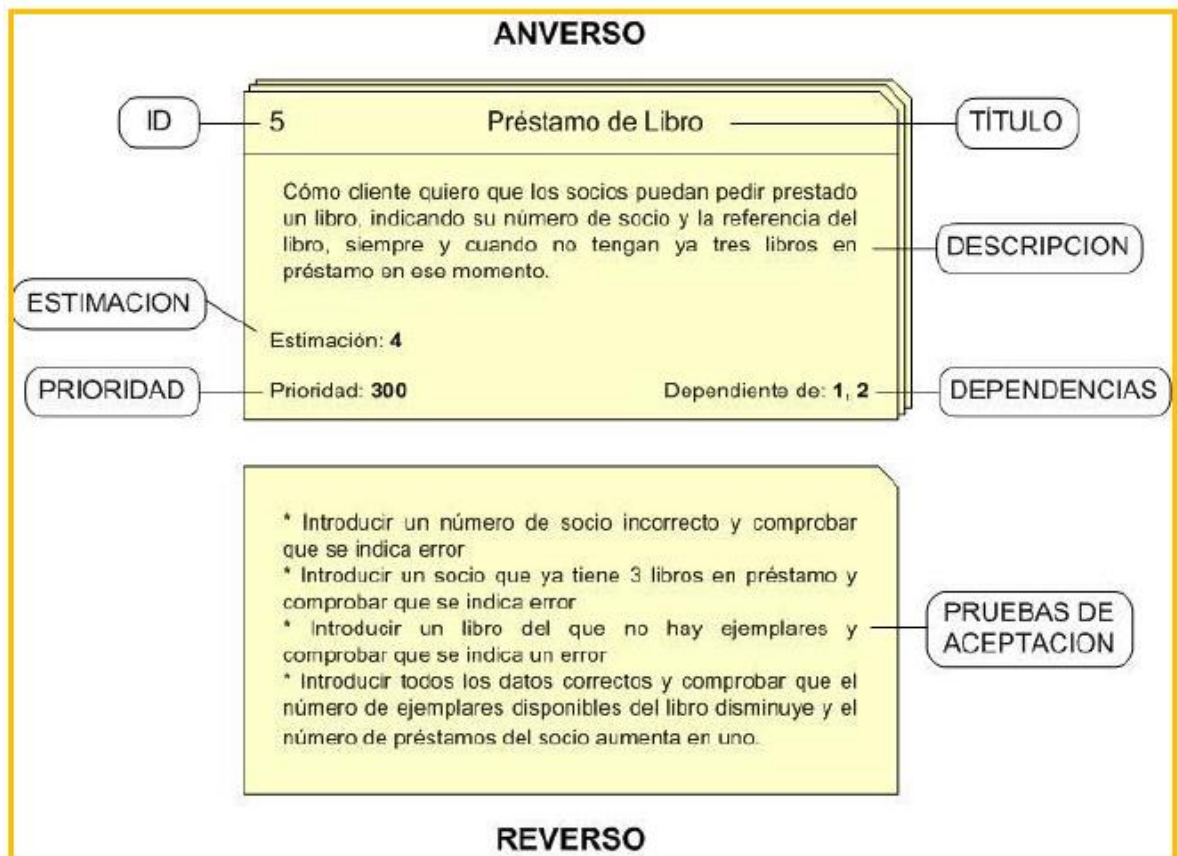


Figura 12. Historia de Usuario

Fuente. (Trigas, 2012)

Pila de Producto: es una lista de las historias de usuario, que se ordena por prioridad y se asigna a cada miembro del Scrum Team. Esta se construye como herramienta de comunicación directa entre el equipo. La figura 6 muestra el formato que esta debe llevar y toda la información necesaria, esta contiene los mismos datos que las historias de usuario solo que sin el detalle del requisito y en forma de lista.

(ScrumInc, Scrum.Org, 2014) Afirma que:

- Registra los requisitos vistos desde el punto de vista del cliente.
- Está formada por la lista de funcionalidades o "historias de usuario".

- c) La lista será creada por el cliente o usuario con la ayuda del Scrum Master.
- d) Es una referencia para el equipo.
- e) La lista debes ser ordenada por la prioridad de los elementos.
- f) Se elabora con la colaboración del Equipo partiendo de la visión del Product Owner.

Id	Prioridad	Descripción	Est.	Por
1	Muy alta	Plataforma tecnológica	30	AR
2	Muy Alta	Interfaz de usuario	40	LM
3	Muy Alta	Un usuario se registra en el sistema	40	LM
4	Alta	El operador define el flujo y textos de un expediente	60	AR
5	Alta	xxx	999	CC

Figura 13. Pila de Producto

Fuente. (ScrumInc, Scrum.Org, 2014)

Lista de Tareas: se elabora durante la planificación de un Sprint por el Scrum Team, es una lista de control de avance de cada historia, donde se descompone cada historia de usuario en tareas más pequeñas, se les estima un tiempo determinado posteriormente se va registrando en avance de cada una en cada día. La figura 14 muestra un formato bastante sencillo que puede tener este componente, donde se detalla el sprint en que se va a trabajar; inicio y días de duración. Las tareas a realizarse, su estado y el registro de horas trabajadas.

SPRINT		INICIO	DURACIÓN						
1		01-mar-07	5	L	M	MI	J	V	
				01-mar	02-mar	03-mar	04-mar	05-mar	
				TAREAS PENDIENTES	5	4	3	2	0
				ESFUERZO	40	32	26	14	0
SPRINT BACKLOG					ESFUERZO				
Est	HU	Tarea	Estado	Responsable					
6	1	Descripcion 1	Terminado	Kathy	6				
10	1	Descripcion 2	Terminado	Pedro	10	10	8	4	
4	2	Descripcion 3	Terminado	Pedro	4	2			
6	3	Descripcion 4	Terminado	Kathy	6	6	4		
14	4	Descripcion 5	Terminado	Kathy	14	14	14	10	

Figura 14. Lista de Tareas

Fuente. (ScrumInc, Scrum.Org, 2014)

(ScrumInc, Scrum.Org, 2014) Afirma que:

- a) Lista que descompone las funcionalidades de la pila del producto (historias de usuario) en las tareas necesarias para construir un incremento.
- b) Descompone el trabajo en unidades de tamaño más pequeñas
- c) Es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender.
- d) A medida que el trabajo se ejecuta, se va actualizando la estimación del trabajo.
- e) Se hace el seguimiento del trabajo en cada Scrum Diario.
- f) Realizarla con todos los miembros del equipo.
- g) Las tareas demasiado grandes deben descomponerse en otras más pequeñas.

El formato que tiene la lista se puede gestionar mediante; hojas de cálculo, herramientas de software colaborativas o pizarras, donde es importante ir cambiando el estado de cada tarea para visualizar su avance.

1.3.3. Fases del Proyecto

Antes de entrar en las 3 etapas que se mencionan a continuación, se debe establecer o asignar los roles a cada participante para que cada uno asuma las actividades que se le corresponde.

(ScrumInc, Scrum.Org, 2014) Afirma: Que existen tres fases bien definidas:

- a) **Preparación del Proyecto:** Se intenta comprender el caso de negocio donde se toman decisiones que agreguen valor al producto.
- b) **Planificación del Sprint:** En esta etapa se desarrolla el "Sprint Planning Meeting", que es donde el equipo trabaja para conseguir un incremento del producto.
- c) **Desarrollo del Sprint:** Se proporciona criterios Objetivos de gestión y seguimiento. Mide el trabajo pendiente. Se indica de forma clara el propósito del proyecto, para que todo el equipo sea capaz de entender las necesidades del producto y del cliente. Definir "TERMINADO" y unidades de estimación.

La Figura 15 muestra el diagrama de detallado de las 3 fases de SCRUM, donde los procesos dentro del desarrollo de los Sprint se van repitiendo hasta que los objetivos del proyecto sean cumplidos en su totalidad.

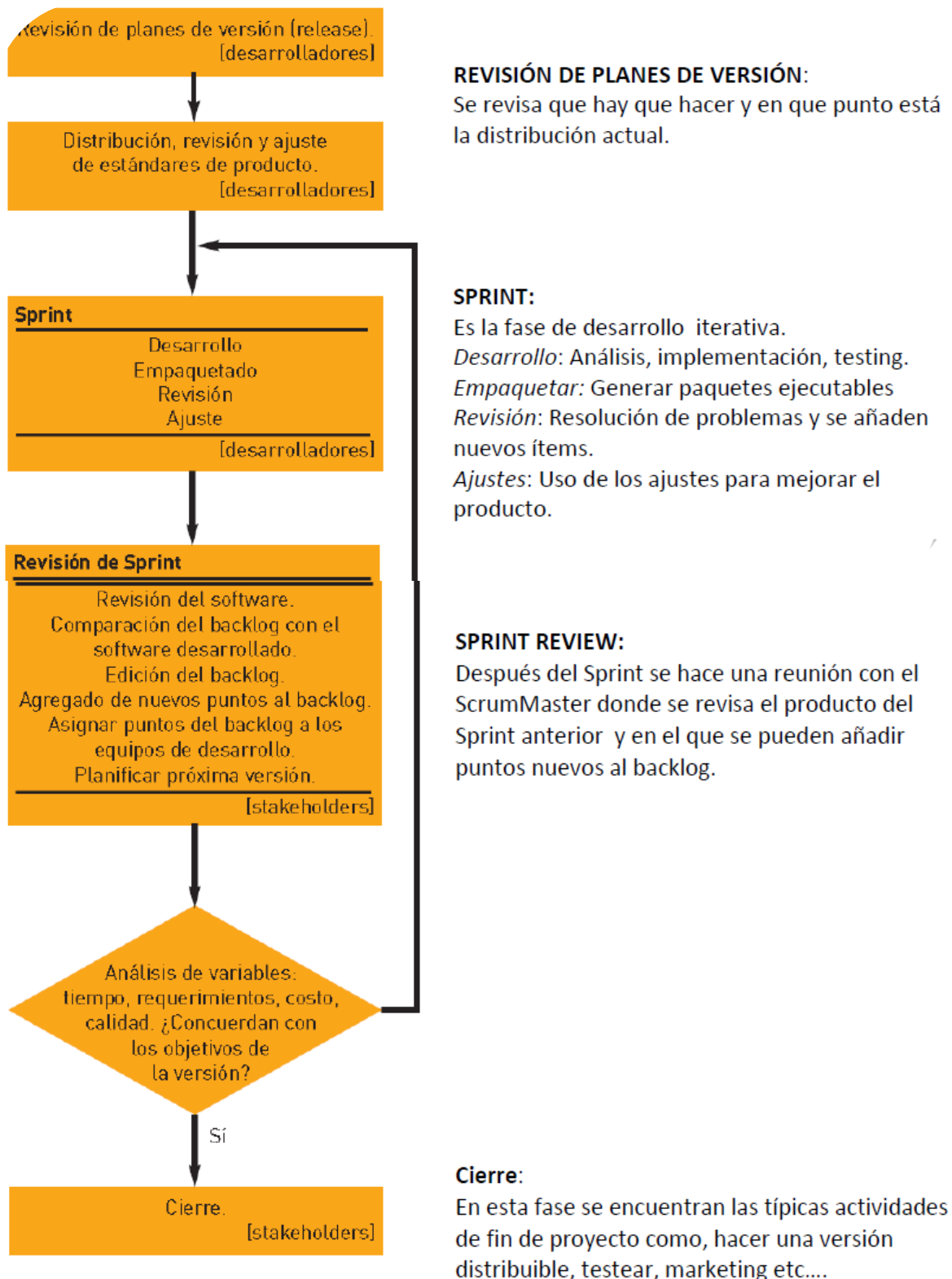


Figura 15. Diagrama Detallado de las fases de Scrum

Fuente. (ScrumInc, Scrum.Org, 2014)

1.3.4. Ventajas

- a) Definiciones de iteraciones con fechas cortas de trabajo (2 a 4 semanas).
- b) El cliente puede comenzar a utilizar el producto rápidamente.
- c) El cliente puede decidir los nuevos objetivos a realizar.
- d) Se agiliza el proceso, porque se divide el problema en pequeñas tareas.
- e) Menos probabilidad de que se den sorpresas o desarrollos inesperados porque el cliente va viendo poco a poco lo que se está desarrollando.
- f) Es aplicable a cualquier tecnología y lenguaje de programación.

1.3.5. Desventajas

- a) Dejar tareas inconclusas puedan obstaculizar la planeación de nuevas sprints y se deba volver al problema original.
- b) Alto nivel de stress de los miembros del equipo, el desgaste puede ser excesivo y estresante lo que puede disminuir el rendimiento.
- c) Permite fallar si es necesario.
- d) El equipo de trabajo con poca experiencia.

- **Framework Bootstrap**

Es un enfoque de diseño web destinado a la elaboración de sitios web para proporcionar una visualización óptima para una experiencia de navegación fácil y con un mínimo de cambio de tamaño, paneo y desplazamiento a través de una amplia gama de dispositivos (de los monitores de ordenador de escritorio al terminal de móviles). (Arias, 2014).

Este un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones. “Un sitio diseñado con Bootstrap adapta su diseño a las condiciones de observación mediante el uso de fluidos, las proporciones basadas en cuadrículas, imágenes flexibles y CSS3 en los medios” (Arias, 2014),

“Las imágenes flexibles también están dimensionadas en unidades relativas, a fin de evitar que se muestren fuera de su elemento contenedor” (Arias, 2014).

“Las consultas de los medios de comunicación permiten a las paginas utilizar diferentes reglas de estilo CSS, basados en las características del dispositivo donde la página está siendo mostrada, más comúnmente la anchura del navegador” (Arias, 2014).

- **SpringToMeter**

Es una herramienta de gestión y seguimiento de proyectos ágiles ya sean estos SCRUM O XP, es muy sencilla y fácil de utilizar, no posee un registro de avances calendarizado con la fecha actual, es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones. Permite “simplificar el intercambio de datos con programas externos, todos los cuadros y hojas de cálculo en Sprintometer se pueden exportar a Microsoft Excel. El formato XML para archivos locales también se puede usar.” (Sprintometer Team, 2011)

(Sprintometer Team, 2011) Las siguientes características principales son compatibles con Sprintometer:

- a) Seguimiento de proyectos SCRUM y XP.
- b) Burn Down mejorado y otros gráficos útiles.
- c) Seguimiento de sprint / iteración con composición de equipo variable.
- d) Seguimiento separado de las tareas de desarrollo y prueba.
- e) Seguimiento de solicitudes de cambio.
- f) Predicción de la desviación del plan de sprint / iteración programado.
- g) Cálculo del alcance diario para desarrolladores y probadores para estar en el buen camino.
- h) Exportar a Microsoft Excel para todos los cuadros y hojas de cálculo.
- i) Datos almacenados en un archivo local o en una base de datos compartida en el servidor público www.sprintometer.com
- j) Conexión segura de HTTPS al servidor público.
- k) GUI moderna y fácil de usar en el estilo de Microsoft Office 2007.
- l) Asignación de recursos a tareas e historias de usuarios.
- m) Seguimiento protegido con contraseña del presupuesto del proyecto.
- n) Numerosos informes con todos los parámetros ágiles importantes.
- o) Guardar / cargar datos del proyecto en formato XML.

CAPÍTULO 2

Desarrollo

2.1. Preparación del proyecto

A continuación, se analiza el caso de negocio y se plantean los requisitos del sistema a través de la especificación de las funciones de los diferentes equipos de trabajo y gracias a la construcción de los diferentes artefactos de la metodología SCRUM, como son la pila de producto y la pila de tareas.

El sistema ha sido denominado **SISMANCAR** por sus siglas SIS: Sistema, MAN: Mantenimiento y CAR: carro, pretende ser diseñado como un eficiente sistema, que ara la diferencia en la planificación y ejecución de actividades del parque automotor, proporcionando procesos veloces y automatizados, manejo de datos inmediatos y sobre todo la estabilidad de un sistema confiable que brinda disponibilidad y eficacia en el servicio. Para una correcta gestión y seguimiento del proyecto SCRUM se utiliza la herramienta SPRINGTOMETER, en la cual se crea el proyecto de SISMANCAR.

2.1.1. Dimensión del Proyecto

SISMANCAR está dirigido a los diferentes actores del proceso funcional del parque automotor como son: administrativos del GAD, choferes, mecánicos y toda la comunidad publica e con el propósito de realizar una adecuada administración de los equipos, sus mantenimientos y evitar costos en daños de vehículos por mal mantenimiento , el sistema permitirá el registro de vehículo, plantillas de mantenimientos, planes y accidentes, también permite conocer la ubicación o ruta al que se transportó el vehículo. Y lo más importante es que el sistema permitirá planificar operaciones a los vehículos, alertara con un panel de notificaciones sobre el mantenimiento preventivo de los equipos.

El sistema estará terminado cuando se cumpla con todos los módulos descritos en el ALCANCE del proyecto, es indispensable cumplir con las historias de Usuario de prioridad: 1, 2, 3, y las historias de usuario con prioridad 4 serán desarrolladas solo si los usuarios finales del parque automotor lo requiere debido a políticas internas de emisión de documentos.

La estimación inicial para la finalización del proyecto será de 6 meses, el periodo de implementación y pruebas se inicia en enero del 2017.

2.1.2. Roles del Sistema

El Scrum Team o Equipo de Trabajo es fundamental para alcanzar metas y cumplir los objetivos del negocio, para satisfacer los requerimientos funcionales del sistema es necesaria la colaboración efectiva de todos los involucrados por ello a continuación se establecen los diferentes compromisos de cada rol participante de este proyecto.

ROL	NOMBRE	CARGO
Manager / Propietario del proyecto o Product Owner	Ing. Tomas Espinoza	Director del Área de Transporte del GAD
Jefe de proyecto o Scrum master	Msc. Mauricio Rea	Director de tesis
SteakHolder	Msc. Pablo Landeta	Delegado de tesis
Propietario del proyecto o Product Owner	Ing. Marco Sánchez	Técnico Automotriz del Área de Transporte del GAD
Scrum Team	Katherine Flores	Tesista

Tabla 2. Roles del Equipo de Trabajo

Fuente. Propia

2.1.3. Pila de Producto

En esta pila se describen la lista de necesidades del cliente donde se registran los requisitos funcionales del sistema vistos desde el punto de vista del stakeholder, también se detalla cada necesidad en una historia de usuario.

ID	PRIORIDAD	DESCRIPCION	ESTIMACIÓN	POR
H1	1	Plataforma Tecnológica	62 H	KF
H2	1	Seguridad: Usuarios y Roles	40 H	KF
H3	1	Administración General del Sistema	52 H	KF
H4	1	Administración de Parámetros	30 H	KF
H5	2	Gestión de Equipos	40 H	KF
H6	2	Gestión de Planes de Mantenimiento	120 H	KF
H7	3	Gestión de Mantenimientos No Planificados	88 H	KF
H8	3	Gestión de Revisiones Vehiculares	88 H	KF
H9	3	Gestión de Rutas y Kilometrajes	124 H	KF
H10	3	Aprobación de Peticiones	70 H	KF
H11	4	Inventario de Herramientas	60 H	KF
H12	4	Notificaciones	82 H	KF

Tabla 3. Pila de Producto

Fuente. Propia

2.1.4. Historias de Usuario

Cada necesidad descrita en la Pila de Producto se detalla como historia de usuario en el [ANEXO A](#).

2.1.5. Pila de Tareas

En esta lista se descompone las funcionalidades de la pila del producto (historias de usuario) en las tareas necesarias para construir un incremento, es decir descompone el trabajo en unidades de trabajo más pequeñas.

ESTIMACION	TAREA	ESTADO	POR
3 H	Instalación y Configuración de las herramientas	Por Hacer	KF
15 H	Creación de los Modelos de BDD	Por Hacer	KF
3 H	Creación de la Base de Datos del Sistema	Por Hacer	KF
15 H	Creación de los Procedimientos Almacenados	Por Hacer	KF
13 H	Pruebas con la Base de Datos	Por Hacer	KF
5 H	Levantamiento del Entorno y Conexión a la BDD	Por Hacer	KF
8 H	Pruebas de Integración	Por Hacer	KF

Tabla 4. Pila de Tareas H1: Plataforma Tecnológica

Fuente. Propia

ESTIMACION	TAREA	ESTADO	POR
5 H	Login y autenticación de acceso	Por Hacer	KF
6 H	Registro de datos personales y foto	Por Hacer	KF
5 H	Creación de usuarios y roles	Por Hacer	KF
8 H	Administración de usuarios y roles	Por Hacer	KF
8 H	Métodos de Autenticación y Acceso	Por Hacer	KF
8 H	Pruebas	Por Hacer	KF

Tabla 5. Pila de Tareas H2: Seguridad: Usuarios y Roles

Fuente. Propia

TABLA 2.18
PILA DE TAREAS H3: Administración General del Sistema

ESTIMACION	TAREA	ESTADO	POR
8 H	Configuración de Parámetros Generales	Por Hacer	KF
6 H	Configuración de Carpetas del Sistema	Por Hacer	KF
8 H	Servicio de E-mail	Por Hacer	KF
8 H	Servicio de SMS	Por Hacer	KF
6 H	Configuración de los Respaldos de BDD	Por Hacer	KF
16 H	Pruebas	Por Hacer	KF

Tabla 6. Pila de Tareas H3: Administración General del Sistema

Fuente. Propia

TABLA 2.19
PILA DE TAREAS H4: Administración de Parámetros

ESTIMACION	TAREA	ESTADO	POR
5 H	Creación de Tablas en la BDD	Por Hacer	KF
5 H	Creación de Procedimientos Almacenados	Por Hacer	KF
6 H	Pruebas BDD	Por Hacer	KF
6 H	Generación de Listas Administrables	Por Hacer	KF
8 H	Pruebas Funcionalidad	Por Hacer	KF

Tabla 7. Pila de Tareas H3: Administración de Parámetros

Fuente. Propia

TABLA 2.20
PILA DE TAREAS H5: Gestión de Equipos

ESTIMACION	TAREA	ESTADO	POR
5 H	Creación de Equipos	Por Hacer	KF
3 H	Agregar imágenes al crear	Por Hacer	KF
6 H	Búsqueda de Equipos	Por Hacer	KF
8 H	Edición y Eliminación de Equipos	Por Hacer	KF
8 H	Administración de Galería	Por Hacer	KF
10 H	Pruebas	Por Hacer	KF

Tabla 8. Pila de Tareas H3: Gestión de Equipos

Fuente. Propia

TABLA 2.21
PILA DE TAREAS H6: Gestión de Planes de Mantenimiento

ESTIMACION	TAREA	ESTADO	POR
30 H	Creación de Plantillas	Por Hacer	KF
24 H	Asignación de Planes	Por Hacer	KF
16 H	Administración de Plantillas	Por Hacer	KF
34 H	Consulta de Plantillas y Planes	Por Hacer	KF
16 H	Pruebas	Por Hacer	KF

Tabla 9. Pila de Tareas H6: Gestión de Planes de Mantenimiento

Fuente. Propia

TABLA 2.22
PILA DE TAREAS H7: Gestión de Mantenimientos no Planificados

ESTIMACION	TAREA	ESTADO	POR
28 H	Creación de Mantenimientos Correctivos	Por Hacer	KF
20 H	Búsqueda de Mantenimientos Correctivos	Por Hacer	KF
24H	Edición y Eliminación de Mantenimientos Correctivos	Por Hacer	KF
16 H	Pruebas	Por Hacer	KF

Tabla 10. PILA DE TAREAS H7: Gestión de Mantenimientos no Planificados

Fuente. Propia

TABLA 2.23
PILA DE TAREAS H8: Gestión de Revisiones Vehiculares

ESTIMACION	TAREA	ESTADO	POR
48 H	Generar a revisiones	Por Hacer	KF
20 H	Búsqueda de revisiones	Por Hacer	KF
20 H	Pruebas	Por Hacer	KF

Tabla 11. PILA DE TAREAS H8: Gestión de Revisiones Vehiculares

Fuente. Propia

TABLA 2.24
PILA DE TAREAS H9: Gestión de Rutas y Kilometrajes

ESTIMACION	TAREA	ESTADO	POR
32 H	Generar asignación de rutas	Por Hacer	KF
30 H	Búsqueda de rutas	Por Hacer	KF
32 H	Registro de Kilometrajes	Por Hacer	KF
30 H	Pruebas	Por Hacer	KF

Tabla 12. PILA DE TAREAS H9: Gestión de Rutas y Kilometrajes

Fuente. Propia

TABLA 2.25			
PILA DE TAREAS H10: Aprobación de Peticiones			
ESTIMACION	TAREA	ESTADO	POR
17 H	Aprobación de Planes de Mantenimiento	Por Hacer	KF
17 H	Aprobación de Planes de Mantenimientos Correctivos	Por Hacer	KF
14 H	Aprobación de Revisiones Vehiculares	Por Hacer	KF
14 H	Aprobación de Rutas	Por Hacer	KF
8 H	Pruebas	Por Hacer	KF

Tabla 13. PILA DE TAREAS H10: Aprobación de Peticiones

Fuente. Propia

TABLA 2.26			
PILA DE TAREAS H11: Inventario de Herramientas			
ESTIMACION	TAREA	ESTADO	POR
19 H	Agregar nuevos repuestos	Por Hacer	KF
18 H	Búsqueda de repuestos	Por Hacer	KF
15 H	Edición y Eliminación de repuestos	Por Hacer	KF
8 H	Pruebas	Por Hacer	KF

Tabla 14. PILA DE TAREAS H11: Inventario de Herramientas

Fuente. Propia

TABLA 2.27			
PILA DE TAREAS H12: Notificaciones			
ESTIMACION	TAREA	ESTADO	POR
24 H	Módulo de Notificaciones	Por Hacer	KF
16 H	Generar ordenes de trabajo	Por Hacer	KF
26 H	Alertas	Por Hacer	KF
16 H	Pruebas	Por Hacer	KF

Tabla 15. PILA DE TAREAS H12: Notificaciones y Reportes

Fuente. Propia

2.2. Planificación del Sprint

En esta sección después de realizar la reunión de planificación del Sprint (Sprint Plannig Meeting) se escoge las tareas que se van a desarrollar en cada sprint, también se indica las fechas de inicio y fin del sprint y la duración de cada etapa.

En la Figura 16 se detallan los 5 Sprint definidos las fechas y días en los que se van a desarrollar, así como el costo que tendría el desarrollo de cada uno.

Nombre Sprint	Fecha de Inicio	Fecha Final	Días presupuesto	Hombre Dias	Presupuesto
Totales:			105	420.0	\$10080.0
Seguridad y Administración General	01 Aug 2016	31 Aug 2016	22	88	\$2112.0
Equipos y Planes de Mantenimiento	01 Sep 2016	29 Sep 2016	21	84	\$2016.0
Rutas y Kilometrajes	03 Oct 2016	31 Oct 2016	20	80	\$1920.0
No Planificados y Revisiones	01 Nov 2016	30 Nov 2016	22	88	\$2112.0
Notificaciones, Alertas y Reportes	01 Dec 2016	29 Dec 2016	20	80	\$1920.0

Figura 16. Planificación del Sprint

Fuente. Sprintometer 6.54

2.2.1. Sprints

Cada Sprint tienen asignados a cada uno de ellos historias de usuario y tareas, en la Figura 17 se visualiza la estructura del Sprint que maneja el software Sprintometer 6.54 y se detalla lo siguiente:

1. La Fecha en la que inicio el trabajo en el sprint, el porcentaje de las horas de todo el proyecto y el número de horas de esa fecha que han sido hechas (en el caso de la tareas de tipo diseño, o análisis), codificadas (en el caso de la tareas de tipo codificación), probadas (en el caso de la tareas de tipo pruebas).
2. h/q el porcentaje o número de horas que se realizaron esa fecha/ el porcentaje o número de horas que faltan por hacer.
3. Las Historias que posee el Spring con sus respectivas tareas, el color de las tareas depende del tipo que se asignó, se pueden crear tipos diferentes de tareas pero por defecto tenemos 3: ■ Análisis o Diseño, ■ Codificación o Desarrollo, ■ Pruebas.
4. Asignado 1 es la persona a la que se asignó la tarea en este caso como solo una persona desarrolla se creó perfiles de la misma persona con diferente cargo en este caso: KathyAn (Análisis o Diseño), KathyCo (Codificación o Desarrollo), KathyPr (Pruebas). Hecho %: Las horas estimadas de cada tarea.

				Estimacion\Fecha:	May 11
				Hecho %:	-
			1	Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/440
				Codificado hoy/para hacer:	0/340
				Probado hoy/Para hacer:	0/90
3	Story 1	Tool		Hecho %:	-
	1	DESIGN		10	0/10
	2	DEVELOPMENT	4	10	0/10
	3	TESTING		5	0/5
	Story 2	Component		Hecho %:	-
	1	DEVELOPMENT		130	0/130
	2	TESTING		20	0/20

Figura 17. Estructura del Sprint

Fuente. Sprintometer 6.54

2.2.1.1. Seguridad y Administración general

En la Figura 16 se detalla el Sprint 1 que posee 4 historias de Usuario.

				Estimacion\Fecha:	Aug 01
				Hecho %:	-
				Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/184
				Codificado hoy/para hacer:	0/107
Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea	Asignado 1	Asignado 2	Probado hoy/Para hacer:	0/59
H1	Plataforma Tecnológica			Hecho %:	-
<u>1</u>	<i>Instalación y Configuración de las herramientas</i>	<i>Kathy An</i>		3	0/3
<u>2</u>	<i>Creación de los Modelos de BDD</i>	<i>Kathy An</i>		15	0/15
<u>3</u>	<i>Creación de los Procedimientos Almacenados</i>	<i>Kathy Co</i>		15	0/15
<u>4</u>	<i>Creación de la Base de Datos del Sistema</i>	<i>Kathy Co</i>		3	0/3
<u>5</u>	<i>Pruebas con la Base de Datos</i>	<i>Kathy Pr</i>		13	0/13
<u>6</u>	<i>Levantamiento del Entorno y Conexión a la BDD</i>	<i>Kathy Co</i>		5	0/5
<u>7</u>	<i>Pruebas de Integración</i>	<i>Kathy Pr</i>		8	0/8
H2	Seguridad: Usuarios y Roles			Hecho %:	-
<u>1</u>	<i>Login y autenticación de acceso</i>	<i>Kathy Co</i>		5	0/5
<u>2</u>	<i>Registro de datos personales y foto</i>	<i>Kathy Co</i>		6	0/6
<u>3</u>	<i>Creación de usuarios y roles</i>	<i>Kathy Co</i>		5	0/5
<u>4</u>	<i>Administración de usuarios y roles</i>	<i>Kathy Co</i>		8	0/8
<u>5</u>	<i>Métodos de Autenticación y Acceso</i>	<i>Kathy Co</i>		8	0/8
<u>6</u>	<i>Pruebas</i>	<i>Kathy Pr</i>		8	0/8
H3	Administración General del Sistema			Hecho %:	-
<u>1</u>	<i>Configuración de Parámetros Generales</i>	<i>Kathy Co</i>		8	0/8
<u>2</u>	<i>Configuración de Carpetas del Sistema</i>	<i>Kathy Co</i>		6	0/6
<u>3</u>	<i>Servicio de E-mail</i>	<i>Kathy Co</i>		8	0/8
<u>4</u>	<i>Servicio de SMS</i>	<i>Kathy Co</i>		8	0/8
<u>5</u>	<i>Configuración de los Respaldos de BDD</i>	<i>Kathy Co</i>		6	0/6
<u>6</u>	<i>Pruebas</i>	<i>Kathy Pr</i>		16	0/16
H4	Administración de Parámetros			Hecho %:	-
<u>1</u>	<i>Creación de Tablas en la BDD</i>	<i>Kathy Co</i>		5	0/5
<u>2</u>	<i>Creación de Procedimientos Almacenados</i>	<i>Kathy Co</i>		5	0/5
<u>3</u>	<i>Pruebas BDD</i>	<i>Kathy Pr</i>		6	0/6
<u>4</u>	<i>Generación de Listas Administrables</i>	<i>Kathy Co</i>		6	0/6
<u>5</u>	<i>Pruebas de Funcionalidad</i>	<i>Kathy Pr</i>		8	0/8

Figura 18. SPRINT 1: Seguridad y Administración General

Fuente. Sprintometer 6.54

2.2.1.2. Equipos y Planes de Mantenimiento

En la Figura 17 se detalla el Sprint 2 que posee 2 historias de Usuario.

				Estimacion\Fecha:	Sep 04
				Hecho %:	-
				Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/160
				Codificado hoy/para hacer:	0/144
Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea	Asignado 1	Asignado 2	Probado hoy/Para hacer:	0/16
H5	Gestión de Equipos			Hecho %:	-
<u>1</u>	Creación de Equipos	Kathy Co		5	0/5
<u>2</u>	Agregar imágenes al crear	Kathy Co		3	0/3
<u>3</u>	Búsqueda de Equipos	Kathy Co		6	0/6
<u>4</u>	Edición y Eliminación de Equipos	Kathy Co		8	0/8
<u>5</u>	Administración de Galería	Kathy Co		8	0/8
<u>6</u>	Pruebas	Kathy Co		10	0/10
H6	Gestión de Planes de Mantenimiento			Hecho %:	-
<u>1</u>	Creación de Plantillas	Kathy Co		30	0/30
<u>2</u>	Asignación de Planes	Kathy Co		24	0/24
<u>3</u>	Administración de Plantillas	Kathy Co		16	0/16
<u>4</u>	Consulta de Plantillas y Planes	Kathy Co		34	0/34
<u>5</u>	Pruebas	Kathy Pru		16	0/16

Figura 19. SPRINT 2: Equipos y Planes de Mantenimiento

Fuente. Sprintometer 6.54

2.2.1.3. Rutas y Kilometrajes

En la Figura 18 se detalla el Sprint 3 que posee 2 historias de Usuario.

				Estimacion\Fecha:	Oct 02
				Hecho %:	-
				Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/184
				Codificado hoy/para hacer:	0/146
Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea	Asignado 1	Asignado 2	Probado hoy/Para hacer:	0/38
H9	Gestión de Rutas y Kilometrajes			Hecho %:	-
<u>1</u>	Generar asignación de rutas	Kathy Co		32	0/32
<u>2</u>	Búsqueda de rutas	Kathy Co		30	0/30
<u>3</u>	Registro de Kilometrajes	Kathy Co		32	0/32
<u>4</u>	Pruebas	Kathy Pr		30	0/30
H11	Inventario de Herramientas			Hecho %:	-
<u>1</u>	Agregar nuevos repuestos	Kathy Co		19	0/19
<u>2</u>	Búsqueda de repuestos	Kathy Co		18	0/18
<u>3</u>	Edición y Eliminación de repuestos	Kathy Co		15	0/15
<u>4</u>	Pruebas	Kathy Pr		8	0/8

Figura 20. SPRINT 3: Rutas y Kilometrajes

Fuente. Sprintometer 6.54

2.2.1.4. No Planificados y Revisiones

En la Figura 19 se detalla el Sprint 4 que posee 2 historias de Usuario.

				Estimacion\Fecha:	Nov 02
				Hecho %:	-
				Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/176
				Codificado hoy/para hacer:	0/160
Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea	Asignado 1	Asignado 2	Probado hoy/Para hacer:	0/16
H8	Gestión de Revisiones Vehiculares			Hecho %:	-
<u>1</u>	<i>Generar a revisiones</i>	<i>Ka Co</i>		48	0/48
<u>2</u>	<i>Búsqueda de revisiones</i>	<i>Ka Co</i>		20	0/20
<u>3</u>	<i>Pruebas</i>	<i>Ka Pr</i>		20	0/20
H7	Gestión de Mantenimientos No Planificados			Hecho %:	-
<u>1</u>	<i>Creación de Mantenimientos Correctivos</i>	<i>Ka Co</i>		28	0/28
<u>2</u>	<i>Búsqueda de Mantenimientos Correctivos</i>	<i>Ka Co</i>		20	0/20
<u>3</u>	<i>Edición y Eliminación de Mantenimientos Correctivos</i>	<i>Ka Co</i>		24	0/24
<u>4</u>	<i>Pruebas</i>	<i>Ka Pr</i>		16	0/16

Figura 21. SPRINT 4: No Planificados y Revisiones

Fuente. Sprintometer 6.54

2.2.1.5. Notificaciones, Alertas y Reportes

En la Figura 20 se detalla el Sprint 5 que posee 2 historias de Usuario.

				Estimacion\Fecha:	Dec 04
				Hecho %:	-
				Codificado %:	-
				Probado %:	-
				Hecho hoy/Para hacer:	0/152
				Codificado hoy/para hacer:	0/128
Nº Historia, Nº tarea	Nombre Historia, Nombre Tarea	Asignado 1	Asignado 2	Probado hoy/Para hacer:	0/24
H12	Notificaciones y Reportes			Hecho %:	-
<u>1</u>	<i>Módulo de Notificaciones</i>	<i>Ka Co</i>		24	0/24
<u>2</u>	<i>Generar ordenes de trabajo</i>	<i>Ka Co</i>		16	0/16
<u>3</u>	<i>Construir Reportes de la información primordial</i>	<i>Ka Co</i>		26	0/26
<u>4</u>	<i>Pruebas</i>	<i>Ka Pr</i>		16	0/16
H10	Aprobación de Peticiones			Hecho %:	-
<u>1</u>	<i>Aprobación de Planes de Mantenimiento</i>	<i>Ka Co</i>		17	0/17
<u>2</u>	<i>Aprobación de Planes de Mantenimientos Correctivos</i>	<i>Ka Co</i>		17	0/17
<u>3</u>	<i>Aprobación de Revisiones Vehiculares</i>	<i>Ka Co</i>		14	0/14
<u>4</u>	<i>Aprobación de Rutas</i>	<i>Ka Co</i>		14	0/14
<u>5</u>	<i>Pruebas</i>	<i>Ka Pr</i>		8	0/8

Figura 22. SPRINT 5: Notificaciones, Alertas y Reportes

Fuente. Sprintometer 6.54

2.3. Desarrollo de los Sprint

Se busca obtener una aplicación con un alto nivel de portabilidad, de fácil distribución y con un alto nivel de reusabilidad. En sus etapas iniciales, la aplicación será instalada en un servidor local de prueba, pero se busca obtener una pieza de software que se pueda instalar en cualquier servidor, con un mínimo de esfuerzo y con el mejor performance posible.

2.3.1. Sprint 1 Seguridad y Administración General

En este Sprint se crea todo lo necesario para que un Sistema cualquiera pueda funcionar primero se prepara el entorno de desarrollo, se realiza un análisis de las historias de usuario para crear la capa de datos en su totalidad, después se realiza el **MODULO DE SEGURIDAD** del sistema que incluye la administración de usuarios, roles y métodos de autenticación, también se crea los servicios que el sistema posee como servicios de E-mail, SMS, Respaldos etc. Finalmente se crea el **MODULO DE ADMINISTRACION GENERAL DEL SISTEMA** que incluye la administración de parámetros y configuraciones para que la aplicación sea totalmente administrable por el usuario.

2.3.1.1. Análisis y Diseño

En la Tabla 16 se muestra los tipos o roles de usuario que tiene el sistema y las actividades que puede realizar cada uno.

TIPO	FORMACION	HABILIDADES	ACTIVIDADES
Administrador	Ingeniero Automotriz o Ingeniero Informático	Manejo de aplicaciones Web, Conocimiento experto del negocio	Realiza todas las actividades de todos los tipos incluidas las siguientes: - Administración de Usuarios y Roles del sistema. - Administración y Configuración de parámetros generales del sistema, respaldos y servicios de Email y SMS
Supervisor	Ingeniero Automotriz	Comunicación y liderazgo, toma de decisiones.	- Aprobación de Planes de Mantenimiento - Aprobación de Accidentes - Aprobación de Revisiones Vehiculares
Técnico	Ingeniero Automotriz	Conocimiento experto del negocio	- Configuración de Parámetros del sistema - Administración de Equipos - Administración de Plantillas y Planes de Mantenimiento - Administración de Revisiones Vehiculares - Administración de Herramientas - Acceso a notificaciones y Disponibilidad
Secretaria	Instrucción Básica	Atención al cliente	- Administración de Rutas - Asignación de Kilometrajes

Tabla 16. Roles o Tipos de Usuario

Fuente. Propia

En la Figura 23 se muestra el modelo de datos inicial del sistema, este modelo fue el que se diseñó inicialmente, en el [ANEXO B](#) se encuentra el modelo de datos final.

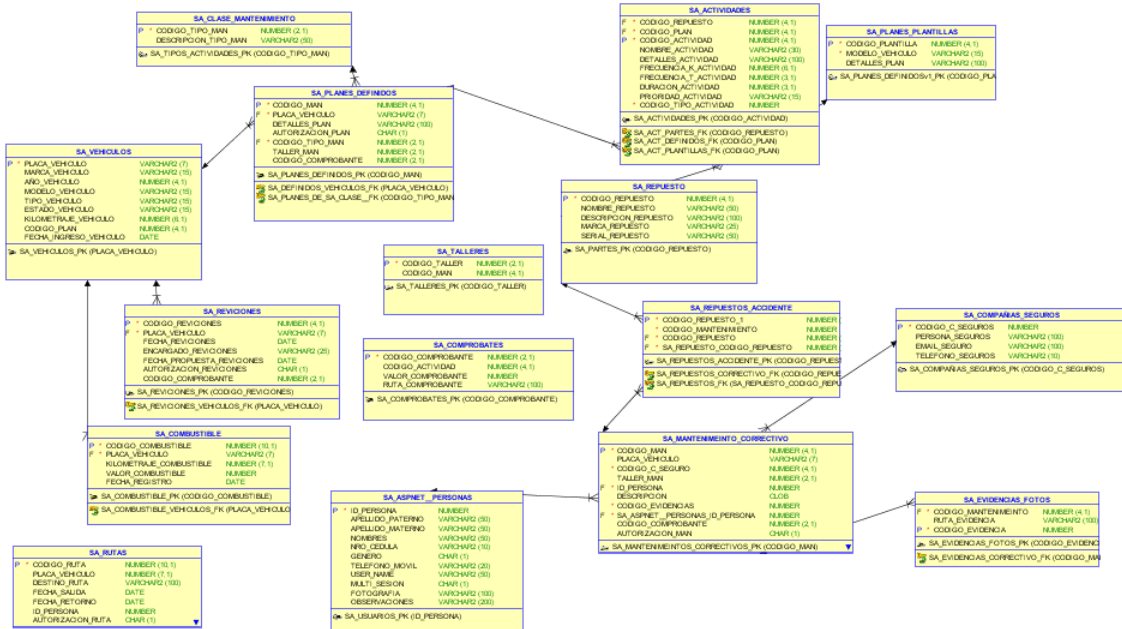


Figura 23. Modelo de Datos

Fuente. Propia

2.3.1.2. Codificación y Pruebas de Desarrollador

Se crea la base de datos **mantenimiento** según el modelo de datos diseñado, también los procedimientos almacenados de consulta, creación, modificación y eliminación de cada tabla.

2.3.1.3. Revisión del Sprint

- Se verifica la finalización de todas las tareas planificadas a través de las diferentes pruebas de validación de las historias de usuario y nacen nuevos requerimientos.
- Se ve la necesidad de agregar más tablas a la base de datos y los procedimientos almacenados necesarios para la configuración de parámetros generales, notificaciones, tablas temporales para realizar procesos y de historiales.
- Se agrega en la administración de parámetros la funcionalidad de Configuración de alertas de planes de mantenimiento donde se muestra el régimen actual tanto de tiempo como de kilometraje y se da la opción de editar estos datos.

- Se completa y verifica los nuevos requerimientos con el usuario final por lo que se concluye correctamente este sprint.

2.3.1.4. Retrospectiva

Se crean las siguientes tablas:

SA_TIPOS_ACTIVIDADES

SA_ENCARGADOS

SA_LOCALIZACION_VEHICULO

SA_PARTES

SA_PROVEEDOR_VEHICULO

SA_TIPO_VEHICULO

SA_USO_VEHICULO

2.3.1.5. Incremento del Producto

En la Figura 24 se muestra el acceso principal y el Login del sistema.



Figura 24. Pantalla Login

Fuente. Propia

En la Figura 25 se muestra el menú de la administración general del sistema y la interfaz de administración del servicio de E-mail.

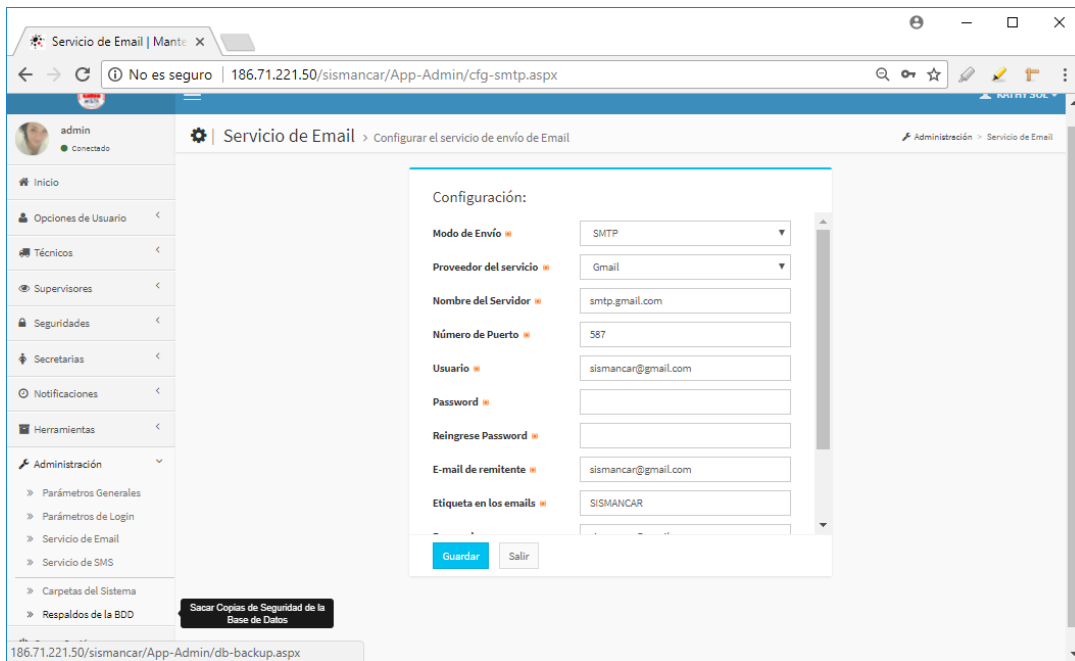


Figura 25. Administración General del Sistema

Fuente. Propia

En la Figura 26 se muestra la Administración de Parámetros

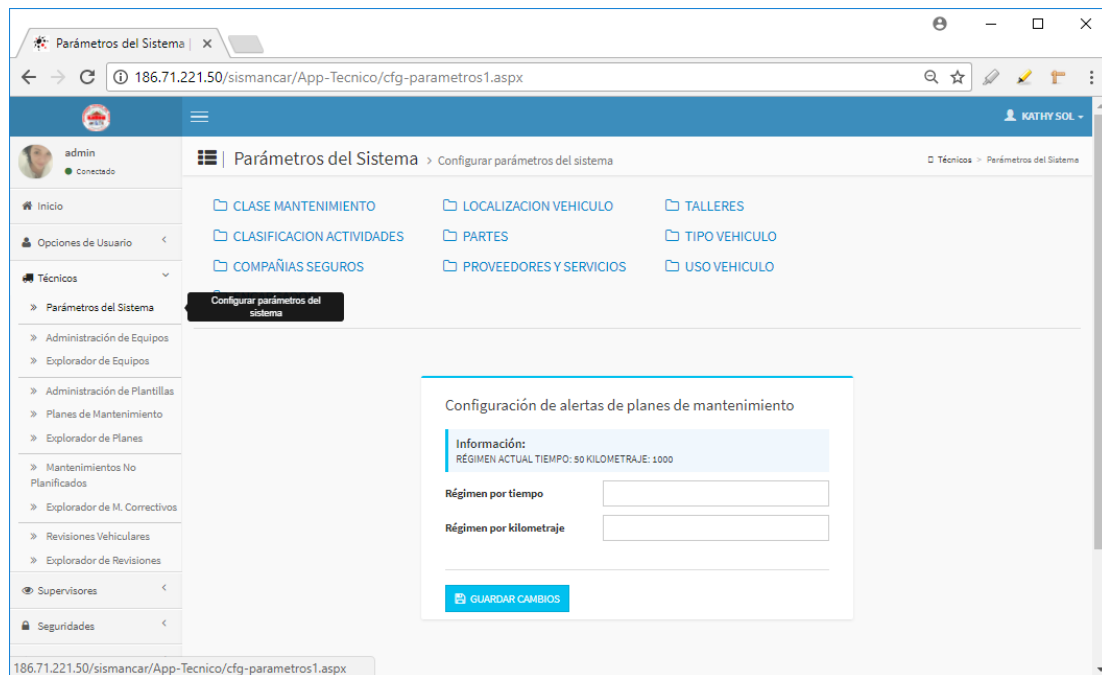


Figura 26. Administración de Parametros

Fuente. Propia

2.3.2. Sprint 2 Equipos y Planes de Mantenimiento

En este Sprint se crea el **MODULO DE GESTON DE EQUIPOS** que incluye el CRUD de los vehículos y la administración de las imágenes que posee cada vehículo, también se crea el **MODULO DE GESTION DE PLANES DE MANTENIMIENTO** que incluye el CRUD de plantillas de Mantenimiento, la asignación de planes de mantenimientos y búsqueda de ambos.

2.3.2.1. Codificación y Pruebas de Desarrollador

Se crean los servicios web:

- a) *obtenerMaximoFotos* (): Obtiene el ID_FOTOS máximo de la tabla SA_FOTOS_VEHICULOS.
- b) *insertarFotosVehiculos* (placa, ruta, nFoto): Obtiene el ID_FOTOS máximo de la tabla SA_FOTOS_VEHICULOS.
- c) *insertarVehiculo* (vehiculo): consume el procedimiento almacenado insert_vehiculo, el cual agrega un vehiculo a la tabla SA_VEHICULOS.
- d) *obtenerFotos* (nFoto, placa): consume el procedimiento almacenado insert_fotos_vehiculos, que añade un registro a la tabla SA_FOTO_VEHICULOS.

2.3.2.2. Revisión del Sprint

- Se verifica la finalización de todas las tareas planificadas a través de las diferentes pruebas de validación de las historias de usuario y al no existir nuevos requerimientos se da por concluido este sprint.

2.3.2.3. Incremento del Producto

En la Figura 27 se muestra el módulo de administración de equipos

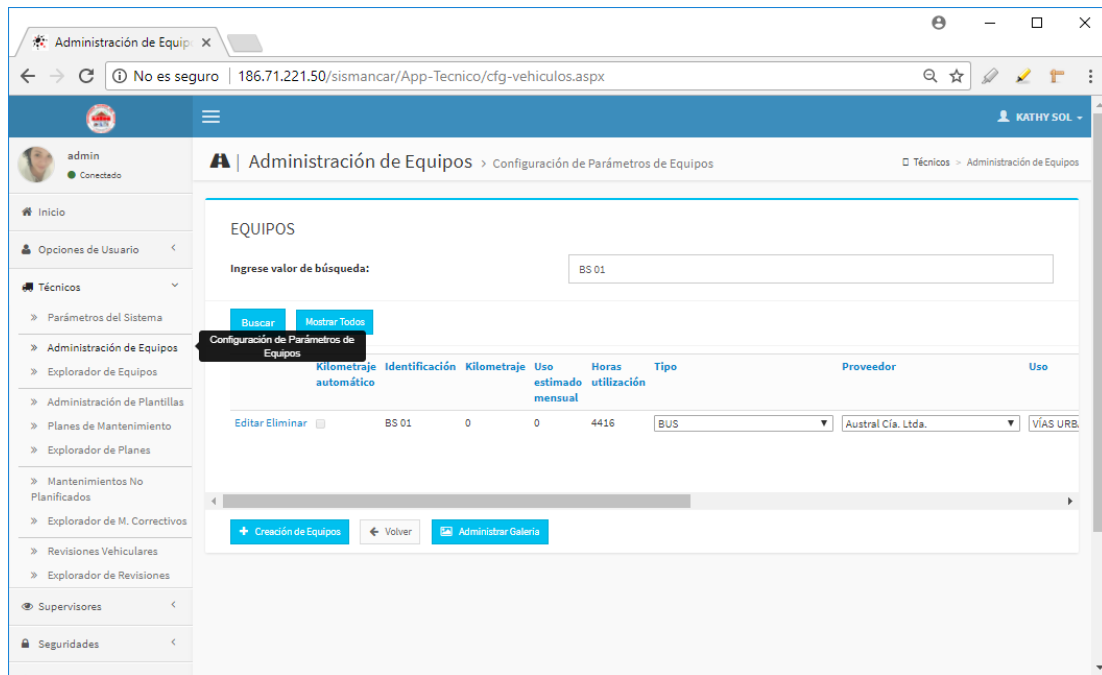


Figura 27. Administración de Equipos

Fuente. Propia

En la Figura 28 se muestra el módulo de administración de Plantillas y Asignación de Planes de Mantenimiento.

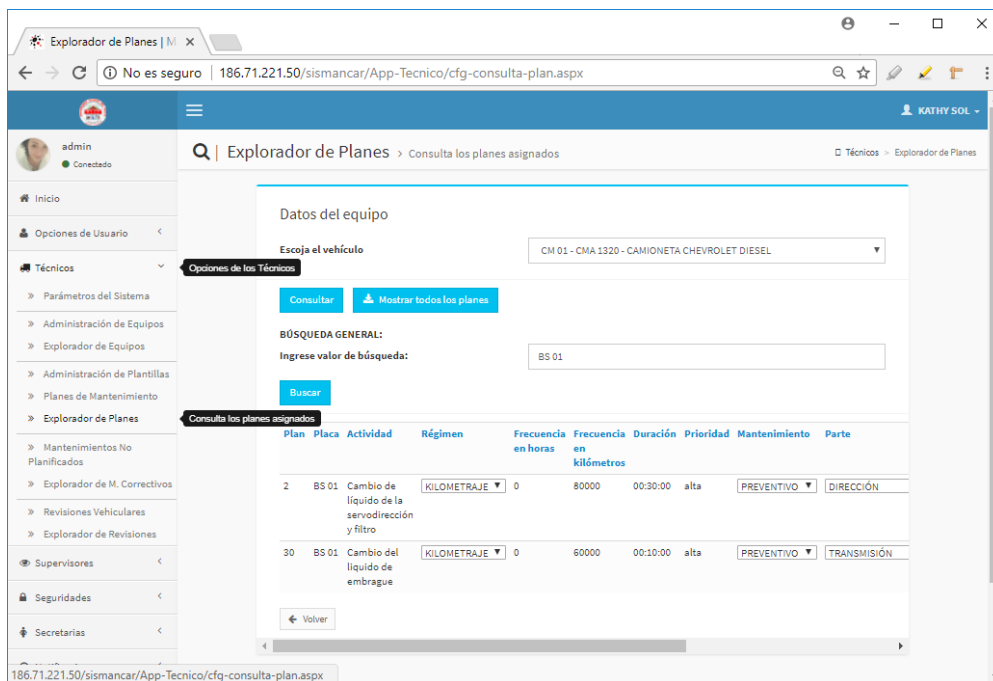


Figura 28. Plantillas y Asignación de Planes de Mantenimiento

Fuente. Propia

2.3.3. Sprint 3 Rutas y Kilometrajes

En este Sprint se crea el **MÓDULO DE GESTIÓN DE RUTAS Y KILOMETRAJES** que incluye la asignación y búsqueda de rutas y el registro de kilometrajes. También se desarrolla el CRUD de los repuestos que se necesitan a la hora de realizar los mantenimientos ya sean correctivos o preventivos.

2.3.3.1. Codificación y Pruebas de Desarrollador

Se crean los servicios web:

a) *obtenerDetalleVehiculo* (identificación): consume el procedimiento almacenado `select_todo_vehiculos`, el cual obtiene toda la información detallada de un vehículo de la tabla `SA_VEHICULOS`.

b) *insertarKilometraje* (kilometraje): agrega un nuevo registro en la tabla `SA_KILOMETRAJES`.

c) *actualizarKilometraje* (placa, kilometraje): realiza un `update KILOMETRAJE_VEHICULO` de la tabla `SA_VEHICULOS` dependiendo del número de identificación del vehículo.

d) *obtenerKilometrajeActual* (placa): obtiene el `KILOMETRAJE_VEHICULO` de la tabla `SA_VEHICULOS` dependiendo del número de identificación del vehículo.

2.3.3.2. Revisión del Sprint

- Se verifica la finalización de todas las tareas planificadas a través de las diferentes pruebas de validación de las historias de usuario y nacen nuevos requerimientos.
- Se ve la necesidad de agregar la funcionalidad de aprobación de rutas para el tipo de usuario supervisor ya que estas serán válidas solo si un jefe inmediato lo aprueba.
- Este nuevo requerimiento se desarrollara en el sprint 5 por lo que se concluye correctamente este sprint.

2.3.3.3. Incremento del Producto

En la Figura 29 se muestra la interfaz de asignación de rutas.

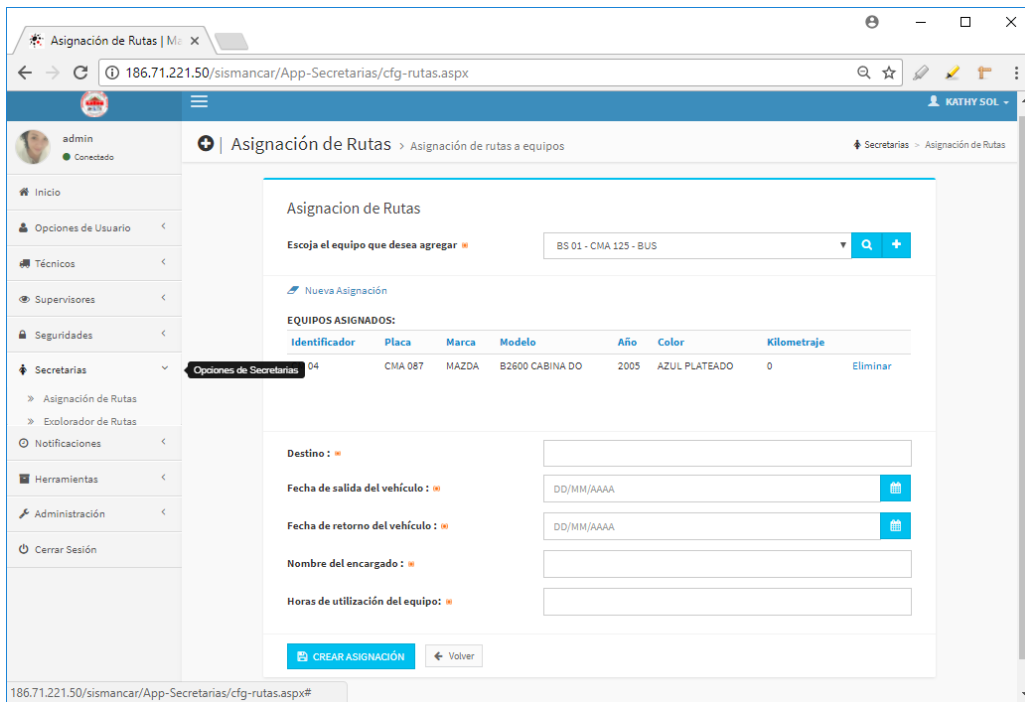


Figura 29. Asignacion de Rutas

Fuente. Propia

En la Figura 30 se muestra la administración de repuestos.

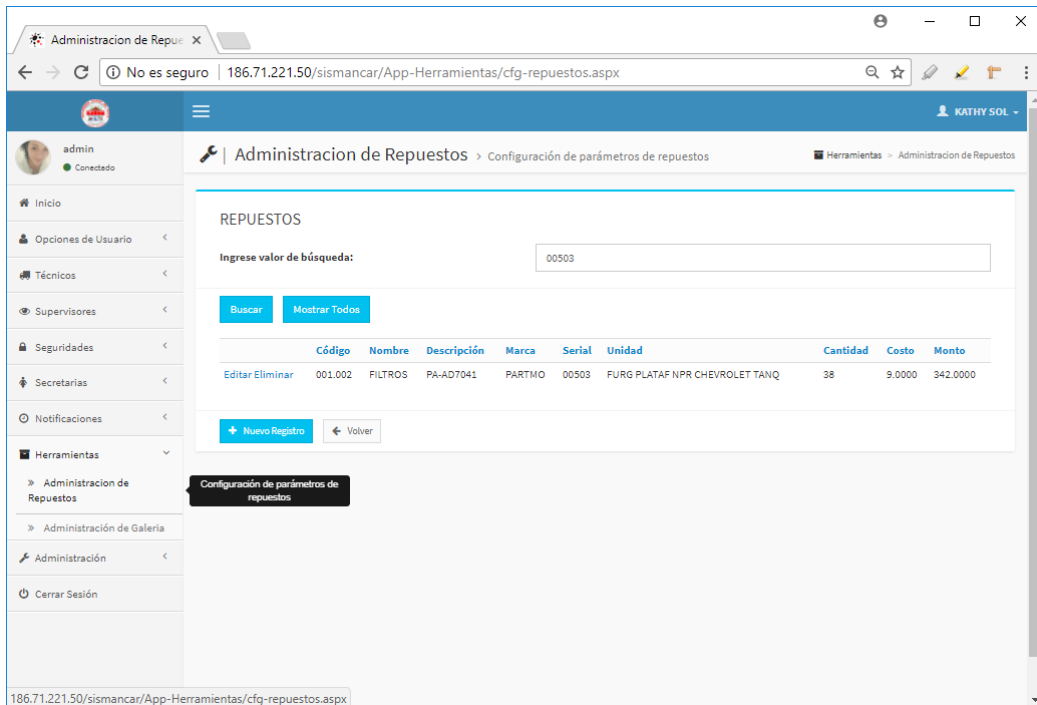


Figura 30. Administracion de Repuestos

Fuente. Propia

2.3.4. Sprint 4 Mantenimientos no Planificados y Revisiones Vehiculares

En este Sprint se crea el **MÓDULO DE GESTIÓN DE MANTENIMIENTOS NO PLANIFICADOS** que incluye el CRUD de los Accidentes. También se creó el **MÓDULO DE GESTIÓN DE REVISIONES VEHICULARES** donde se añade la funcionalidad de creación y búsqueda de revisiones.

2.3.4.1. Codificación y Pruebas de Desarrollador

Se crean los servicios web:

- a) *copiarVehiculosTemporales* (placa): almacena registros en la tabla SA_VEHICULOS_TEMPORALES.
- b) *eliminarVehiculosTemporales* (): consume el procedimiento almacenado *delete_vehiculo_temporales*, que elimina registros en la tabla SA_VEHICULOS_TEMPORALES.
- c) *insertarMantenimientoCorrectivo* (accidente): consume el procedimiento almacenado *insert_mantenimiento_correctivo*, agrega un nuevo registro en la tabla SA_MANTENIMEINTO_CORRECTIVO.
- d) *insertarRevision* (revision): consume el procedimiento almacenado *insert_revicion*, agrega un nuevo registro en la tabla SA_REVICIONES.

2.3.4.2. Revisión del Sprint

- Se verifica la finalización de todas las tareas planificadas a través de las diferentes pruebas de validación de las historias de usuario y nacen nuevos requerimientos.
- Se ve la necesidad de agregar la administración de las evidencias que se vinculan al mantenimiento correctivo ya que una vez signan si se desea cambiar los documentos no es posible
- Se agrega en la asignación de planes de mantenimiento no planificados la administración de evidencias que permite seleccionar el vehículo deseado y editar las evidencias.
- Se completa y verifica los nuevos requerimientos con el usuario final por lo que se concluye correctamente este sprint.

2.3.4.3. Retrospectiva

Se crean los servicios web:

a) obtenerMaximaEvidencia (): obtiene el máximo de CODIGO_HOJA de la tabla SA_HOJA_ACCIDENTES.

b) insertarEvidencia (evidencia): consume el procedimiento almacenado insert_evidencia_fotos, agrega un nuevo registro en la tabla SA_EVIDENCIAS_FOTOS.

2.3.4.4. Incremento del Producto

En la Figura 31 se muestra la interfaz de asignación de mantenimientos no planificados

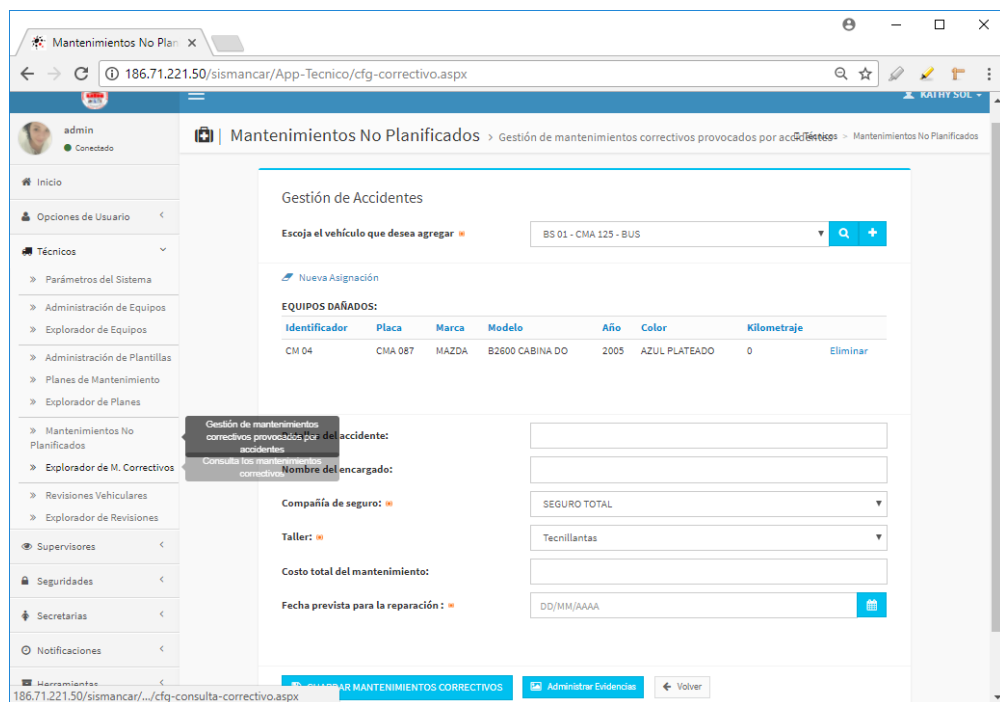


Figura 31. Asignacion de Accidentes

Fuente. Propia

En la Figura 32 se muestra la asignación de revisiones vehiculares

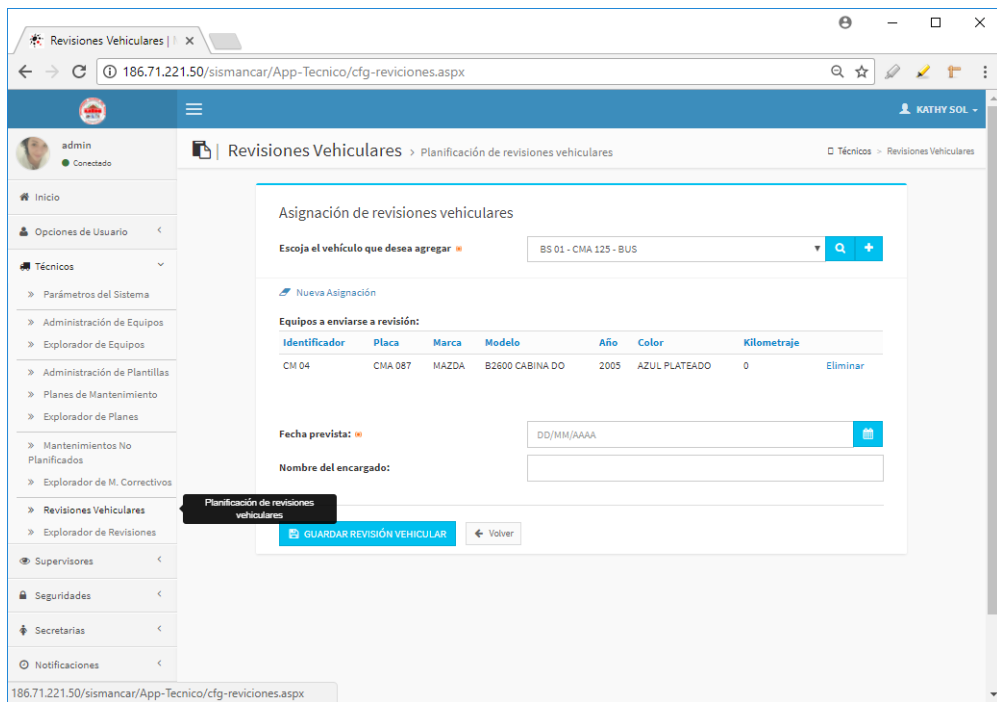


Figura 32. Asignación de Revisiones

Fuente. Propia

En la Figura 33 se muestra la administración de Evidencias

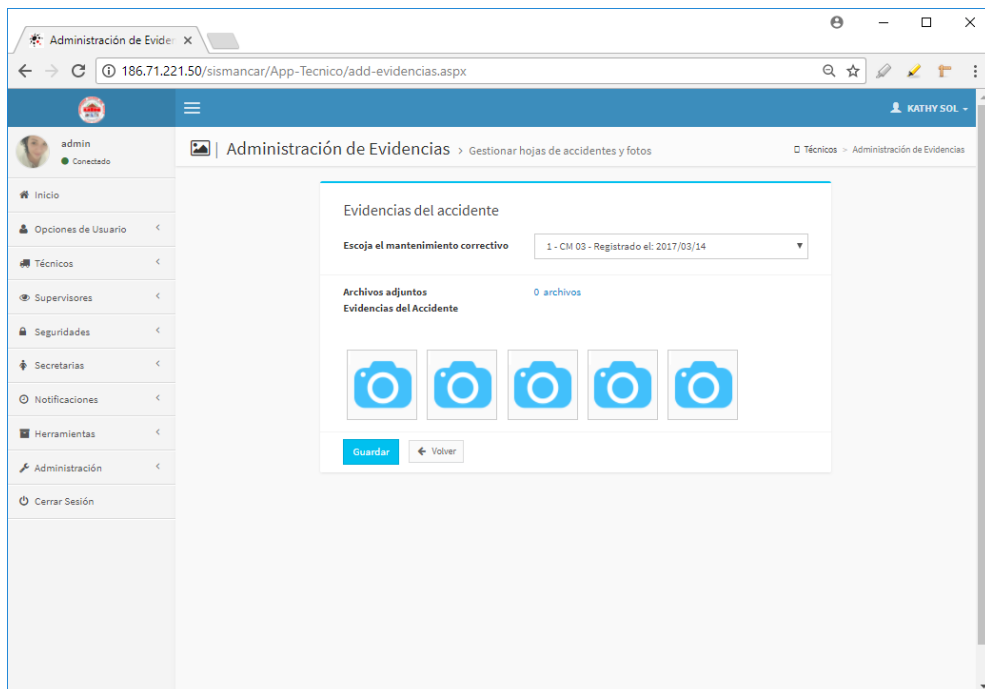


Figura 33. Administración de Evidencias

Fuente. Propia

2.3.5. Sprint 5 Notificaciones Alertas y Reportes

En este Sprint se crea el **MÓDULO DE NOTIFICACIONES Y REPORTE**s que incluyen las notificaciones de planes de mantenimiento, revisiones vehiculares y accidentes, así como la generación de la orden de trabajo y los reportes de la información más relevante del sistema, también se realiza la aprobación de peticiones de cada módulo; en esta aprobaciones se incluye la funcionalidad de aprobación de rutas definida en el tercer sprint.

2.3.5.1. Análisis y Diseño

TABLA 2.13 APROBACIÓN DE RUTAS	
H13	
Para que a un vehículo se le pueda asignar una ruta es necesario que un supervisor apruebe dichas rutas ya que la generación de estas le compete a la secretaria, por lo tanto se crea agrega esta funcionalidad al módulo de aprobación de peticiones al cual solo los supervisores tendrán acceso donde se encargará de aprobar o rechazar rutas	
Estimación:	20 H
Prioridad: 3	Dependencias: H1, H2, H4, H5, H6, H7, H8, H10
<ul style="list-style-type: none">• Aprobar y Rechazar Rutas	

Tabla 17. Historia de Usuario: Aprobación de Rutas

Fuente. Propia

2.3.5.2. Codificación y Pruebas de Desarrollador

Se crean los servicios web:

- insertarHistorial (historial): consume el procedimiento almacenado insert_historial, agrega un nuevo registro en la tabla SA_HISTORIAL_SUPERVISOR.
- actualizarEstadoAccidente (valor, id): consume el procedimiento almacenado change_estado_correctivo, que actualiza el estado de REVICION_MAN de la tabla SA_HISTORIAL_SUPERVISOR.

c) actualizarEstadoPlanMantenimiento (valor, id): consume el procedimiento almacenado change_estado_correctivo, que actualiza el estado de REVICION_PLAN de la tabla SA_PLAN.

d) actualizarEstadoRevision (valor, id): consume el procedimiento almacenado change_estado_revision, que actualiza el estado de REVICION_REVICION de la tabla SA_REVICIONES.

e) actualizarEstadoRuta (valor, id): consume el procedimiento almacenado change_estado_rutas, que actualiza el estado de REVICION_RUTA de la tabla SA_RUTAS.

2.3.5.3. Revisión del Sprint

- Se verifica la finalización de todas las tareas planificadas a través de las diferentes pruebas de validación de las historias de usuario y al no existir nuevos requerimientos se da por concluido este sprint.

2.3.5.4. Incremento del Producto

En la Figura 34 se muestra la interfaz de la aprobación de peticiones.

APROBAR	RECHAZAR	Registro	Plantilla	Equipo	Nombre	Actividad	Régimen	Frecuencia en horas	Frecuencia en kilometraje	Mantenimiento de Arranque
		7-12-2017	199	CM 03	Inspección general de la dirección y sus vínculos	k	0	5000	250000	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	6-20-2017	2	CM 08	Cambio de liquido de la servodirección y filtro	k	0	80000	70000	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	6-20-2017	3	CM 08	Inspección del nivel del liquido	k	0	5000	1500	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	6-01-2017	2	CM 08	Cambio de liquido de la	k	0	1000	80000	

Registro	Plantilla	Equipo	Nombre	Actividad	Régimen	Frecuencia en horas	Frecuencia en kilometraje	Duración	Mantenimiento de Arranque	Mantenimiento Previsto
CM 08			Cambio de liquido de la		k	0	80000	00:30:00	70000	150000

Figura 34. Aprobación de peticiones

Fuente. Propia

CAPÍTULO 3

Resultados

3.1. Resultados Obtenidos

Cabe destacar que el proceso de implementación y constatación de resultados, el cual es posterior al desarrollo de SISMANCAR, se encuentra documentado en la tesis “IMPLEMENTACIÓN DE UN SISTEMA DE MANTENIMIENTO TOTAL PRODUCTIVO EN EL GAD MUNICIPAL DE SAN GABRIEL” (Sánchez, 2017) trabajo que se toma como referencia para la validación de los resultados a mencionar.

Se implementa el sistema de mantenimiento automotriz denominado SISMANCAR en el parque automotor del GAD Municipal de Montufar (San Gabriel), obteniendo los siguientes resultados.

Automatizar los procesos administrativos y automotrices que intervienen en el flujo normal del departamento de transporte. Para ello se recolecto información sobre: las tareas que se realizan diariamente, personas involucradas, inventario de 34 vehículos y maquinaria pesada, inventario de repuestos, organización de plantillas de mantenimiento, asignación del planes de mantenimiento, registro de kilometrajes, asignación de rutas a los vehículos, contingencia de accidentes, etc.

El cumplimiento de todos los requisitos funcionales de prioridad alta establecidos, gracias a una excelente ingeniería de requisitos, seguimiento del proyecto constante y la participación total de los StakeHolders. Todo esto se vio posible ya que la metodología de desarrollo SCRUM propicia agilidad, fácil organización y control en el desarrollo, retroalimentación por parte de los usuarios en cada etapa concluida.

Una solución de software con beneficios propios de MBI3.0 & EDRA los cuales son:

- Gestión de usuarios, roles y permisos a todo el sistema.
- Parametrización y administración del sistema, accesible al usuario final.
- Escalabilidad.
- Alta compatibilidad de componentes

Al implementar SISMANCAR en el GAD de Montufar y tras estar en un periodo de pruebas de 6 meses, se verifica un mejoramiento notable en el proceso de logística del departamento el cual se detallara a continuación en el análisis de impactos.

3.2. Análisis de Impactos

Con SISMANCAR se convierte en realidad la sistematización y fortalecimiento de uno de los procesos más importantes para el beneficio de la institución, como lo es la logística del parque automotor.

(Sánchez, 2017) Afirma que a principal debilidad en los procesos administrativos y operativos del área de mantenimiento automotriz es: La falta de planificar, monitorear y control de actividades que se realizar exitosamente con la implementación del sistema de mantenimiento automotriz SISMANCAR, luego de su desarrollo e implementación, se destacan los siguientes aspectos positivos, que mejoran en gran medida dichas actividades.

El sistema permitió optimizar el tiempo en todas las actividades de administración ya que se automatizaron los procesos y se cuenta con el acceso a información inmediata a través de búsquedas en catálogos de vehículos, repuestos, rutas, etc.

También se redujo el tiempo y esfuerzo de los usuarios en movilización, ya que el sistema al estar en la web tiene accesibilidad a la información, desde cualquier lugar sin necesidad de trasladarse al GAD.

Se obtuvo transparencia en los procesos, gracias al nivel de seguridad que maneja, el sistema y toda actividad, ya que se cuenta con la supervisión de un jefe inmediato en todas las actividades que se planea realizar.

3.2.1. Impacto Tecnológico

Gracias a los beneficios que trajo consigo esta iniciativa, el GAD abre las puertas para aplicar soluciones informáticas en la mayoría de sus procesos, por lo que la institución pasa a una etapa de transformación tecnológica y gracias a la Arquitectura EDRA que se aplica esa transformación será mucho más sencilla.

3.2.2. Impacto Económico

Se reducen los costos de operación debido a que con la correcta planificación de las revisiones vehiculares y trabajos en talleres por mantenimientos correctivos, resulta más barato enviar varios vehículos a un mismo lugar.

La implementación del sistema llevado a cabo en la tesis antes mencionada (Sánchez, 2017) el autor afirma que el análisis de reducción de costos en el parque automotor en un periodo de seis meses, determinó que el proyecto de estudio e implementación del sistema de mantenimiento automotriz SISMANCAR en el GAD Municipal de la Ciudad de San

Gabriel, es de beneficio para los intereses de la Institución. Se obtuvo una reducción del 13.11% del presupuesto semestral, mismo que supera el 5% del objetivo plateado.

3.2.3. Impacto Ambiental

Anteriormente se documentaba todos los procesos que se llevaban a cabo en el parque automotor, por lo que toda esta información se debe imprimir y luego almacenar ordenadamente, estas actividades implican a que el GAD disponga de recursos varios como espacio de almacenamiento y hojas de papel para poder imprimir toda la información. Lo mencionado en este párrafo implica afectación al medio ambiente, ya que el consumo excesivo de papel implica su extracción y por ende un fuerte impacto ambiental.

CONCLUSIONES

- Los resultados de SISMANCAR aportan notablemente con el mejoramiento de la logística de las actividades del parque automotor de GAD Municipal de Montufar, ya que estos van de la mano con el desarrollo tecnológico, además el aplicativo brinda la disponibilidad inmediata de la información, seguridad en los datos y calidad de servicio a los usuarios.
- EDRA en principios es una metodología bastante rígida orientada hacia el tipo de proceso de la Banca, por lo que para solventar ese problema se integró con MBI que hace mucho énfasis en la utilización de las mejores prácticas de Microsoft Corporation, a esa fusión se la denomina como MBI 3.0 que combina lo mejor de las 2 metodologías.
- Al realizar un diseño con EDRA & MBI3.0 se tiene un proyecto con la capacidad de integrarse con nuevos desarrollos, que sea flexible, fiable y adaptable a las necesidades del GAD, también rapidez para producir y empezar a funcionar, y la capacidad de aislar cambios generales de la aplicación de la lógica de negocio.
- Con la utilización de la metodología SCRUM se aceleró el proceso de desarrollo, ya que el marco de trabajo que utiliza es muy sencillo y fácil de implementar, se basa en el trabajo en equipo por lo que hubo la participación constante de los usuarios y el incremento constante del producto.
- En la finalización de cada Sprint se realizaron pruebas de integración de todas las funcionalidades para que se vaya generando un incremento correctamente, en la retrospectiva del sprint siempre se generaban nuevos requerimientos que fueron solventados dentro de los Sprint posteriores.
- Utilizar Herramientas de Microsoft Corporation propone mayor facilidad al momento del desarrollo con bloques de código reutilizable, una comunidad bastante amplia que la respalda, la integración entre herramientas mucho más sencilla y una capaz de crear aplicaciones de base de datos seguras y contrarrestar las amenazas.

RECOMENDACIONES

- Se debería implementar este sistema en las organizaciones donde existan parques automotores especialmente en las entidades públicas, ya que daría soluciones a muchas problemáticas especialmente a gastos por daños y gastos injustificados.
- Para el seguimiento de una metodología de desarrollo es necesario un software que te permita llevar el control y seguimiento del avance del el proyecto.
- Para levantar requerimientos de procesos bastante complejos o ajenos a tu campo de especialización es recomendable asesorarse de un experto en el tema, que te guie desde el levantamiento de requisitos, y durante todo el desarrollo.
- No es propicio utilizar una metodología que solo se centra a un caso específico de proceso o ámbito.

REFERENCIAS

- Arias, M. (2014). Webs Responsivas. Responsive Design con Bootstrap. España: IT Campus Academy.
- Autores, V. (2016). Manual. Instalación y configuración del software de servidor Web (UF1271). Madrid: CEP S.L.
- Baca, G. (2015). Proyectos de sistemas de información. México DF.: Patria.
- Ceballos, J. (2017). Visual Basic.NET. Curso de Programación. Ra-Ma.
- Cuello, J., & Vittone, J. (2013). Diseñando apps para móviles. Catalina Duque Giraldo.
- Debrauwer, L. (2013). Patrones de diseño de java. Barcelona: Ediciones ENI.
- Deman, T., Elmaleh, F., Neil, S., & Van Jones, M. (2014). Windows Server 2012 R2: Administración avanzada. Barcelona: ENI.
- Domínguez Mateos, F., Paredes Velasco, M., & Santacruz Valencia, L. P. (2014). Programación multimedia y dispositivos móviles. Madrid: RA-MA.
- Fagas, F. (2007). MICROSOFT. Obtenido de <https://msdn.microsoft.com/es-es/library/bb972204.aspx#XSLTsection127121120120>
- Foster, R. (2013). CodeIgniter 2 Cookbook. Livery Place: Packt Publishing Ltd.
- Gabillaud, J. (2015). SQL Server 2014 Administración de una base de datos transaccional con SQL Server Management Studio. Barcelona: ENI.
- Green, B., & Seshadri, S. (2013). AngularJS. Canada: O'Neill Media.
- Gutierrez, A. (2013). Framework de desarrollo. Obtenido de <http://jordisan.net/blog/2006/que-es-un-framework/>
- ITSON. (2014). Concepto de Sistema de Información. Obtenido de http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm
- Lerma, R., Murcia, J., & Mifsud, E. (2013). Aplicaciones Web. McGraw-Hill: Madrid.

Lezama, I. H. (Diciembre de 2014). entorno cliente/servidor. Obtenido de <http://es.slideshare.net/IzraelHdzLezama/apuntes-de-entorno-cliente-servidor-iii-42201097>

Lobato, F., & Villagr a, F. (2010). Gesti n log stica y comercial. Madrid: Macmillan Iberia, S.A.

L pez, B. (2013). Estructuras de datos orientadas a objetos. M xico D.F.: Alfaomega.

Medina, S. (2015). SQL Server 2014 : Soluciones pr cticas de administraci n. RA-MA.

Miguel, A. (2014). Webs Responsivas. Responsive Design con Bootstrap. Espa a: ItCampus Academy.

MINEDUC. (2015). Evaluaci n docente. Obtenido de <http://www.conocimiento.gob.ec/uce-expide-reglamento-de-evaluacion-integral-del-desempeno-docente/>

Moreno, J. C., & Gonz lez, M. (2014). Sistemas inform ticos y redes locales. Madrid: RA-MA.

Mu oz, V. J. (2013). El nuevo PHP. Conceptos avanzados. Espa a: Bubok Publishing.

NTE INEN 1489, I. E. (25 de Enero de 2012). lawresource.org. Recuperado el 16 de Febrero de 2016, de lawresource.org: https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjh592r_v_KAhUEHx4KHcG1CnYQFggaMAA&url=http%3A%2F%2Fwww.arch.gob.ec%2Findex.php%2Fnormas%2Fdoc_download%2F291-norma-inen-1489-2012.html&usg=AFQjCNFjD_tGWRVML1ro2gs

NTE INEN 2251, I. E. (20 de marzo de 2003). law.resource.org. Recuperado el 10 de Febrero de 2016, de law.resource.org: <https://law.resource.org/pub/ec/ibr/ec.nte.2251.2003.pdf>

Ollero, C. (2015). Programaci n con lenguajes de guion en p ginas web (UF1305). Madrid: CEP, S.L.

OLLIVIER, S., & GURY, P.-A. (2016). AngularJS: Desarrolle hoy las aplicaciones web de mañana. Barcelona: ENI.

Peñafiel, M. (2014). Aplicaciones web. Obtenido de <https://es.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>

Perkins, B., Vibe, J., & Reid, J. (2016). Beginning C# 6 Programming with Visual Studio 2015. Canadá: John Wiley & Sons, Inc.

Ramírez, A. (2015). Logística comercial internacional. Barranquilla: Universidad del Norte.

Ramos, A., & María, R. (2014). Aplicaciones Web. Madrid: Parainfo.

Rankins, R., Bertucci, P., Gallell, i. C., & Silverstein, A. (2015). Microsoft SQL Server 2014 Unleashed. USA: SAMS.

Rodríguez, J. M. (2013). Sistemas de información. Obtenido de <http://www.ual.es/~jmrodri/sistemasdeinformacion.pdf>

Sánchez, M. (06 de Noviembre de 2017). Implementación de un sistema de mantenimiento total productivo en el GAD municipal de San Gabriel. (M. A. Sánchez Delgado, Ed.) Recuperado el 2017, de <http://repositorio.utn.edu.ec>: <http://repositorio.utn.edu.ec/handle/123456789/7280>

ScrumInc, Scrum.Org. (Julio de 2014). La Guía Definitiva de Scrum: Las Reglas del Juego. Obtenido de <https://creativecommons.org>: <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish-European.pdf>

SENPLADES. (2013). Plan Nacional del Buen Vivir (). Obtenido de <http://www.buenvivir.gob.ec/web/guest>

Sommerville, I. (2005). Ingeniería del software. Madrid: Pearson Educación.

Sommerville. (2005). Modelo cliente servidor. México: Trillas.

Sprintometer Team, A. (31 de 07 de 2011). sprintometer.com. Obtenido de <http://sprintometer.com/>

TRASOBARES, A. H. (2014). LOS SISTEMAS DE INFORMACIÓN: EVOLUCIÓN Y DESARROLLO. Obtenido de <http://files.grnadasistemasdeinformaion->

cur.webnode.es/200000024-5b31e5c2ae/Dialnet-LosSistemasDeInformacion-793097%20(1).pdf

Trigas, M. (18 de Junio de 2012). UOC. Obtenido de Universidad Oberta de Catalunya:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Upton, D. (2007). Codeigniter for Rapid Php Application Development. Olton: Packt Publishing Ltd.

Urbina, L. (01 de Abril de 2014). Luis Urbina Instalaciones petrolíferas. Recuperado el 21 de Abril de 2016, de Luis Urbina Instalaciones petrolíferas: http://www.luisurbina.com/instalaciones_estaciones_surtidores.php

Urdaneta, M. (2014). La Evaluación de desempeño y la gestión de RRHH.

Valderrey, P. (2014). Administración de sistemas gestores de bases de datos. Madrid: RA-MA.

Valvuela, Z. J., Cardona, S. A., & Villa, D. A. (2008). Programación avanzada en java. Armenia: Elizcom.

Velasco, A. (2013). Aplicaciones web. México: PARANINFO.

Vignaga, A. (2014). Arquitecturas y tecnologías de desarrollo web. Obtenido de http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/LP/AM/01/Arquitecturas_y_tecnologias_para_el_desarrollo_de_aplicaciones_web.pdf

Villada, J. (2014). Instalación y configuración del software de servidor Web. IFCT0509. Málaga: IC Editorial.

ANEXOS

ANEXO A: Historias de Usuario

TABLA 2.4 PLATAFORMA TECNOLÓGICA	
H1	
El sistema web debe poseer el diseño de una interfaz amigable para el usuario final, que guarde la información necesaria en una base de datos, que publique servicios web para ser consumidos por otras tecnologías, para que así pueda ser gestionada y escalable fácilmente.	
Estimación:	62 H
Prioridad: 1	Dependencias:
<ul style="list-style-type: none">• Almacenar información en el esquema creado.• Ejecutar todos los procedimientos almacenados creados en la Base de Datos.• Probar la conexión entre la base de datos y la plataforma de desarrollo.• Levantar el servidor y verificar el funcionamiento correcto de toda la infraestructura	

Tabla 18. Historia de Usuario: Plataforma Tecnológica

Fuente. Propia

TABLA 2.5 SEGURIDAD: USUARIOS Y ROLES	
H2	
El acceso al sistema se realizara con la autenticación de un nombre de usuario y contraseña (Login), el cual dependiendo del rol que posea accederá a los diferentes módulos de acuerdo a las configuraciones de los permisos asignados a esa sesión. Cada usuario almacena sus datos necesarios y una foto que lo identifique, también podrá gestionar sus datos y contraseña. El administrador del sistema puede crear usuarios, crear roles y administrarlos así como bloquear usuarios y cambiar métodos de autenticación y acceso.	
Estimación:	40 H
Prioridad: 1	Dependencias: H1
<ul style="list-style-type: none">• Acceder al sistema con varios usuarios correctos e incorrectos para comprobar su seguridad• Acceder con diferentes roles a la plataforma para validar los permisos de acceso.• Ingresar los datos y la foto de varios usuarios y editarla.• Cambiar la contraseña• Crear varios usuarios y roles• Vincular y editar los roles asignados a los usuarios• Bloquear diferentes usuarios• Cambiar los correos electrónicos de los usuarios• Configurar diferentes métodos de autenticación.	

Tabla 19. Historia de Usuario: Seguridad Usuarios y Roles

Fuente. Propia

TABLA 2.6 ADMINISTRACIÓN GENERAL DEL SISTEMA	
H3	<p>El sistema debe poseer varios parámetros configurables que mejoran el mantenimiento posterior del mismo como: el título, la URL, la institución a la que pertenece, pagina offline, página de bloqueo, carpetas virtuales y físicas del sistema, prefijos, etc.</p> <p>Debe poseer un servicio de E-mail y el servicio de mensajes cortos así como las diferentes configuraciones para su administración.</p> <p>Se realizaran respaldo de la base de datos y también la posibilidad de gestionar los respaldos de la BDD.</p>
Estimación:	52 H
Prioridad: 1	Dependencias: H1, H2

- Cambiar las diferentes configuraciones generales.
- Cambiar la dirección de las carpetas físicas y virtuales del sistema.
- Enviar e-mail y SMS a diferentes usuarios.
- Cambiar la configuración de los e-mails SMS.
- Realizar respaldos de la BDD
- Cambiar la ruta del respaldo de seguridad.

Tabla 20. Historia de Usuario: Administración General del Sistema

Fuente. Propia

TABLA 2.7 ADMINISTRACIÓN DE PARÁMETROS	
H4	<p>Para una correcta administración se hace necesarios crear listas administrables de algunos datos generales que se utilizan a lo largo del proceso funcional de todo el sistema parámetros como: Tipos de vehículo, tipos de mantenimiento, clases de actividades de mantenimiento, partes del vehículo, talleres de reparación, compañías de seguros, etc.</p> <p>La información básica que se necesita almacenar solo consta del nombre del parámetro, se podrá agregar más información de ser necesario.</p>
Estimación:	30 H
Prioridad: 1	Dependencias: H1, H2

- Agregar nuevos parámetros a todas las listas.
- Modificar el nombre de los datos.
- Eliminar elementos de las listas.
- Visualizar el detalle de cada ítem.

Tabla 21 . Historia de Usuario: Administración de Parámetros

Fuente. Propia

TABLA 2.8
GESTIÓN DE EQUIPOS

H5

Se debe crear un módulo de Administración de equipos (vehículos):

- Se podrá agregar nuevos vehículos al inventario con sus respectivas imágenes.
- Se podrá buscar un vehículo por todos sus atributos, así como mostrar una lista de todos los que han sido almacenados.
- Se podrá editar y eliminar los datos de los equipos.
- Se creará una administración de la galería de vehículos.

Todos los datos generales de los equipo, como el tipo de vehículo se debe vincular a la lista de parámetros creados anteriormente para un óptimo manejo de información.

Estimación: 40 H

Prioridad: 2 **Dependencias:** H1, H2,H4

- Agregar varios vehículos con sus respectivas imágenes.
- Editar la información de los equipos.
- Eliminar vehículos.
- Realizar búsquedas de los vehículos por diferentes filtros necesarios.
- Cambiar las fotos de los vehículos.
- Mostrar todos los equipos almacenados en la BDD.

Tabla 22. Historia de Usuario: Gestión de Equipos

Fuente. Propia

TABLA 2.9
GESTIÓN DE PLANES DE MANTENIMIENTO

H6

Se debe crear un módulo de Administración de Plantillas de mantenimientos, Las plantillas son planes de mantenimiento generales que se crean y se asignan de acuerdo a la marca y modelo del vehículo.

- Se podrá agregar nuevas plantillas seleccionando la marca principalmente.
- Se podrá buscar platillas según el tipo de vehículo.
- Se podrá editar y eliminar los datos de cada actividad vinculada a cada plantilla.

Se debe crear un módulo de Asignación de Planes de mantenimiento donde se vinculan los equipos con su respectivo plan.

- Se podrá agregar a una lista los vehículos a los cuales se les va a asignar las plantillas.
- En la lista a la cual se va a vincular se podrá ajustar configuraciones de la actividad si es necesario ya que algunos atributos pueden cambiar debido a cualidades únicas de cada equipo como el tiempo de uso que tiene.
- Solo si se realiza la vinculación los ajustes a cada actividad se harán efectivos, las plantillas no se verán afectadas en la vinculación.
- Se podrá realizar la búsqueda de los planes de mantenimiento asignados a cada vehículo, así como todos los planes asignados a todos los vehículos.

Todos los datos generales de las plantillas y planes de mantenimiento, como las marcas y tipos de actividades se deben vincular a la lista de parámetros creados anteriormente para un óptimo manejo de información.

Estimación: 120 H

Prioridad: 2 **Dependencias:** H1, H2, H4, H5

- Crear varias plantillas de mantenimiento para la mayoría de marcas de equipos que se encuentran en el inventario.
- Editar la información de las plantillas.
- Eliminar actividades de las plantillas.
- Realizar búsquedas de plantillas según un tipo de vehículo.
- Vincular un vehículo con su plantilla.
- Vincular varios vehículos con una plantilla.
- Cambiar los datos de una plantilla al vincular y comprobar que se genere el plan de mantenimiento y que no afecte el cambio a la plantilla.
- Buscar todos los planes creados y buscar por cada vehículo.

Tabla 23. Historia de Usuario: Gestión de Planes de Mantenimiento

Fuente. Propia

H7	TABLA 2.10 GESTIÓN DE MANTENIMIENTOS NO PLANIFICADOS
<p>Se debe crear un módulo de Administración de Mantenimientos no Planificados (Correctivos o Accidentes) :</p> <ul style="list-style-type: none"> - Se despliega una lista de todos los equipos en inventario. - Se podrá agregar uno o más vehículos a una lista para posteriormente asignarles un mantenimiento correctivo. - Se agrega información relevante del accidente, como los costos de mantenimiento, talleres, compañías de seguros etc. - Se genera el mantenimiento correctivo. - Se debe agregar evidencias a los mantenimientos correctivos, así que se escoge el mantenimiento y se vinculas fotos y archivos necesarios. - Se podrá buscar un mantenimiento correctivo por todos sus atributos, así como mostrar una lista de todos los que han sido almacenados. <p>Todos los datos generales de accidentes, como talleres y compañías de seguros se deben vincular a la lista de parámetros creados anteriormente para un óptimo manejo de información.</p>	
Estimación:	88 H
Prioridad: 3	Dependencias: H1, H2, H4, H5

- Agregar un vehículo a la lista.
- Agregar varios vehículos a la lista.
- Agregar información del accidente.
- Generar varios mantenimientos no planificados.
- Agregar varias evidencias a los diferentes mantenimientos y editar algunos.
- Realizar búsquedas por sus diferentes filtros.
- Mostrar todos los mantenimientos almacenados en la BDD.

Tabla 24. Historia de Usuario: Gestión de Mantenimientos no Planificados

Fuente. Propia

TABLA 2.11

GESTIÓN DE REVISIONES VEHICULARES**H8**

Se debe crear un módulo de Administración de Revisiones Vehiculares al cual solo accederá el rol de Secretaria del sistema :

- Se despliega una lista de todos los equipos en inventario.
- Se podrá agregar uno o más vehículos a una lista para posteriormente enviarlos a revisión.
- Se agrega la fecha prevista y el encargado.
- Se genera la revisión.
- Se podrá buscar una revisión por todos sus atributos, así como mostrar una lista de todos los que han sido almacenados.

Estimación: 88 H

Prioridad: 3 **Dependencias:** H1, H2, H4, H5

- Agregar un vehículo a la lista.
- Agregar varios vehículos a la lista.
- Agregar información de la revisión.
- Generar varias revisiones.
- Realizar búsquedas por sus diferentes filtros.
- Mostrar todos los mantenimientos almacenados en la BDD.

Tabla 25. Historia de Usuario: Gestión de Revisiones Vehiculares

Fuente. Propia

TABLA 2.12

GESTIÓN DE RUTAS Y KILOMETRAJES**H9**

Se debe crear un módulo de Administración de Rutas:

- Se podrá agregar uno o más vehículos a una lista para posteriormente enviarlos a revisión.
- Se agrega el destino las fechas previstas, el encargado y las horas de utilización del equipo.
- Se crea la asignación de ruta al vehículo.
- Se podrá buscar una ruta por todos sus atributos, así como mostrar una lista de todos los que han sido almacenados.

Se debe crear un módulo de Registro de Kilometrajes:

- A un vehículo se registra el kilometraje recorrido ya sea por ruta o por otra movilización y se actualiza el kilometraje actual.

Estimación: 124 H

Prioridad: 3 **Dependencias:** H1, H2, H4, H5

- Agregar un vehículo a la lista.
- Agregar varios vehículos a la lista.
- Agregar información de la ruta.
- Generar varias rutas.
- Realizar búsquedas por sus diferentes filtros.
- Mostrar todos los registros almacenados en la BDD.
- Registrar varios kilometrajes.

Tabla 26. Historia de Usuario: Rutas y Kilometrajes

Fuente. Propia

TABLA 2.13
APROBACIÓN DE PETICIONES

H10

Para que los mantenimientos y revisiones empiecen a ser calendarizados es necesario que un supervisor apruebe dichas solicitudes ya que la generación de estas le compete al técnico, por lo tanto se crea un módulo al cual solo los supervisores tendrán acceso donde se encargará de aprobar o rechazar Mantenimientos y Revisiones.

Estimación: 70 H

Prioridad: 3 **Dependencias:** H1, H2, H4, H5, H6, H7, H8,

- Aprobar y Rechazar Planes de Mantenimiento
- Aprobar y Rechazar Mantenimientos Correctivos
- Aprobar y Rechazar Revisiones Vehiculares

Tabla 27. Historia de Usuario: Aprobación de Peticiones

Fuente. Propia

TABLA 2.14
INVENTARIO DE HERRAMIENTAS

H11

Se crea un módulo de herramientas donde se agregan administraciones de diferentes materiales, es necesario crear un inventario de los repuestos que se adquieren en los mantenimientos, se deben agregar nuevos repuestos, editar y eliminar registros y realizar búsquedas por cualquier filtro y mostrar todos los repuestos, también es necesario agregar imágenes de los repuestos por lo tanto se realiza una administración de la galería donde se podrán buscar repuestos y cambiar las imágenes.

Estimación: 60 H

Prioridad: 4 **Dependencias:** H1, H2

- Crear Repuestos
- Agregar imágenes a los repuestos
- Realizar búsquedas con todos los filtros necesarios
- Mostrar todos los registros almacenados.
- Editar información.
- Eliminar repuestos.
- Acceder a la Galería y editar algunas imágenes.

Tabla 28. Historia de Usuario: Inventario de Herramientas

Fuente. Propia

TABLA 2.15
NOTIFICACIONES Y REPORTE

H12

Desde que las solicitudes son aprobadas, y con los registros de kilometrajes, fechas y horas trabajadas según las rutas o recorridos, si el mantenimiento planificado, mantenimiento correctivo o revisión vehicular se acercan, el sistema genera notificaciones que alertan sobre la aproximación de estos procesos y que deben realizarse, para realizar estos procesos es necesario generar una orden de trabajo impresa. También es necesario generar reportes de los diferentes datos.

Estimación: 82 H

Prioridad: 4 **Dependencias:** H1, H2, H5, H6, H7, H8, H9, H10, H11

- Generar notificaciones con diferentes estados
- Generar notificaciones de Planes de Mantenimiento, mantenimientos correctivos y revisiones vehiculares.
- Generar ordenes de Trabajo
- Generar Reportes de la información primordial del sistema

Tabla 29. Historia de Usuario: Notificaciones y Reportes

Fuente. Propia

ANEXO B: Interfaces del Sistema

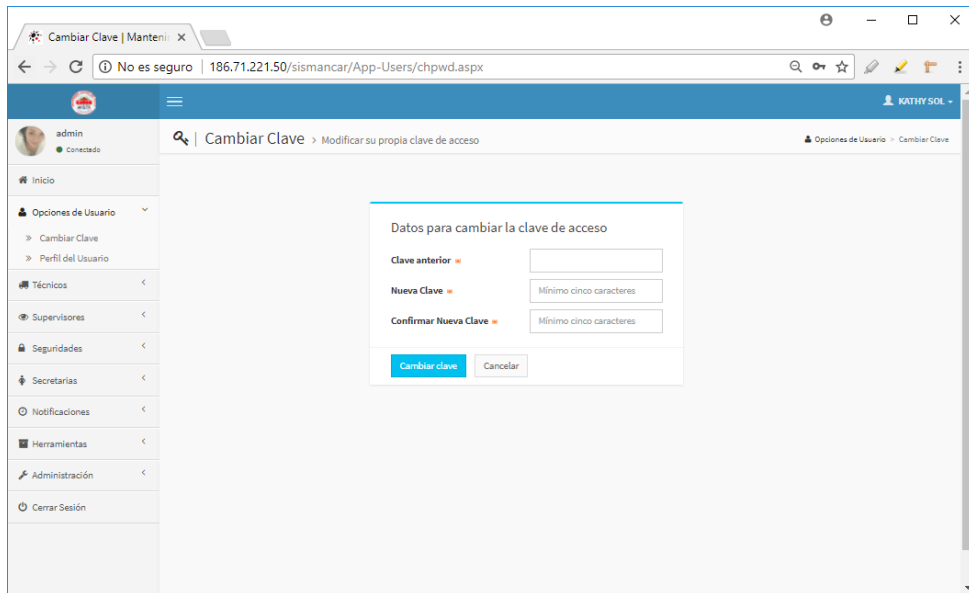


Figura 35. Actualizacion Clave

Fuente. Propia

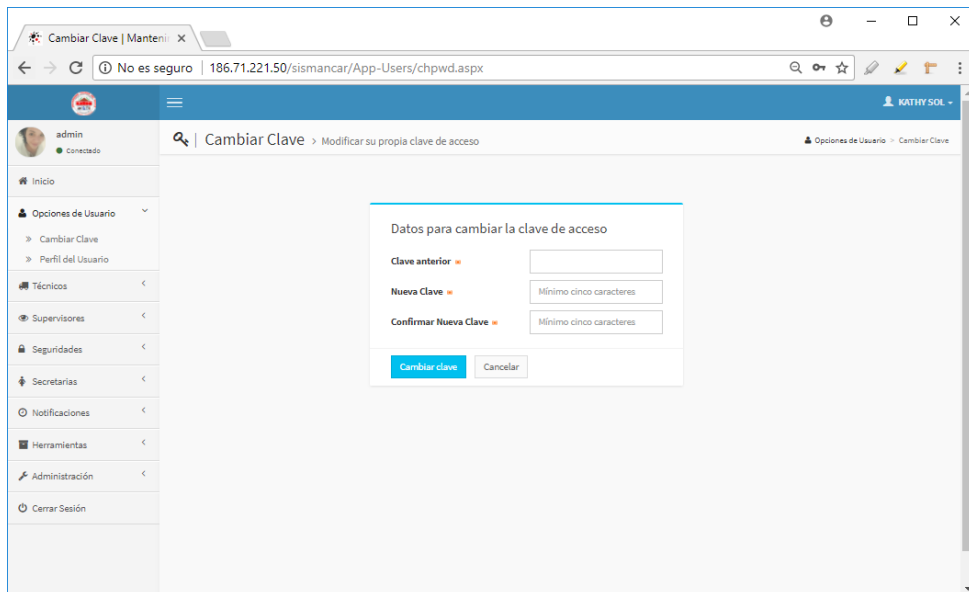


Figura 36. Actualizacion Clave

Fuente. Propia

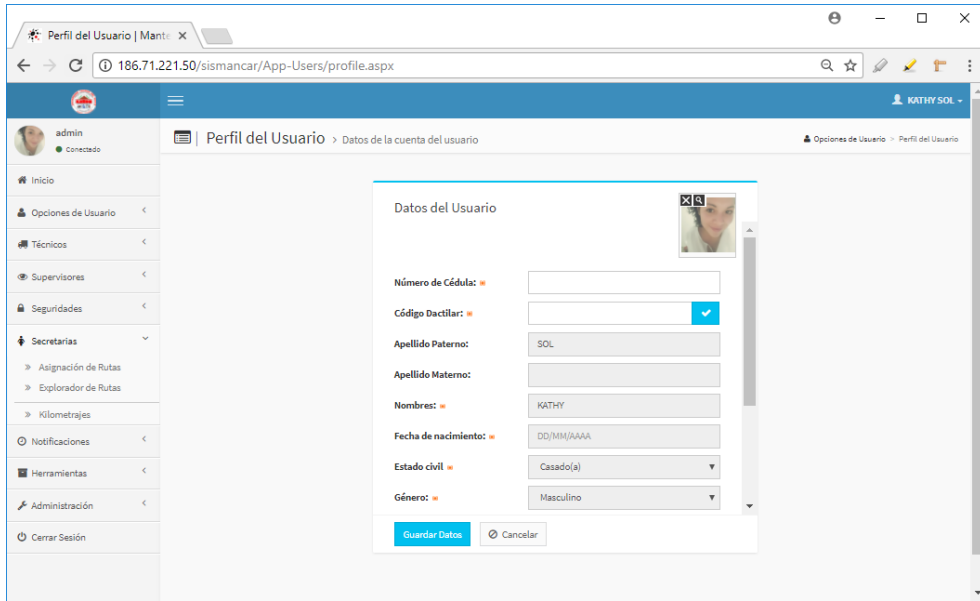


Figura 37. Perfil de Usuario

Fuente. Propia

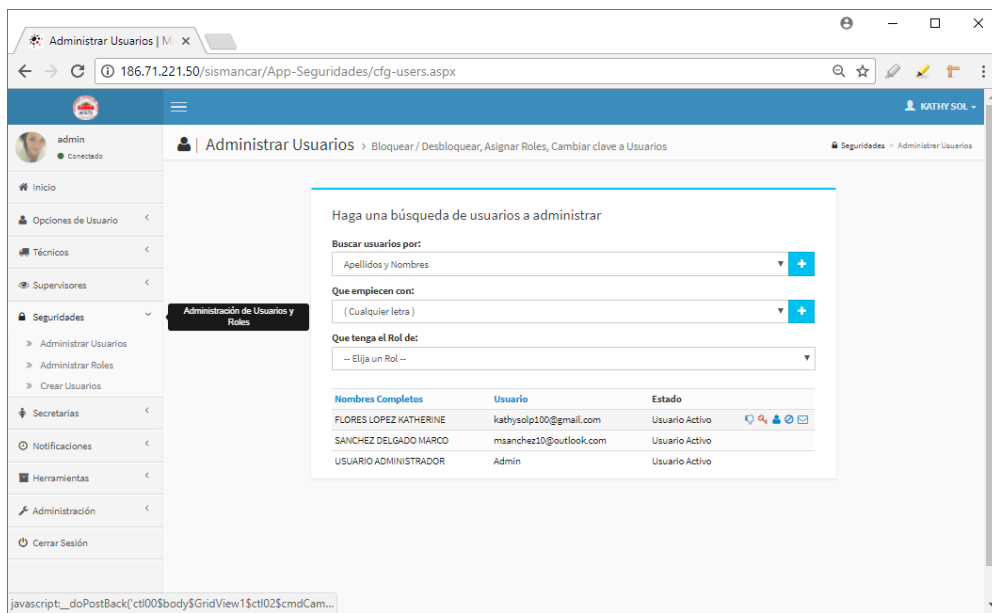


Figura 38. Administración de Usuarios

Fuente. Propia

ANEXO C: CERTIFICADO



GOBIERNO AUTÓNOMO DESCENTRALIZADO
MUNICIPAL DE MONTÚFAR



ALCALDÍA

San Gabriel, 13 de Julio del 2017

CERTIFICADO

Por medio del presente nos permitimos certificar que el Sr. Marco Aníbal Sánchez Delgado portador de la cedula 040165274-8, ha culminado de manera satisfactoria con el proceso de estudio e implementación del Sistema de Mantenimiento Automotriz SISMANCAR, cumpliendo con el respectivo proceso para análisis de mantenimiento preventivo, correctivo y predictivo de los vehículos del GAD Municipal de Montufar en el periodo de 6 meses desde el 2 de enero del 2017 hasta la presente fecha, de igual manera menciono las ventajas obtenidas, en el cual nos ayudó a documentar y mantener organizada la información que requiere la Jefatura de Transportes para su respectivo mantenimiento vehicular en base a la programación de los mismos, calculando de forma automática los insumos a utilizar y mano de obra; así como la reducción de costos programados de la Municipalidad en un porcentaje del 13.1%. Es necesario mencionar que el SISTEMA SISMANCAR se encuentra en normal funcionamiento dentro de la Dirección de Obras Públicas en la Unidad de Transportes.

Atentamente,



Dña. Juan Acosta
ALCALDE



Tpo. Tomás Espinosa
JEFE DE TRANSPORTE



San Gabriel, Calle Sucre 23-81 y Bolívar Teléfono: 2290 123 Ext 209