



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

DESARROLLO DE UNA SOLUCIÓN CON TECNOLOGÍAS WEB PARA EL PROCESO DE INSPECCIÓN DE VEHÍCULOS DE COMPAÑÍAS DE SEGUROS.

Trabajo Especial de Grado Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Bachiller
Ricardo Antonio Cerqueira Salazar
C.I.: V-19940594
Para optar al título de
Licenciado en Computación
Tutor:
Prof. Franky Uzcategui

Caracas, octubre de 2017

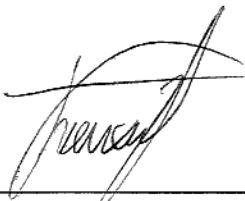
UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "Desarrollo de una Solución con Tecnologías Web para el proceso de Inspección de Vehículos de Compañías de Seguros", y presentado por el Br. Ricardo Antonio Cerqueira Salazar C.I. 19.940.594, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día 18 de 07 de 2017, a las 10 a.m., para que su autor lo defendiera en forma pública en la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los 18 días del mes de 07 del año 2017.



Prof. Franky Uzcátegui
Tutor

Profa. Mercy Ospina
Jurado Principal



Prof. Franklin Sandoval
Jurado Principal

DEDICATORIA

A mis familiares y por sobre todo a mi madre
por su apoyo incondicional y siempre creer en uno.

A mis amigos, compañeros de trabajo
y todos aquellas personas que hicieron esto posible.

Ricardo Cerqueira

AGRADECIMIENTOS

Principalmente a mi madre Rosa Cerqueira, quien es mi modelo a seguir por su gran apoyo incondicional siempre con un "si se puede", lo cual me motivo mucho a avanzar en contra de las adversidades encontradas en el camino, aquí en la universidad, una mujer luchadora la cual espero este orgullosa de mi por este logro que hemos alcanzado juntos.

A mis familiares, mis abuelos, mis tíos los cuales siempre me brindaron buenos consejos, palabras de apoyo y una comida caliente. Personas que estaban siempre pendientes de mi y que me recordaban que siempre que necesitara ayuda la familia estaba para ello.

A mis amigos, los que están lejos y los que están cerca aquellos con los que siempre pude contar en momentos de incertidumbre, ya sea buscando consejos o ayudando a desconectarme un poco, liberar la mente y agarrar impulso para los siguientes retos.

A mis profesores, aquellos que cambiaron mi modo de pensar, creando una mente crítica y sembraron conocimiento con el cual puedo destacar en la materia y lograr un óptimo rendimiento en los retos, tanto del ámbito laboral como cotidiano.

A mi tutor Franky Uzcátegui por su paciencia y por su orientación en este trabajo de investigación. Gracias por su guía para poder cumplir la meta de graduarme.

A la Universidad Central de Venezuela la cual que vence la sombra por formar al profesional que soy hoy en día y aunque actualmente se encuentre en momentos difíciles, siempre se lucha por su autonomía y que el conocimiento sea enseñado de manera transparente y objetivo.

A todos, gracias totales.

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

DESARROLLO DE UNA SOLUCIÓN PARA EL PROCESO DE INSPECCIÓN DE VEHÍCULOS EN UNA COMPAÑÍA DE SEGUROS.

Autor: Ricardo Antonio Cerqueira Salazar, C.I V-19940594

Tutor: Prof. Franky Uzcátegui Polo

RESUMEN

El presente trabajo especial de grado expone el desarrollo de un sistema de información con tecnologías web, el cual fue elaborado como el módulo web, del proyecto compuesto por una aplicación móvil y una aplicación web, cuya finalidad es de automatizar el proceso de inspección de vehículos en una compañía de seguros. Para el desarrollo de este proceso es indispensable que el vehículo sea examinado por un perito el cual determinará las condiciones con las cuales fue entregado el vehículo y si el cliente puede o no optar por el servicio. Este proceso se conoce como peritaje es imperativo que el cliente se presente con el vehículo, solicite un ticket, espere a ser atendido por el perito que le fue asignado, el perito realice la inspección. Sin embargo, este proceso presenta una serie de inconveniente en cuanto a la persistencia de los datos y la administración se refiere, almacenamiento de la planilla en físico, gestión del proceso de aprobación no está automatizado, por lo que mediante este estudio se creó un sistema que permite una mejor gestión de las inspecciones realizadas.

En el desarrollo de este sistema de información, se aplicó las metodologías de Agilus y Scrum en conjunto. Tomando como base el desarrollo por interfaces de Agilus y como gestión de tiempo los sprints de Scrum.

Se utilizaron diversas herramientas para la creación de este sistema, entre ellas el framework Django para la creación de la aplicación web ya que permite el acceso a la información referente al proceso de inspección de vehículos, desde cualquier lugar donde se tenga acceso a la web, información la cual se encuentra almacenada en una base de datos PostgreSQL la cual se comunica la base de datos central de la compañía de seguros.

PALABRAS CLAVE: Inspección de vehículos, Compañía de seguros, Centro de inspección, Gestión solicitudes inspección.

Índice general

Índice de figuras	IX
Índice de tablas	XI
INTRODUCCIÓN	1
1. Problema de Investigación	2
1.1. Situación actual	2
1.2. Planteamiento del problema	3
1.3. Justificación	4
1.4. Objetivo general	4
1.5. Objetivos específicos	4
1.6. Alcance	5
2. Marco Conceptual	6
2.1. Sistema de información	6
2.2. Tipos de sistemas de información	6
2.2.1. Sistema de procesamiento de transacciones (TPS)	6
2.2.2. Sistema de información gerencial (MIS)	7
2.2.3. Sistema de apoyo a la toma de decisiones (DSS)	7
2.2.4. Sistema de información ejecutiva (EIS)	7
2.3. Base de Datos	7
2.3.1. Sistema Manejador de Base de Datos	8
2.3.2. PostgreSQL	8
2.3.3. MySQL	9
2.3.4. Oracle	9
2.3.5. Cuadro Comparativo SMBD	10
2.4. Tecnologías de desarrollo web	11
2.4.1. HTML	11
2.4.2. CSS	11
2.4.3. JavaScript	11
2.4.4. JQuery	12
2.4.5. JQueryUI	12

2.4.6.	Boopstrap	12
2.4.7.	Django	13
2.4.8.	CakePHP	14
2.4.9.	Ruby on Rails	16
2.4.10.	JSON	16
2.5.	El Seguro	17
2.5.1.	Contrato del seguro	17
2.5.2.	Funciones del Seguro	17
2.5.2.1.	Funciones sociológicas del seguro	17
2.5.2.2.	Funciones económicas del seguro	17
2.5.2.3.	Funciones laborales del seguro	18
2.5.3.	Póliza de seguro	18
2.5.4.	Seguro de Vehículo	18
2.5.5.	El Perito	19
2.5.5.1.	Aspectos técnicos	19
2.5.5.2.	Aspectos administrativo-legales	20
2.5.5.3.	Aspecto negociador	20
2.5.5.4.	Inspección para asegurar un Vehículo	21
2.5.5.5.	Caso de Estudio	22
2.5.5.6.	Planilla de Suscripción	23
3.	Marco Metodológico	24
3.1.	Metodología Tradicional	24
3.1.1.	RUP	24
3.1.1.1.	Ciclo de vida	24
3.1.2.	Cascada	26
3.1.2.1.	Fases del Modelo	26
3.1.3.	Cuadro Comparativo	27
3.2.	Metodología Ágil	27
3.2.1.	SCRUM	28
3.2.1.1.	Ciclo de vida	28
3.2.1.2.	Spring	28
3.2.1.3.	Planificación del sprint	29
3.2.1.4.	Scrum diario	29
3.2.1.5.	Revisión de las Iteraciones	29
3.2.1.6.	Retrospectiva	29
3.2.2.	XP	30
3.2.2.1.	Ciclo de vida	30
3.2.2.2.	Fase I: Exploración	31
3.2.2.3.	Fase II: Planificación de la Entrega	31
3.2.2.4.	Fase III: Iteraciones	32
3.2.2.5.	Fase IV: Producción	32
3.2.2.6.	Fase V: Mantenimiento	33

3.2.2.7.	Fase VI: Muerte del Proyecto	33
3.2.3.	Agilus	33
3.2.3.1.	Ciclo de vida	34
3.2.4.	Cuadro Comparativo	35
3.3.	Cuadro comparativo [Metodologías Tradicionales vs Ágiles]	36
4.	Marco Aplicativo	37
4.1.	Descripción general de la solución	37
4.2.	Aplicación de la metodología Scrum	39
4.2.1.	Lista de Objetivos (Pila del Producto)	39
4.2.2.	Lista de tareas de la iteración (Pila del Sprint)	41
4.3.	Requerimientos del Sistema	42
4.4.	Perfiles de Usuario	42
4.5.	Descripción del flujo asociado a la solución	44
4.5.1.	Proceso de creación de solicitud	44
4.5.2.	Proceso de gestión de solicitud de inspección	46
4.6.	Análisis del modelo de datos y definición	47
4.6.1.	Listado de tablas de la aplicación	47
4.6.2.	Modelo de datos	50
4.7.	Descripción de los módulos del sistema y sus interfaces	51
4.7.1.	Interfaces de la aplicación	52
4.8.	Fase de pruebas	69
4.8.1.	Pruebas funcionales	69
4.8.2.	Pruebas de aceptación	73
	CONCLUSIÓN Y RECOMENDACIONES	79
	BIBLIOGRAFÍA	81

Índice de figuras

1.1. Arquitectura Propuesta. Fuente: El Autor.	3
2.1. Proceso Actual. (2016)	22
2.2. Proceso de Suscripción. (2016)	23
3.1. Proceso RUP. (Tomada de DTyOC (2016))	25
3.2. Proceso Cascada. (Tomada de Wiki Uploads (2013))	26
3.3. Proceso Scrum. (Tomada de Methods & Tools (2004))	28
3.4. Proceso Scrum. (Tomada de Javier Cifuentes (2004))	31
3.5. El método AgilUs: etapas, actividades y artefactos.	34
4.1. Arquitectura Propuesta. Fuente: El Autor.	38
4.2. Proceso de creación de solicitud de inspección. Fuente:El Autor.	44
4.3. Proceso de gestión de solicitud de inspección. Fuente:El Autor.	46
4.4. Modelo de datos de la aplicación. Fuente:El Autor.	50
4.5. Diagrama de módulos del sistema. Fuente:El Autor.	51
4.6. Interfaz de Inicio de Sesión. Fuente:Print Screen.	52
4.7. Interfaz de Recuperar Contraseña. Fuente: Print Screen.	53
4.8. Interfaz de Registro de Usuarios. Fuente: Print Screen.	53
4.9. Interfaz de Configuración del Perfil de Usuario. Fuente: Print Screen.	54
4.10. Interfaz de bandeja edición usuario por Administrador. Fuente: Print Screen.	55
4.11. Modal Inactivar Usuario. Fuente: Print Screen.	55
4.12. Modal Eliminar Usuario. Fuente: Print Screen.	56
4.13. Interfaz de edición de usuario por Administrador. Fuente: Print Screen.	56
4.14. Interfaz de creación de solicitud de inspección. Fuente: Print Screen.	57
4.15. Modal crear solicitud de inspección. Fuente: Print Screen.	58
4.16. Interfaz de bandeja solicitudes de inspección por Taquilla. Fuente: Print Screen.	58
4.17. Modal eliminar solicitud de inspección. Fuente: Print Screen.	59
4.18. Interfaz de bandeja solicitudes de inspección usuario Inspector. Fuente: Print Screen.	59
4.19. Interfaz del formulario de registro de los datos del vehículo. Fuente: Print Screen.	60
4.20. Interfaz del formulario de registro de condiciones generales del vehículo. Fuente: Print Screen.	61

4.21. Interfaz del formulario de registro de la mecánica del vehículo. Fuente: Print Screen.	62
4.22. Interfaz del formulario de registro de los accesorios del vehículo. Fuente: Print Screen.	63
4.23. Interfaz del formulario de registro de los detalles y datos del vehículo. Fuente: Print Screen.	64
4.24. Interfaz del formulario de registro de los documentos del vehículo. Fuente: Print Screen.	64
4.25. Interfaz de la vista previa de los datos del vehículo (Parte 1). Fuente: Print Screen.	66
4.26. Interfaz de la vista previa de los datos del vehículo (Parte 2). Fuente: Print Screen.	67
4.27. Modal finalizar la solicitud de inspección. Fuente: Print Screen.	68
4.28. Modal finalizar la solicitud de inspección. Fuente: Print Screen.	68
4.29. Resultados de la pregunta 1. Fuente: El Autor.	74
4.30. Resultados de la pregunta 2. Fuente: El Autor.	74
4.31. Resultados de la pregunta 3. Fuente: El Autor.	75
4.32. Resultados de la pregunta 4. Fuente: El Autor.	75
4.33. Resultados de la pregunta 5. Fuente: El Autor.	76
4.34. Resultados de la pregunta 6. Fuente: El Autor.	76
4.35. Resultados de la pregunta 7. Fuente: El Autor.	77
4.36. Resultados de la pregunta 8. Fuente: El Autor.	77
4.37. Resultados de la pregunta 9. Fuente: El Autor.	78

Índice de tablas

2.1. Tabla comparativa de los Sistemas Manejadores de Base de Datos	10
3.1. Tabla comparativa de metodologías tradicionales	27
3.2. Tabla comparativa de metodologías ágiles	35
3.3. Diferencias entre metodologías ágiles y tradicionales	36
4.1. Pila del Producto. Fuente: El Autor.	39
4.2. Pila del Producto. Fuente: El Autor.	40
4.3. Instrumento evaluación módulos del sistema (Parte 1). Fuente: El autor . . .	69
4.4. Instrumento evaluación módulos del sistema (Parte 2). Fuente: El autor . . .	70
4.5. Matriz de ciclos de pruebas. Fuente: El autor	71
4.6. Encuesta de aceptación. Fuente: El Autor	73

INTRODUCCIÓN

En la actualidad el mercado asegurador venezolano se encuentra en una alta demanda. Debido a esto las empresas aseguradoras poseen un alto índice de clientes cuyo manejo de información es indispensable para dar un buen servicio.

El alto volumen de información manejada en la suscripción de vehículos de cada uno de los clientes genera un fuerte impacto administrativo, tanto de personal como de tiempo. Los peritos encargados de la inspección de los vehículos, diariamente manejan un gran volumen de planillas las cuales pueden llegar a traspapelarse.

Por lo cual se propone una solución computacional mediante la cual se plantea desarrollar una aplicación web que permitirá el almacenamiento de dichas planillas de suscripción, para llevar un mejor manejo de las mismas. Administrarlas en caso de una re-inspección y poder gestionar mejor tramites como: atención del cliente, suscripción al seguro e inspección del vehículo. Garantizando así un mejor desempeño y velocidad en la atención de los clientes generando así una mayor satisfacción para los mismos.

La estructura de este documento consiste de las siguientes secciones: El primer capítulo Marco Teórico consiste de los conceptos y definiciones sobre los temas y tecnologías utilizadas para el desarrollo de este estudio.

El segundo capítulo Marco Metodológico consta de las diferentes metodologías tanto ágiles como tradicionales para el desarrollo de software. En este capítulo se plantearán las diferentes características entre ambas tecnologías mostrando sus ventajas y desventajas. Además se expondrá un cuadro comparativo en el cual se podrá establecer cual metodología es la mas conveniente a utilizar.

El tercer capítulo al ser una solución que abarca el peritaje de vehículos, se incluyen conceptos y términos asociados con el área de seguro, peritaje e inspecciones de vehículos.

Por ultimo en el cuarto capítulo se presenta la propuesta de Trabajo Especial de Grado, mostrando los objetivos que se desean lograr al culminar, en conjunto con la propuesta de solución, su justificación y el alcance de la misma. Además se especificarán las diferentes tecnologías que se utilizarán para el desarrollo de la propuesta.

Capítulo 1

Problema de Investigación

En este capítulo se detallan la problemática y los puntos a tratar en este documento de tesis, además de la justificación por el cual se está realizando el presente trabajo. Se presentarán los objetivos generales y específicos del documento así como el alcance y las limitaciones que tendrá el mismo.

1.1. Situación actual

Hoy en día el mercado asegurador Venezolano, sufre una gran demanda, para poder con el alto flujo de cliente y prestar un servicio eficiente y eficaz, posee una gran necesidad de adaptarse a las nuevas tecnologías y desarrollar un proceso, optimizado de su modo de realizar las inspecciones de vehículos.

Actualmente, el proceso de inspección se encuentra realizándose de manera que los documentos generados son físicos, por lo cual su persistencia e integridad se ve afectada por el ambiente en donde se resguarden luego de la inspección. Es por ello que con frecuencia se buscan soluciones y/o alternativas que puedan cumplir con el objetivo y optimizar este proceso, utilizando las tecnologías actuales que estén al alcance de las personas.

Para el desarrollo de este proyecto es esencial conocer el procedimiento de inspección de vehículos, para dar una solución que cumpla con los requerimientos de dicho proceso. Este proceso de inspección se realiza al momento de que los clientes quieren solicitar una póliza de seguro automovilístico validando la integridad del vehículo, por lo que optimizar este proceso ayudaría a las aseguradoras a ampliar su campo en donde prestan sus servicios.

1.2. Planteamiento del problema

Actualmente al asistir a la cita para la inspección del vehículo, el perito registra las especificaciones del mismo en una hoja para luego ser archivada y guardada, este proceso se realiza manualmente.

El procesos presenta una serie de inconveniente en cuando a la persistencia de los datos y la administración se refiere:

- Las planillas de inspección se guardan solamente en físico.
- El riesgo de perder información que no esta respaldada.
- El acceso a la información, al ser una búsqueda manual, no es eficiente.

Ante la situación descrita, se plantea elaborar una solución con tecnología Web, que permita a los peritos, mantener un registro digital de las planillas de inspección de los vehículos.

Para ello se propone el desarrollo de la siguiente arquitectura con las siguientes tecnologías:

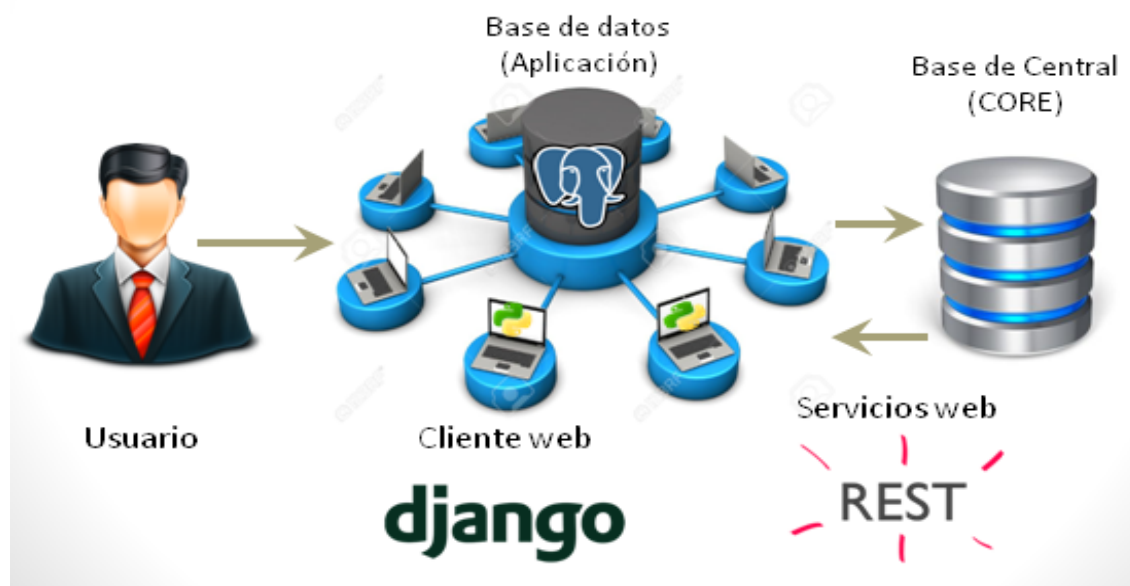


Figura 1.1: Arquitectura Propuesta. Fuente: El Autor.

Para la base de datos utilizaremos PostgreSQL, ya que es una de las bases de datos mas potentes de software libre, con un alto rendimiento, seguridad y a su vez estar disponible prácticamente para todas las versiones de los sistemas operativos Unix y también Windows. Su versatilidad y robustez la hace el candidato perfecto para el sistema que se busca implementar.

Se propone utilizar el framework de Django para el desarrollo del sistema. Ya que siendo uno de los muchos frameworks que están establecidos sobre la base del patrón MVC en sus beneficios se encuentra la separación de responsabilidad y organización del código. Este framework incorpora un patrón llamado MTV (Modelo-Template-Vista), los templates son donde se implementan todas las interfaces de los usuarios, que serán desplegadas por las vistas en el lado del servidor y los múdeles corresponde a la base da datos donde persistirá la información. Además posee un sistema jerárquico de plantilla que proporciona la reutilización de código y la extensibilidad de las aplicaciones. Posee un buen soporte para PostgreSQL, la base de datos que se piensa utilizar para el sistema.

1.3. Justificación

Un sistema con tecnologías web, es la mejor solución para sistematizar la administración de las planillas de inspección y para el almacenamiento de las mismas de forma digital, para que de esta forma la empresa aseguradora pueda llevar un registro cuyo gestión sea mas eficiente y de múltiple acceso.

La implementación de este sistema representa una reducción riesgos y costos a largo plazo para la compañía, permitiendo que la gestión de las planillas de inspección y los procesos vinculados con ellas se lleven acabo de manera óptima.

1.4. Objetivo general

Desarrollar una solución para el proceso de inspección de vehículos en una compañía de seguros.

1.5. Objetivos específicos

- Analizar especificaciones del proceso de inspección de vehículos para suscripción.
- Diseñar modelo de datos, de las interfaces y de los componentes de los módulos del sistema.
- Desarrollar componentes diseñados.

- Realizar pruebas de funcionalidad y de usabilidad.

1.6. Alcance

Como alcance del trabajo especial de grado se tiene previsto lograr el registro de usuarios que accederán al sistema para gestionar las planillas, la captura de los datos de la planilla de inspección, la edición de la planilla en caso de incongruencias antes de ser almacenada y guardar el conjunto de datos en la base de datos para su posterior consulta

Capítulo 2

Marco Conceptual

En este capítulo se expondrán los diferentes concepto asociados a las tecnologías, enfoques y herramientas de desarrollo. Además de los términos orientados al área de Seguro, que es donde esta enfocada la solución de este proyecto.

2.1. Sistema de información

Laudon y Laudon (1996) hace mención que los SI son conjuntos organizados de elementos dirigidos a recoger, procesar, almacenar y distribuir información de manera que pueda ser utilizada por las personas adecuadas dentro de las empresas, de modo que desempeñen sus actividades de manera eficaz y eficiente

Desde el punto de vista empresarial, los sistemas de información tienen como propósito perfeccionar las actividades llevadas a cabo en una organización, y así alcanzar ventajas competitivas.

2.2. Tipos de sistemas de información

2.2.1. Sistema de procesamiento de transacciones (TPS)

Laudon y Laudon (1996) hace referencia a los TPS como aquellos sistemas que recopilan, almacenan y alteran la información creada a partir de transacciones llevadas a cabo dentro de una organización se denomina sistema de procesamiento de transacciones. Tiene como finalidad procesar las transacciones diarias de una empresa, acumulando toda la información recibida en una base de datos para su posterior consulta.

2.2.2. Sistema de información gerencial (MIS)

McLeod (2000) establece que un sistema de información gerencial es aquel utilizado por la empresa para solventar inconvenientes en la misma. Es decir, el objetivo del mismo es la suministración de información para la resolución de problemas a través de la interacción entre tecnologías y personas.

Los datos aportados por el sistema deben disponer de cuatro cualidades elementales: calidad, oportunidad, cantidad y relevancia.

2.2.3. Sistema de apoyo a la toma de decisiones (DSS)

Laudon y Laudon (1996) estipula que este sistema se basa en el estudio y la comparación entre un conjunto de variables con el objeto de contribuir a la toma de decisiones dentro de una empresa. El apoyo dado por el sistema involucra la estimación, valoración y balance entre alternativas. Al igual que el sistema de información gerencial, esta tecnología interacciona con personas en el filtrado de información que permite optar por la decisión mas acertada.

2.2.4. Sistema de información ejecutiva (EIS)

Laudon y Laudon (1996) hace referencia que esta tecnología es utilizada por los gerentes de una empresa, ya que permite acceder a la información interna y externa de la misma, disponiendo de los datos que puedan llegar a afectar su buen rendimiento. De esta manera, el ejecutivo podrá conocer el estado de todos los indicadores, incluso aquellos que no cumplan con las expectativas y a partir de esto, tomar las medidas que considere adecuadas.

2.3. Base de Datos

“Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible.”(CCM (2016))
Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. Sistema de información es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

2.3.1. Sistema Manejador de Base de Datos

“Un sistema manejador de bases de datos (SGBD, por sus siglas en inglés) o DataBase Management System (DBMS) es una colección de software muy específico, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.” (Emilio Vega (2015))

Como su propio nombre indica, el objetivo de los sistemas manejadores de base de datos es precisamente el de manejar un conjunto de datos para convertirlos en información relevante para la organización, ya sea a nivel operativo o estratégico.

Lo hace mediante una serie de rutinas de software para permitir su uso de una manera segura, sencilla y ordenada. Se trata, en suma, de un conjunto de programas que realizan tareas de forma interrelacionada para facilitar la construcción y manipulación de bases de datos, adoptando la forma de interfaz entre éstas, las aplicaciones y los mismos usuarios.

El SMDB puede dividirse en tres subsistemas:

- El sistema de administración de archivos: para almacenar información en un medio físico
- El DBMS interno: para ubicar la información en orden
- El DBMS externo: representa la interfaz del usuario

A continuación se presentarán algunos ejemplos de SMDB.

2.3.2. PostgreSQL

Rafael (2016) afirma: “PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.”

PostgreSQL utiliza un modelo cliente/servidor y usa multi-procesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo.

PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a el sistema.

2.3.3. MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Hinz, DuBois, y Stephens (2005) mencionan: “MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.”

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

2.3.4. Oracle

“Oracle la Primera Base de Datos Diseñada para Grid Computing, es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general. Oracle Corporation :es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer. (IES San Vicente (2016))

Desarrollado sobre Oracle Database, Oracle Content Database ha sido diseñada para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

2.3.5. Cuadro Comparativo SMBD

SMDB	Ventajas	Desventajas
PostgreSQL	- Multiplataforma. - Diseñado para ambientes de alto volumen. - Herramientas gráficas de diseño y administración. - Tiene mejor soporte que los proveedores comerciales. - Licencia abierta.	-La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva. -Es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tienen otros SMBD. -El soporte es en línea.
MySQL	- Tiene una mayor velocidad al realizar operaciones. - Baja probabilidad de corromper datos. - Característica única puede agrupar transacciones. - Permite la gestión de diferentes usuarios y permisos. - Es Licencia abierta.	-Un gran porcentaje de las utilidades de MySQL no están documentadas. -Poco intuitivo.
Oracle	- Es considerado como uno de los sistemas gestores de bases de datos más completo. - Permite el uso de particiones. - Consultas en paralelo. - Las herramientas de configuración de Oracle son tal vez las mejores en el mercado.	- Un Oracle mal configurado puede ser desesperadamente lento. - Es privativo, costos no muy asequibles.

Tabla 2.1: Tabla comparativa de los Sistemas Manejadores de Base de Datos

2.4. Tecnologías de desarrollo web

A continuación se expondrán las diversas tecnologías de desarrollo web utilizadas para la creación de este proyecto.

2.4.1. HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), (Graham (1995)) hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros.

Los documentos HTML son descritos por las etiquetas HTML, cada etiqueta HTML describe diferentes contenidos de documentos.

2.4.2. CSS

“Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.” (Collison (2007))

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Es usado para definir los estilos de nuestra pagina web, incluyendo diseño, layouts y una variedad de vistas en los diferentes dispositivos y sus tamaños de pagina.

2.4.3. JavaScript

(Flanagan (2011)) define a javascript como un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipeado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.4.4. JQuery

Alvarez (2012) sostiene que: jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.4.5. JQueryUI

Wellman (2011) establece que: jQuery UI es un complemento que permite implementar componentes diversos para generar interfaces de usuario en páginas web, además de otras funcionalidades básicas para crear aplicaciones web enriquecidas. Como su propio nombre indica, está basado en el popular framework Javascript y podemos encontrar links, explicaciones, así como demos y descargas a partir del sitio web oficial de jQuery.

Es una biblioteca de componentes y cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery (find something, manipulate it: encuentra algo, manipúlalo).

2.4.6. Bootstrap

Bootstrap o twitter-bootstrap es un framework creado originalmente por dos desarrolladores/diseñadores de twitter para acelerar el diseño de nuevas aplicaciones web.

“El framework proporciona clases css y código javascript para definir el layout de la página, crear componentes que respondan a eventos y estilizar los elementos html más habituales.” (Otto, Thornton, y cols. (2015))

La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo.

Podemos decir que los principios en los que se basa son:

- Responsive Design: consiste en que la página trata de “hacer lo correcto” al ser visualizada independientemente del dispositivo y tamaño de la pantalla
- Mobile first: Al contrario que en la versión 2, en la 3, el diseño responsivo es la opción por defecto al trabajar con bootstrap
- Cross Browser: Trata de ser compatible con la mayoría de navegadores.
- Integración con jQuery: Está muy integrado con jquery para el que define nuevos plugins
- Buenas prácticas: Trata de emplear algunas de las prácticas más extendidas en cuanto a usabilidad, uso de css3/html5, organización del código

2.4.7. Django

Holovaty y Kaplan-Moss (2009) estipula que: Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como: Modelo-vista-controlador

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.(Holovaty y Kaplan-Moss (2009))

Proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos

Django fue desarrollado por Adrian Holovaty, Simon Willison, Jacob Kaplan-Moss y Wilson Miner mientras trabajaban en World Online, y originalmente se utilizó para administrar tres sitios web de noticias.

Los orígenes de Django en la administración de páginas de noticias son evidentes en su diseño, ya que proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como

parte de cualquier página hecha con Django y que puede administrar varias páginas hechas con Django a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos).

La distribución principal de Django también aglutina aplicaciones que proporcionan un sistema de comentarios, herramientas para syndicar contenido via RSS y/o Atom, "páginas planas" que permiten gestionar páginas de contenido sin necesidad de escribir controladores o vistas para esas páginas, y un sistema de redirección de URLs.

Otras características de Django son:

- Un mapeador objeto-relacional.
- Aplicaciones modulares que pueden instalarse en cualquier página gestionada con Django.
- Una API de base de datos robusta.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URLs basado en expresiones regulares.
- Un sistema "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

2.4.8. CakePHP

Golding (2008) sostiene que: CakePHP es un marco de desarrollo [framework] rápido para PHP, libre, de código abierto. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web. Nuestro principal objetivo es que puedas trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

Con CakePHP el desarrollo web ya no es monótono porque ofrecemos las herramientas para que empieces a escribir el código que realmente necesitas: la lógica específica de tu aplicación. Consigue una copia de CakePHP, empieza con lo verdaderamente importante y no re-inventes la rueda cada vez que te incorpores a un nuevo proyecto.

CakePHP tiene un equipo de desarrolladores y una comunidad activos, lo que añade valor al proyecto. Con CakePHP, además de no tener que re-inventar la rueda, el núcleo de tu aplicación se mejora constantemente y está bien probado.

Esta es una lista breve con las características de las que disfrutarás al utilizar CakePHP:

- Comunidad activa y amistosa
- Licencia flexible
- Compatible con PHP4 y PHP5
- CRUD integrado para la interacción con la base de datos
- Soporte de aplicación [scaffolding]
- Generación de código
- Arquitectura Modelo Vista Controlador (MVC)
- Despachador de peticiones [dispatcher], con URLs y rutas personalizadas y limpias
- Validación integrada
- Plantillas rápidas y flexibles (syntax de PHP, con ayudantes[helpers])
- Ayudantes para AJAX, Javascript, formularios HTML y más
- Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes
- Listas de control de acceso flexibles
- Limpieza de datos
- Caché flexible
- Localización
- Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache

2.4.9. Ruby on Rails

Hansson (s.f.) establece que Ruby on Rails es un entorno de desarrollo web de código abierto que está optimizado para la satisfacción de los programadores y para la productividad sostenible. Te permite escribir un buen código evitando que te repitas y favoreciendo la convención antes que la configuración.

Un conjunto de librerías, automatismos y convenciones destinados a resolver los problemas más comunes a la hora de desarrollar una aplicación web, para que el programador pueda concentrarse en los aspectos únicos y diferenciales de su proyecto en lugar de los problemas recurrentes.

Rails fue creado en 2003 por David Heinemeier Hansson y desde entonces ha sido extendido por el Rails core team, más de 2.100 colaboradores y soportado por una extensa y activa comunidad.

2.4.10. JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999.

Ecma (2015) establece que JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

2.5. El Seguro

“El seguro constituye la forma más perfecta y técnicamente eficaz para la cobertura de riesgos -transformando los individuales en colectivos- y transfiriéndolos a una organización -el asegurador- estructurada con la técnica y operativa adecuadas para garantizar su compensación, en caso de ocurrir el evento.” Albarán (2008)

2.5.1. Contrato del seguro

El contrato de seguro, es aquel contrato mediante el cual una persona llamada asegurador se obliga, a cambio de una suma de dinero, conocida como prima, a indemnizar a otra llamada asegurado o a la persona que este designe, beneficiario, de un perjuicio o daño que pueda causar un suceso incierto. De tal manera que la suma objeto de indemnización, que fue pactada expresamente, sea pagada cuando ocurra el suceso o riesgo cubierto por el seguro.

2.5.2. Funciones del Seguro

Existen diferentes funciones del seguro, entre ellas tenemos:

2.5.2.1. Funciones sociológicas del seguro

- Fomenta la protección de todos los miembros de una familia o individuos.
- Estimula el sentido de responsabilidad frente a terceros, esencial para: abrir nuevas empresas, realizar nuevas inversiones, crear empleo, etc.
- Contribuye a la estabilidad social protegiendo contingencias derivadas de la vejez y enfermedades o accidentes.
- Financia la prevención de riesgos mediante la reducción de primas. Así, aparte de la colaboración del seguro con otros sectores, en el aspecto individual se destaca el espíritu de previsión representado en el interés que muestra en la prevención de las consecuencias desfavorables de un evento.

2.5.2.2. Funciones económicas del seguro

- Contribuye positivamente al desarrollo económico al eliminar riesgos y estabilizar los resupuestos económicos. Por esto, debe desarrollarse paralelamente al resto de las actividades económicas.
- El seguro es la única actividad económica que posee capacidad para generar ahorro y financiación de inversiones a largo plazo.

2.5.2.3. Funciones laborales del seguro

- El seguro participa en la consecución de empleo directo e indirecto. Se estima que en España casi 49.000 familias “viven del seguro” (empleados, agentes, corredores, peritos, liquidadores, abogados, actuarios y otros profesionales) y que el sector está financiando alrededor de 600.000 puestos de trabajo estable.

2.5.3. Póliza de seguro

Albarán (2008) menciona que la póliza es el documento principal del contrato de seguro, en donde constan los derechos y obligaciones de las partes, es un documento privado redactado en varios folios. Las condiciones generales están impresas, mientras las condiciones particulares están normalmente mecanografiadas.

2.5.4. Seguro de Vehículo

Asegurándome (2016) establece que este tipo de seguro puede amparar tanto los daños propios como los daños causados a un tercero.

- Cobertura Amplia: La aseguradora se compromete a indemnizar al Asegurado, en exceso del deducible y hasta la suma asegurada indicada en el Cuadro Póliza, los daños materiales que sufra el Vehículo Asegurado en el período de vigencia de la Póliza, mientras el vehículo se encuentre en circulación, reposo o en transporte dentro de la República Bolivariana de Venezuela, que ocasione un Daño Parcial, General o No Recuperable, salvo aquellos casos expresamente excluidos en esta Póliza.
- Cobertura de indemnización por robo del vehículo: La aseguradora se compromete a indemnizar al Asegurado, en caso de quedar privado del uso del Vehículo Asegurado, como consecuencia directa de Robo o hurto, una cantidad diaria, desde el día en que se hayan cumplido los requisitos establecidos para notificar el reclamo, de acuerdo con lo establecido en las Condiciones Particulares de la póliza, hasta el día en que:
 - a) La Empresa de Seguros haga efectiva la indemnización correspondiente, una vez transcurrido el plazo previsto en las Condiciones Generales de esta Póliza.
 - b) Se haya recuperado el Vehículo Asegurado en caso robo o hurto. De ser el Asegurado quién recupere el Vehículo Asegurado, se compromete a comunicarlo por escrito a la Empresa de Seguros.

Si el Vehículo Asegurado recuperado requiere reparación por daños sufridos a causa del robo o del hurto, la aseguradora continuará indemnizando la cantidad diaria hasta la fecha prevista para la entrega del mismo. En ningún caso, la indemnización superará el

monto de los días o suma asegurada para esta cobertura indicado en el cuadro-recibo de la póliza.

- Cobertura de responsabilidad civil (Daños a terceros): La Aseguradora se compromete a indemnizar la responsabilidad que pudiera corresponder al propietario o conductor del vehículo asegurado, por lesiones o daños a terceros dentro de Venezuela de acuerdo a la ley y reglamento de tránsito vigente hasta los límites indicados en el cuadro de la póliza.
- Asistencia legal y/o defensa penal: Al ocurrir un accidente de tránsito, esta póliza cubre los gastos por asistencia legal para liberar al conductor y vehículo en caso de ser detenidos, así como todos los gastos correspondientes a su defensa penal si es necesario. Está también cubre los casos de demanda en contra del asegurado.
- Cobertura de accidentes personales para ocupantes de vehículos: La Aseguradora se compromete a indemnizar al Asegurado o Beneficiario, la suma asegurada indicada en el Cuadro Póliza, por las lesiones corporales que afecten la integridad física de los Ocupantes del Vehículo Asegurado, a consecuencia de un accidente amparado por la Póliza mientras se encuentren dentro, subiendo o bajando del Vehículo Asegurado y bajo las Condiciones estipuladas en las condiciones de la Póliza.

2.5.5. El Perito

Carlos (2012) estipula que el Perito es esencial en el engranaje de la compañía de seguros, pero para conocer la verdadera dimensión del trabajo del perito, analizamos sus funciones, que se resumen en tres grandes apartados:

2.5.5.1. Aspectos técnicos

- Valoración económica de los daños, elaborando la peritación y realizando la propuesta de indemnización a la compañía de seguros. Determinación del valor del bien asegurado, como, por ejemplo, el valor venal, el valor de mercado, el valor de los restos y la propuesta del importe líquido de la indemnización, cuando se ha producido un siniestro total o una pérdida total.
- Verificación de siniestros, para la realización de informes de uso interno para la compañía de seguros con la justificación técnica de la ocurrencia del siniestro. Pueden ser informes de rehúes parciales o totales, que pueden aportarse como prueba en un juicio. Los informes de reconstrucción de accidentes de tráfico, a partir de huellas y vestigios, mediante cálculos físicos y matemáticos, pueden ser también un apoyo para la determinación de la culpabilidad en el juicio.

- Revisión de riesgos, para la contratación de nuevas pólizas de vehículos de segunda mano con coberturas de daños propios, lunas, etc.
- Control de calidad de la reparación, mediante la comprobación, en primer lugar, de que la reparación se ha llevado conforme a la peritación en todas y cada una de las partidas asignadas por el perito; a continuación, que la reparación se ha realizado con las debidas garantías técnicas, de calidad y seguridad para los ocupantes del vehículo. Por último, se analizarán los defectos en la reparación, para que sean subsanados por el taller.
- Averías mecánicas: valoración y peritación de los daños mecánicos bajo la cobertura de pólizas de vehículos de renting y de pólizas de garantía de venta de vehículos usados.

2.5.5.2. Aspectos administrativo-legales

- Implicación en la tramitación del siniestro. El perito, en contacto con el tramitador y a través del sistema de gestión de la compañía de seguros, está al día de la tramitación de los siniestros, del tipo de pólizas que comercializa la compañía de seguros, de sus coberturas y exclusiones, de los convenios entre compañías y del conocimiento de la legislación de seguros.

2.5.5.3. Aspecto negociador

- El perito es la imagen de la compañía de seguros, ya que está en contacto con los asegurados, perjudicados, talleres, otras compañías, con lo que su actuación está sujeta a examen continuo, y su comportamiento, a ojos del asegurado, es, por extensión, el de la compañía de seguros.
- El perito debe aportar, en todo momento, argumentos y criterios técnicos en la negociación con el taller.
- Ha de consensuar la peritación: debe llegar a acuerdos con el taller sobre todas y cada una de las partidas que componen una peritación.
- Realiza asesoría legal: al estar en contacto con los asegurados y el taller, en muchas ocasiones, el perito se convierte en el asesor sobre los aspectos legales de los siniestros

2.5.5.4. Inspección para asegurar un Vehículo

Guaranty Car (2012) establece que en el proceso de peritaje y/o inspección es muy importante realizar un registro fotográfico del vehículo antes de comenzar a realizar las operaciones de diagnóstico.

Esto se realiza primero, porque es necesario el registro para el informe generado a las aseguradoras y/o el cliente y segundo, para evitar futuras reclamaciones a la hora determinar el proceso de inspección.

También se debe señalar la importancia que cumple la inspección al interior del vehículo, como soporte o elemento de consulta durante la atención de un siniestro para comparar características del vehículo y elementos de contenido.

En resumen el perito esta encargado de realizar diferentes tipos de inspecciones, para un vehículo nuevo en la aseguradora, tales como:

- Registro Fotográfico.
- Inspección Mecánica.
- Inspección Latonería.
- Inspección Pintura.
- Inspección Vidrios.
- Inspección Chasis.
- Inspección Interiores y guarnecidos.

2.5.5.5. Caso de Estudio

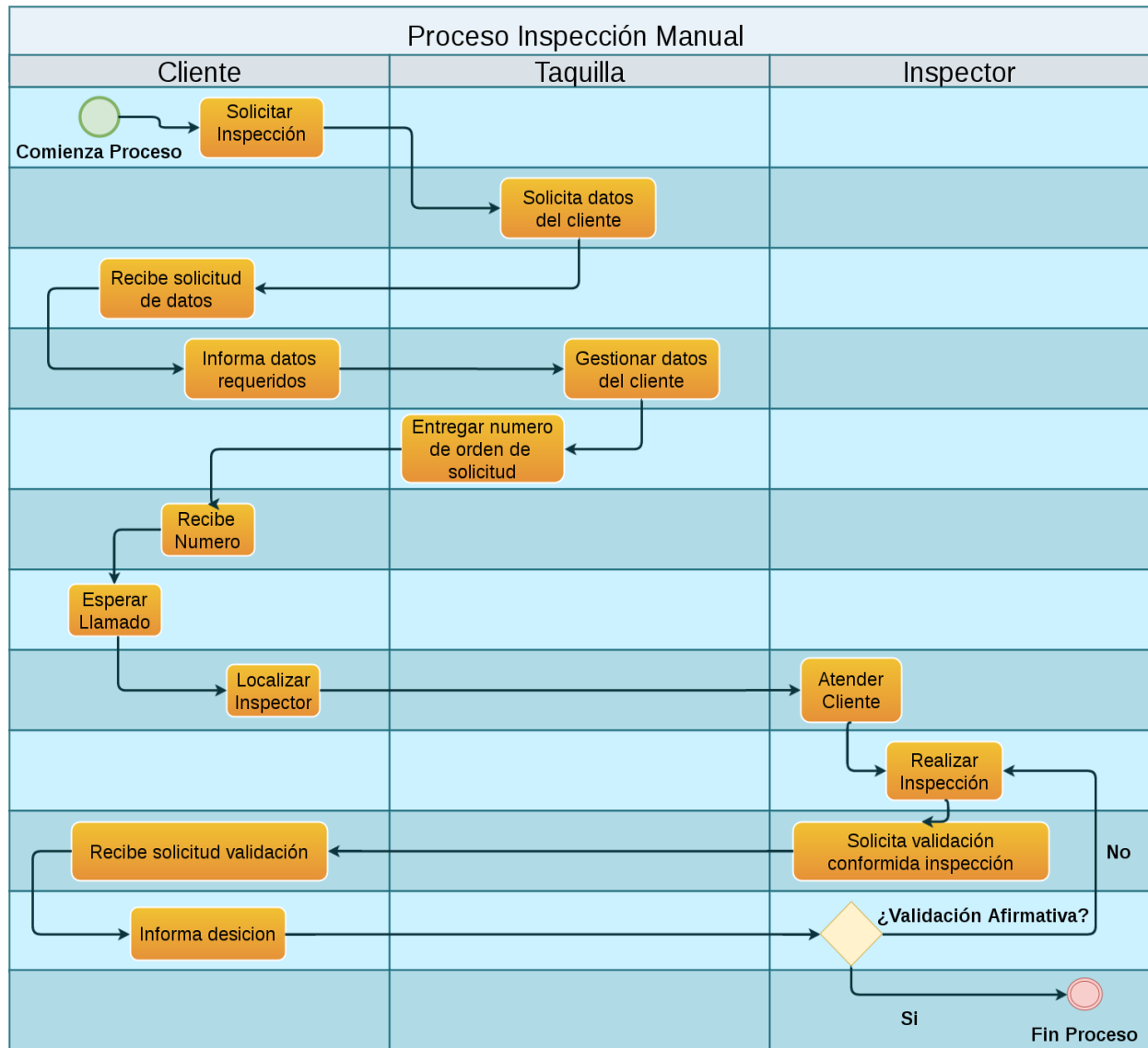


Figura 2.1: Proceso Actual. (2016)

2.5.5.6. Planilla de Suscripción

INFORME DE INSPECCIÓN AUTOMÓVIL										Fecha de Elaboración: 05-10-2015										
El presente formulario deberá ser llenado en todos sus puntos, con la letra de imprenta, legible																				
MOTIVO DE LA SOLICITUD DE INSPECCIÓN																				
COBERTURA AMPLIA				PÉRDIDA TOTAL				AUMENTO DE SUMA ASEGURADA				INCLUSIÓN DE ACCESORIOS								
INCLUSIÓN DEL CASCO				REPARACIÓN				PLAZO DE GRACIA VENCIDO				CAMBIO DE VEHÍCULO								
DATOS DEL VEHÍCULO																				
1. Nombres y Apellidos del Titular: ORIEL HERÁNDEZ						2. Cédula de Identidad: 21123456														
4. Nombres y Apellidos de la Persona que trajo el Vehículo a Inspección: JESUS HERNÁNDEZ						3. Teléfono: 0212-1234567														
7. Marca: AUDI						5. Cédula de Identidad: 0														
8. Modelo: A3						6. Parentesco con el Titular: HERMANO														
13. Año: 2005						11. Color:														
14. Placa: SAV138						12. Tipo: Sedan X Carga Coupe														
15. Peso: 0						16. Automático:														
19. Señal de Carrocería: 98D15824014259323						17. Kilometraje: 0 Km.														
20. Valor Estimado: 35000 Bs.						18. Señal del Motor:														
21. Nombre del Inspector:						22. C.I. N° 0														
CONDICIONES GENERALES DEL VEHÍCULO																				
PARTES			B	R	M	OBSERVACIONES	PARTES			B	R	M	OBSERVACIONES							
23. Latorería:							24. Parilla O Rejilla Frontal:			X										
26. Pintura:			X				27. Stop O Focos Traseros:			X										
29. Tapicería:			X				30. Cocuyos Frontales:			X										
32. Cauchos:			X				33. Retrovisor Interno:			X										
35. Platinas:			X				36. Manillas Y Cerraduras:			X										
38. Tableros:			X				39. Limpia Parabrisas:			X										
41. Faros Y Sus Aros:			X				42. Retrovisores Externos:			X										
44. Emblema De Modelo:			X				45. Guardafangos:				X									
47. Estibos:			X				48. Puertas:			X										
50. Frontal:				X		Abolladura	51. Capot:				X									
53. Modific. Del Modelo:			X				54. Carter Trasero:			X										
25. Techo:				X			28. Compuerta Trasera:			X										
31. Parabrisas:							34. Parachoques:			X										
37. Corneta:							40. Butacas Y Asientos:				X		No Posee							
43. Volantes:							46. Cinturones De Seguridad:			X										
49. Ventanas:							52. Maleta Y Tapa Maleta:					X								
55. Otro:							56. Sistema Eléctrico:			X										
57. Embrague O Croch:				X		Mal Estado	58. Sistema Eléctrico:			X										
59. Otros:			X				60. Otros:				X									
ACCESORIOS DEL VEHÍCULO																				
ACCESORIOS			SI	NO	OBSERVACIONES	ACCESORIOS			SI	NO	OBSERVACIONES	PARTES			SI	NO	OBSERVACIONES			
62. Radio: Antena: SI				X		63. Foco De Posición:				X		64. Rines Especiales:				X				
65. Radio Reproductor:				X		66. Aire Acondicionado:			X			67. Caucho Repuesto:				X				
68. Reproductor De Antena:				X		69. Spoiler:				X		70. Parachoques Espec. (malaburo):			X					
71. Cornetas Especiales:				X		72. Llave De Cruz:				X		73. Rayas Decalativas:				X				
74. Compact Disc:			X			75. Triángulo De Seguridad:				X		76. Tazas De Los Cauchos:				X				
77. Ecualizador:			X			78. Parilla Techo:			X			79. Volante Especial:				X				
80. Televisor:				X		81. Cabina:			X			82. Gato:				X				
83. Planta:			X			84. Faros Especiales:			X			85. Anti-robo:				X				
86. Tranca Pedal:				X		87. Thermo King:				X		88. Otro:				X				
89. Conforter:			X			90. Tranca Palanca:				X		91. Otro:				X				
(*) Indicar Marca y Modelo																				
DETALLES DE DATOS												DOCUMENTOS PRESENTADOS PARA INSPECCIÓN								
92. PIEZA						93. TIPO DE DAÑO			94. COSTO APROX.			95. Registro Automotor Permanente(p.a.):						SI	NO	X
000238 Absorbedor De Parachoques Trasero						Abollado			0			96. Camet De Circulación:						SI	X	NO
000436 Asiento Copiloto												97. Documento De Propiedad (original/copia):						SI	X	NO
												98. Factura De Compra:						SI	X	NO
												99. Planilla Del Seniat De Exoneración De Impuestos:						SI	NO	X
												100. Certificado De Origen:						SI	NO	X
101. Observaciones: LA INSPECCIÓN NO PRESENTA OBSERVACIONES ACTUALMENTE																				
DERECHA FRENTE POSTERIOR IZQUIERDA ☐ BUENO ☐ REGULAR ☐ MALO																				
NOTA: La inspección del vehículo descrito anteriormente, no implica aceptación del riesgo por parte de Horizonte S.A. Los riesgos que asume la Empresa correrán por cuenta de la misma, desde el momento en que el asegurado haya cancelado la prima convenida. Yo, el Inspector declaro que: No he omitido y/o simulado, ningún hecho o circunstancia en las respuestas, que puedan modificar las condiciones del riesgo y que los datos indicados fueron tomados del vehículo descrito en el presente formulario y en las fotografías respectivas. Yo, el Asegurado y/o Solicitante declaro que: a) En caso de fraude o declaración falsa por mi parte, la presente inspección es NULA, siendo de igual manera NULO el Contrato que la misma respalda, censando en cuyo caso toda responsabilidad de la empresa. b) De igual manera certifico todos y cada uno de los SERIALES y NUMERACIONES Y CONDICIONES indicadas por el inspector, en el presente formulario, como los correspondientes a mi (el) vehículo.																				
Firma Inspector de Riesgos												Firma Asegurado o Solicitante								
Sucursal, Agencia o Unidad de Asesoramiento: SERVICIOS GENERALES												Fecha:								
Firma Intermediario de Seguros												Código:								

SH-GVT-FO-0020(08/2013)

Figura 2.2: Proceso de Suscripción. (2016)

Capítulo 3

Marco Metodológico

En el capítulo a continuación se explicaran las diferentes metodologías para el desarrollo de un producto de software, las cuales se encuentran clasificadas en dos tipos Metodologías Tradicionales y Metodologías Ágiles. Posteriormente, se explicara cual fue la metodología utilizada para el desarrollo del proyecto y su implementación.

3.1. Metodología Tradicional

3.1.1. RUP

Jacobson, Rumbaugh, Jacobson, Booch, y Rumbaugh (2000) establece que RUP es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto, con 3 características esenciales, está dirigido por:

- Los Casos de Uso : que orientan el proyecto a la importancia para el usuario y lo que este quiere.
- La arquitectura: que Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- Iterativo e incremental: dividiéndose el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

3.1.1.1. Ciclo de vida

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades

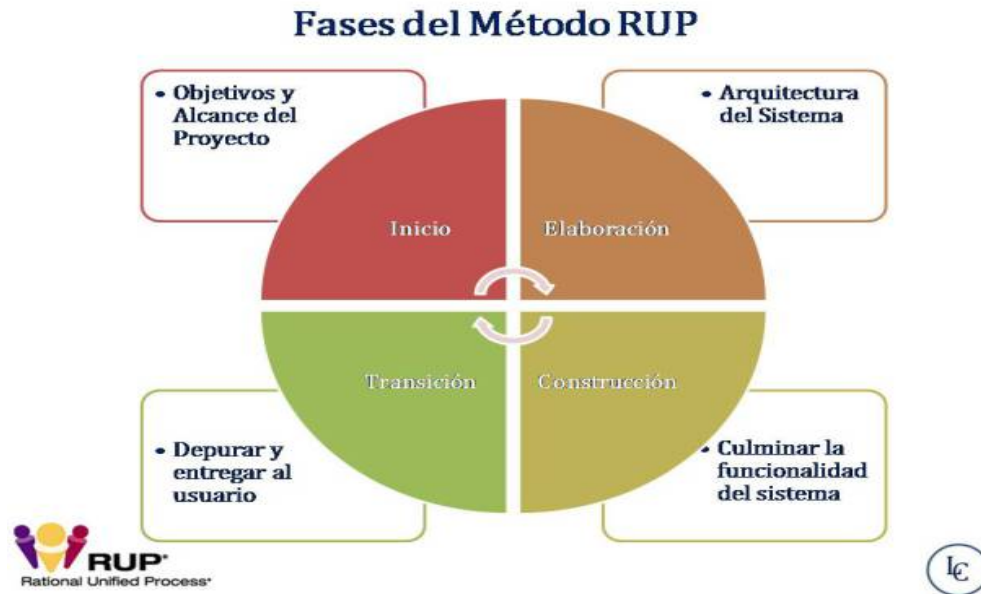


Figura 3.1: Proceso RUP. (Tomada de DTyOC (2016))

- Fase de Inicio: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
- Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
- Fase de Desarrollo: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
- Fase de Transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

3.1.2. Cascada

Red (2013) define el Modelo en Cascada, también llamado Lineal secuencial, como aquel cuyo enfoque metodológico ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

3.1.2.1. Fases del Modelo

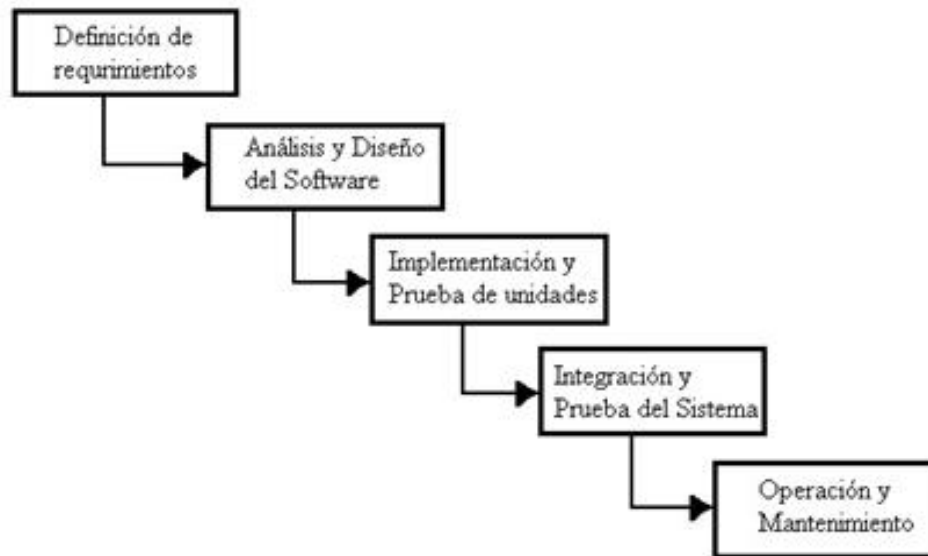


Figura 3.2: Proceso Cascada. (Tomada de Wiki Uploads (2013))

- **Análisis de requisitos:** En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.
- **Diseño del Sistema:** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.
- **Diseño del Programa:** Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

- **Codificación:** Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.
- **Pruebas:** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.
- **Verificación:** Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.
- **Mantenimiento:** Una de las etapas mas criticas, ya que se destina un 75 de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

3.1.3. Cuadro Comparativo

RUP	Casacada
Es un proceso iterativo.	Es un proceso secuencial.
Desarrolla el producto en base al feedback de los accionistas.	El proceso de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba.
Cada iteracion produce una versión ejecutable.	Hasta que el software no esté completo no se opera. Esto es la base para que funcione bien.
Es un marco adaptable de procesos de software	Es un proceso concreto definido.

Tabla 3.1: Tabla comparativa de metodologías tradicionales

3.2. Metodología Ágil

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado

mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

3.2.1. SCRUM

Schwaber y Beedle (2002) establecen que el Scrum es un proceso de la Metodología Ágil que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa.

Entre las ventajas se encuentran la productividad, calidad y que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes estén unidos, comunicados y que el cliente vaya viendo los avances.

3.2.1.1. Ciclo de vida

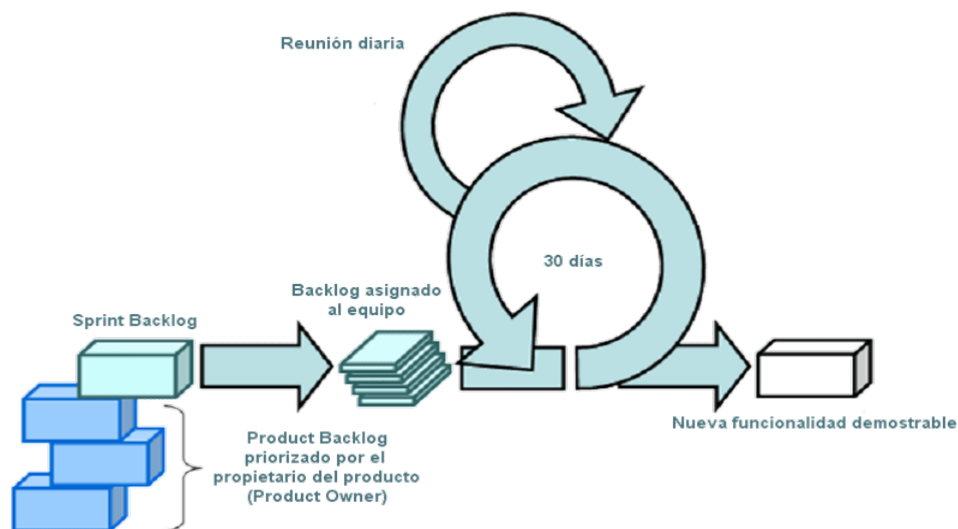


Figura 3.3: Proceso Scrum. (Tomada de Methods & Tools (2004))

3.2.1.2. Spring

Es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad dura máximo 30 días. Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas “carreras”. Durante el proceso no se puede modificar el trabajo que se ha acordado en el Backlog, solo es posible cambiar el curso de un sprint abortándolo y esto solo puede hacerlo el Scrum Master.

3.2.1.3. Planificación del sprint

En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint. Se trata de una reunión conducida por el responsable del funcionamiento del marco scrum (Scrum Master en scrum técnico, o un miembro del equipo, en scrum pragmático) a la que deben asistir el propietario del producto y el equipo completo, y a la que también pueden asistir otros implicados en el proyecto.

La reunión puede durar una jornada de trabajo completa, cuando se trata de planificar un sprint largo (de un mes de duración) o un tiempo proporcional para planificar un sprint más breve. Esta reunión debe dar respuesta a dos cuestiones:

- Qué se entregará al terminar el sprint.
- Cuál es el trabajo necesario para realizar el incremento previsto, y cómo lo llevará a cabo el equipo.

La reunión se articula en dos partes de igual duración, para dar respuesta a una de estas cuestiones, en cada una.

3.2.1.4. Scrum diario

Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las 24 horas siguientes. Informa el avance de cada miembro del equipo e identifica posibles necesidades e impedimentos.

3.2.1.5. Revisión de las Iteraciones

Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Es por tanto la duración del sprint, el período de tiempo máximo para descubrir planteamientos erróneos, mejorables o malas interpretaciones en las funcionalidades del producto.

3.2.1.6. Retrospectiva

Reunión que se realiza tras la revisión de cada sprint, y antes de la reunión de planificación del siguiente, con una duración recomendada de una a tres horas, según la duración del sprint terminado. En ella el equipo realiza autoanálisis de su forma de trabajar,

e identifica fortalezas y puntos débiles. El objetivo es consolidar y afianzar las primeras, y planificar acciones de mejora sobre los segundos.

El objetivo de la revisión del sprint es analizar “QUÉ” se está construyendo, mientras que una reunión retrospectiva se centra en “CÓMO” lo estamos construyendo: “CÓMO” estamos trabajando, con el objetivo de analizar problemas y aspectos mejorables.

Las reuniones retrospectivas realizadas de forma periódica por el equipo para mejorar la forma de trabajo, se consideran cada vez más un componente del marco técnico de scrum, si bien no es una reunión para seguimiento de la evolución del producto, sino para mejora del marco de trabajo. (Juan Palacio, 2014)

3.2.2. XP

Robles y Ferrer (2002) definen a XP como una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos, muy cambiantes, y donde existe un alto riesgo técnico.

3.2.2.1. Ciclo de vida

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se

perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases:



Figura 3.4: Proceso Scrum. (Tomada de Javier Cifuentes (2004))

3.2.2.2. Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

3.2.2.3. Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por

iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

3.2.2.4. Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Wake en proporciona algunas guías útiles para realizar la planificación de la entrega y de cada iteración.

3.2.2.5. Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

3.2.2.6. Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

3.2.2.7. Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

3.2.3. Agilus

Acosta (2011) establece que el Método AgilUs es un método de desarrollo ágil, resultado de una de las líneas de investigación desarrolladas en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, Universidad Central de Venezuela. Se basa en el concepto de usabilidad, en la necesidad de desarrollar software usables. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin de que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo.

AgilUs es un método de desarrollo iterativo e incremental que pone el mayor peso del desarrollo en la consecución de la usabilidad. Se centra en que la construcción y desarrollo de las interfaces de usuario no debe ser una adición estética que se da al final del desarrollo del sistema sino, muy por el contrario, el desarrollo de interfaces de usuario debe guiar las decisiones en Ingeniería de Software. En AgilUs son los usuarios, y no el cliente ni los programadores quienes guían el desarrollo del proyecto.

El Método AgilUs busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un producto de software. El proceso de desarrollo de software engloba las actividades de requisitos, análisis, prototipaje y entrega; así como las evaluaciones de usabilidad correspondientes a cada etapa del proceso. Se realizan en ciclos iterativos hasta alcanzar el producto final. En cada etapa del proceso de desarrollo de software, se incluyen actividades propias para la construcción de la usabilidad.

3.2.3.1. Ciclo de vida

El ciclo de vida de AgilUs hace énfasis en la importancia del usuario y sus evaluaciones. Está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega. Este producto final puede ser posteriormente modificado a través de un mantenimiento correctivo y/o evolutivo, que no está contemplado como parte del método.



Figura 3.5: El método AgilUs: etapas, actividades y artefactos.

- **Requisitos:** Se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del software, ya que un mal análisis de requisitos traería como consecuencia un software que no cumple con las necesidades del usuario.
- **Análisis:** Se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.
- **Prototipaje:** Se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, se genera la guía de estilo, y se realizan evaluaciones de usabilidad apropiadas a esta etapa: las evaluaciones heurísticas y las listas de comprobación.
- **Entrega:** Se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación.

3.2.4. Cuadro Comparativo

Scrum	XP	Agilus
Las iteraciones de entrega son de dos a cuatro semanas y se conocen como sprint.	Las iteraciones de entrega son de una a tres semanas (algo más rápidas).	Las iteraciones de entrega son de una a dos semanas aproximadamente.
Al finalizar un sprint, las tareas que se han realizado del Sprint Backlog y en las que el Product Owner ha mostrado su conformidad ya no se vuelven a tocar en ningún momento. "Lo que se terminar, funciona y está bien, se aparta y ya no se toca".	Las tareas que se van terminando en las diferentes entregas al cliente son susceptibles a modificaciones durante el transcurso de todo el proyecto, incluso después de que funcionen correctamente.	El equipo de desarrollo produce un primer prototipo y a partir de las observaciones, retroalimentación se produce el siguiente.
El scrum team trata de seguir el orden de prioridad que marca el Product Owner en el Sprint Backlog pero si ven que es mejor modificar el orden de prioridad para el desarrollo de tareas, pueden hacerlo.	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente (aunque el equipo de desarrollo le ayude a decidir, ellos son los que mandan).	El equipo de desarrollo sigue estrictamente el orden de prioridad definido para cada una de las fases.
El Scrum es una metodología de desarrollo ágil más basada en la administración del proyecto.	En cambio, el XP se centra más en la propia programación o creación del producto.	Agilus en cambio se centra más el desarrollo en la usabilidad del producto de software.
Cada miembro del "Scrum Team" trabaja de forma individual.	Los miembros programa en parejas en un proyecto de XP.	Los miembros del equipo trabajan en conjunto para completar cada fase de desarrollo.

Tabla 3.2: Tabla comparativa de metodologías ágiles

3.3. Cuadro comparativo [Metodologías Tradicionales vs Ágiles]

Tradicionales	Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo. Posee cierta resistencia a los cambios.	Basadas en heurísticas provenientes de prácticas de producción de código las cuales previenen cambios durante el proyecto.
Define un proceso mucho más controlado junto con numerosas políticas/normas.	Define un proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Genera más artefactos.	Genera pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (¡10 integrantes) y trabajando en el mismo sitio.
Dependencia de la arquitectura de software mediante modelos.	Menor dependencia de la arquitectura de software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

Tabla 3.3: Diferencias entre metodologías ágiles y tradicionales

Capítulo 4

Marco Aplicativo

A continuación, para el siguiente capítulo se presenta una descripción general de la solución en conjunto con las respectivas tecnologías utilizadas para su elaboración. Además se detalla la metodología de desarrollo SCRUM que se aplicó para la elaboración de este proyecto, así como los diferentes requerimientos, perfiles de usuarios que interactúan con el sistema, explicación de las interfaces de la aplicación y las pruebas realizadas que contribuyeron a un mejor desempeño y una mayor aceptación de la aplicación.

4.1. Descripción general de la solución

En el presente trabajo especial de grado se desarrolló un sistema con tecnologías web, en el cual los usuarios pueden gestionar las solicitudes de inspección, tanto generar las solicitudes, registrar los resultados de la inspección y convalidar las mismas. Logrando así una persistencia digital de las solicitudes para una mejor administración de la información.

Para ello se propuso el desarrollo de la siguiente arquitectura con las siguientes tecnologías:

4.2. Aplicación de la metodología Scrum

A continuación, en la siguiente Tabla se muestran las actividades programadas para los Sprint en pro del desarrollo del sistema.

4.2.1. Lista de Objetivos (Pila del Producto)

Sprint	Actividad	Fecha
1.	■ Instalación de las herramientas de desarrollo necesarias. ■ Configuración de los ambientes de desarrollo. ■ Maquetado de las Interfaces ■ Elaboración del Diagrama de Modelo relacional para la base de datos.	01/02/2017 - 14/02/2017
2.	■ Desarrollo del modulo de registro. ■ Realización de pruebas de funcionalidad.	15/02/2017 - 28/02/2017
3.	■ Desarrollo de la funcionalidad de recuperar contraseña ■ Desarrollo del modulo de gestión de solicitudes ■ Desarrollo de la funcionalidad de solicitar inspección ■ Realización de pruebas de funcionalidad	01/03/2017 - 15/03/2017

Tabla 4.1: Pila del Producto. Fuente: El Autor.

Sprint	Actividad	Fecha
4.	▪ Desarrollo de la funcionalidad de gestión de tickets de solicitudes de inspección ▪ Desarrollo de la funcionalidad de consultar solicitudes de inspección ▪ Realización de pruebas de funcionalidad	16/03/2017 - 31/03/2017
5.	▪ Desarrollo de la funcionalidad de realizar inspección ▪ Desarrollo de la funcionalidad de gestionar inspección ▪ Desarrollo de la funcionalidad de generación de pdf ▪ Realización de pruebas de funcionalidad	01/04/2017 - 15/04/2017
6.	▪ Desarrollo del modulo administrativo ▪ Desarrollo de la funcionalidad de Gestión de Usuarios (Taquilla Inspectores) ▪ Realización de pruebas de funcionalidad	16/04/2017 - 30/04/2017
7.	▪ Desarrollo de la funcionalidad de Gestión de Titulares de Vehículo ▪ Desarrollo de la funcionalidad de Gestión de Personas que trajeron el Vehículo a la inspección ▪ Realización de pruebas de funcionalidad	01/05/2017 - 15/05/2017
8.	▪ Desarrollo del Vehículo en formato SVG ▪ Realización de pruebas de funcionalidad	16/05/2017 - 30/05/2017

Tabla 4.2: Pila del Producto. Fuente: El Autor.

4.2.2. Lista de tareas de la iteración (Pila del Sprint)

-Sprint 1: Se comenzó con la instalación de las herramientas necesarias para el desarrollo de la aplicación, las cuales fueron Python 2.7.9, PostgreSQL 9.4.3 y Django 1.10.

Además se realizó el diseño y maquetado de la aplicación, a partir de los componente provistos por el Framework Bootstrap 4 y HTML5 como base para el desarrollo de las interfaces de usuario. También se realizó el modelado de la base de datos en conjunto con los diagramas asociados al sistema. Finalmente se instalaron todas las dependencias necesarias en un ambiente virtual para la realización del sistema.

-Sprint 2: En este Sprint se desarrollaron parte de las funcionalidades pertinentes al modulo de registro, con sus validaciones respectivas para el correcto funcionamiento del sistema de registro. Luego se realizó las pruebas de funcionalidad correspondientes para validar el funcionamiento esperado del sistema.

-Sprint 3: En este Sprint se terminaron de desarrollar las funcionalidades restantes del modulo de registro, se empezó el desarrollo del modulo de gestión de solicitudes, en el cual se implementaron parte de las funcionalidades tales como el solicitar una inspección. Además se realizaron las pruebas pertinentes para el correcto funcionamiento de la aplicación.

-Sprint 4: Se realizaron las funcionalidades de gestión de tickets de solicitud de inspección y consultar solicitudes de inspección, en esta ultima se obtuvieron las especificaciones de los campos por los cuales se filtrarían dichas solicitudes. Luego se realizaron las pruebas necesarias para validar el correcto funcionamiento de las funcionalidades desarrolladas.

-Sprint 5: En este Sprint se desarrollaron las funcionalidades de realizar inspección y la gestión de las mismas, además de la generación de la planilla en la cual se registran todos los datos de las especificaciones del vehículo e información relevante de la inspección. Se realizaron las pruebas funcionales para validar el correcto funcionamiento.

-Sprint 6: En este Sprint se desarrollo parte del modulo de administración, en especifico la funcionalidad de gestión de usuarios que comprende en editar, inactiva y eliminar usuarios. Además se realizaron las pruebas pertinentes para garantizar la correcta funcionalidad.

-Sprint 7: Se desarrollaron las funcionalidades restantes del modulo de administración, gestión de los titulares del vehículo y de las personas que trajeron el vehículo a inspeccionar. Seguidamente se realizaron las pruebas funcionales para validar un correcto funcionamiento.

-Sprint 8: Se desarrollo la imagen del carro en formato SVG y se validaron las partes pertinente para que cambiaran de color dependiendo del estado del vehículo. Se realizaron las pruebas correspondientes para finalizar la planilla de inspección.

4.3. Requerimientos del Sistema

De acuerdo a los requerimientos definidos por el cliente (Product Owner), se estableció la siguiente clasificación:

-Requerimientos Funcionales:

- Registro de usuarios por roles.
- Recuperar contraseña.
- Edición del perfil de usuario.
- Administración de los usuarios del sistema.
- Administración de los clientes del sistema (Titular, Persona que trajo el Vehículo).
- Creación de tickets para la inspección del vehículo.
- Consultar tickets creados.
- Consultar solicitudes de inspección.
- Registro de los datos de la inspección del vehículo.
- Gestionar solicitudes de inspección.
- Creación de una imagen de carro donde se muestren diferentes colores, para las partes del carro, dependiendo su estado.

-Requerimientos no Funcionales:

- Validar las entradas de los datos para el correcto funcionamiento del sistema.

4.4. Perfiles de Usuario

-Administrador: Es el usuario encargado de gestionar tanto usuarios como a los clientes y propietarios de los vehículos inspeccionados.

- Búsqueda de usuarios del sistema.

- Búsqueda de titulares de los vehículos.
- Búsqueda de personas trajeron el vehículo a revisión.
- Edición de titulares de vehículos.
- Edición de personas trajeron el vehículo a revisión.
- Edición y desactivación de usuarios Inspectores.
- Edición y desactivación de usuarios Taquilla.
- Reinicio de usuarios enviado les nuevas contraseñas.

-Inspector: Este usuario puede crear una cuenta en el sistema para gestionar solicitudes de inspección de vehículo. También puede editar la información de su cuenta, ver la información de las solicitudes cerradas.

- Creación y edición de una cuenta con su información básica para el uso del sistema.
- Visualización de las solicitudes de inspección pendientes por realizar.
- Vaciado de la información de la inspección del vehículo en el sistema.
- Gestión de la solicitud de inspección (verificación y guardado).
- Visualizar información de las solicitudes de inspección.
- Búsqueda de solicitudes de inspección realizadas.

-Taquilla: Este usuario puede crear una cuenta en el sistema para gestionar los tickets de atención, de las persona que requieran realizar una inspección.

- Creación y edición de una cuenta con su información básica para el uso del sistema.
- Creación de ticket para solicitud de inspección.
- Cancelar un ticket abierto de una solicitud de inspección
- Búsqueda de tickets abiertos

4.5. Descripción del flujo asociado a la solución

4.5.1. Proceso de creación de solicitud

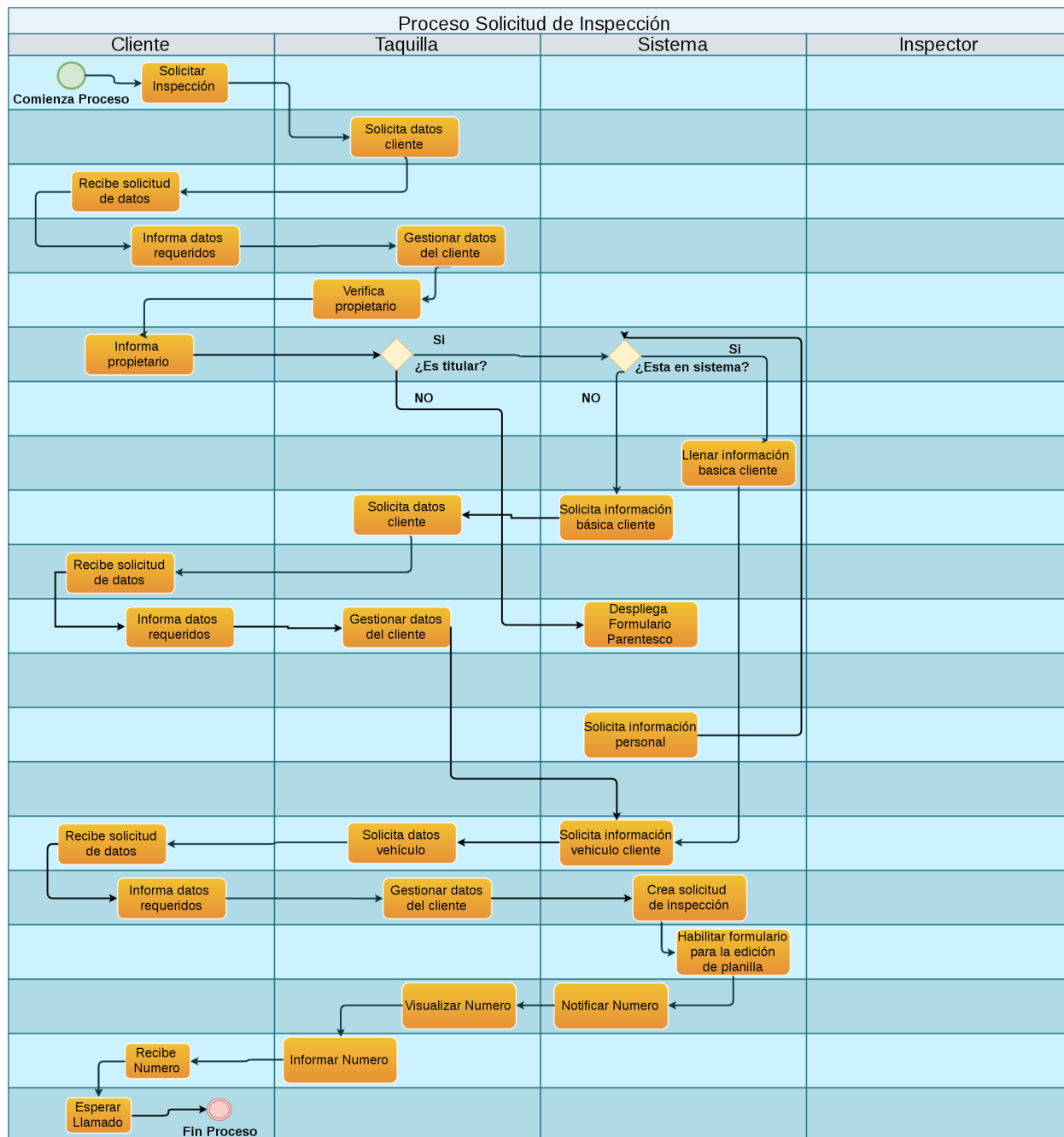


Figura 4.2: Proceso de creación de solicitud de inspección. Fuente:El Autor.

De acuerdo a la Figura anterior, cuando un cliente desea realizar una solicitud de inspección, este indica su numero de orden al personal de la Taquilla, esta persona se encarga de llenar el formulario para la apertura de una solicitud de inspección, llenando el numero de orden (numero de solicitud), placa del carro, motivo de la visita, cédula, nombre, apellido y numero de teléfono, en caso de que el titular no este presente se llenan los datos de la persona que trajo el vehículo a inspeccionar. Luego de ello esta solicitud es guardada y el cliente debe esperar a que un perito lo llame para realizar la inspección.

Listado de excepciones:

- Si el usuario ingreso algún dato erróneo, puede cancelar dicha solicitud y apertura una nueva.
- Si el cliente tuvo alguna emergencia o impedimento para realizar la inspección dicha solicitud puede ser cancelada.

4.5.2. Proceso de gestión de solicitud de inspección

Este proceso esta dividido en dos partes, las cuales se muestran en la figura a continuación.

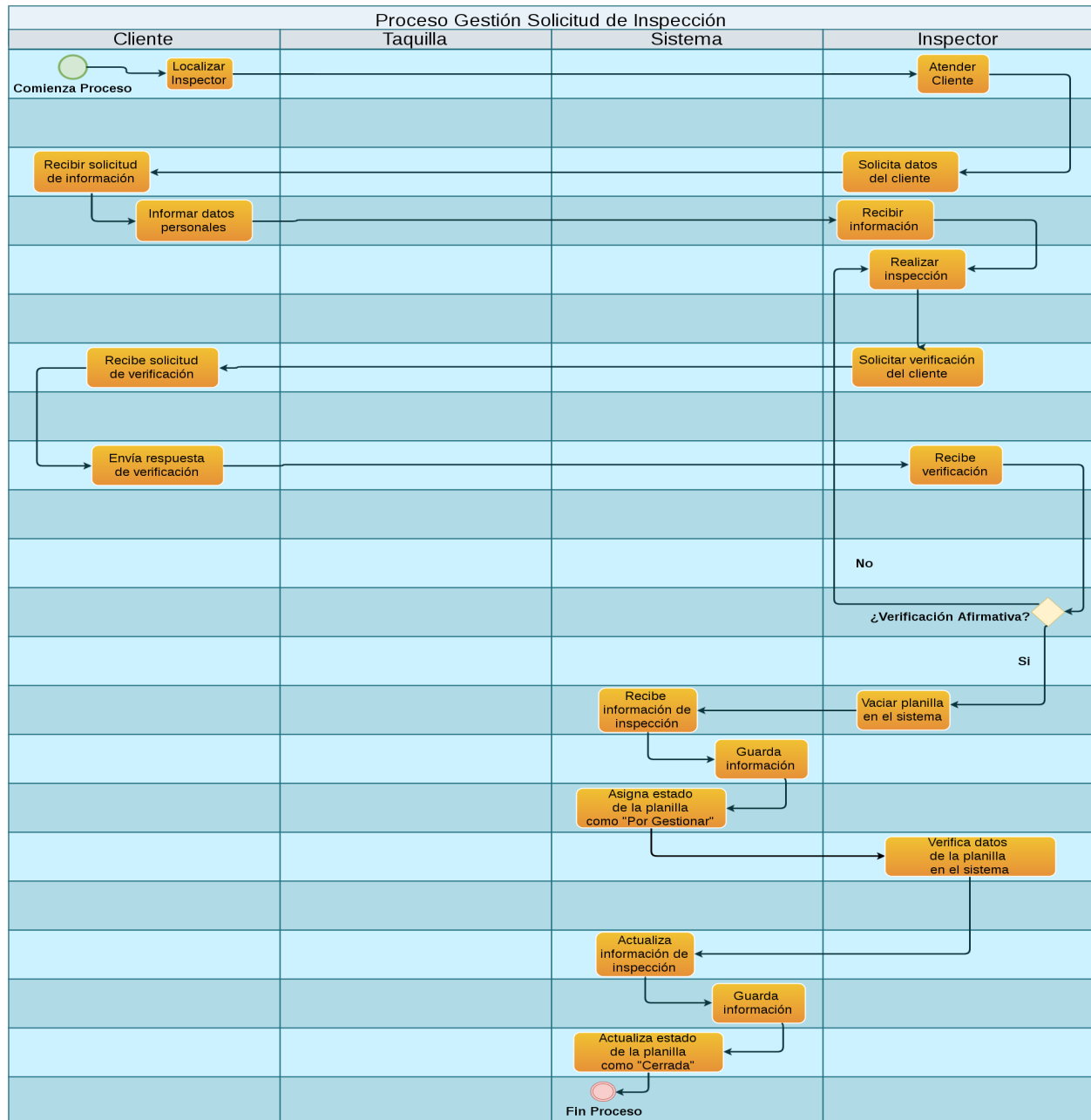


Figura 4.3: Proceso de gestión de solicitud de inspección. Fuente:El Autor.

Cuando un usuario Inspector, desea registrar los resultados de una inspección, este se dirige al sistema, ingresa a la bandeja de solicitudes y elije la cual acaba de completar el perito registra los datos del vehículo, sus detalles mecánicos, accesorios, condiciones generales y documentos presentados al momento de la inspección. Luego de esto valida la información con el cliente y este al dar el visto bueno la solicitud pasa a estado por gestionar, en caso de alguna disconformidad con el cliente se realiza el proceso de inspección hasta que el cliente este satisfecho con la inspección. Mas adelante esta solicitud es gestionada, revisando los datos registrados y validando que todo este correcto, en caso de querer hacer un cambio se puede realizar, luego de ello el Inspector verifica los datos una ultima vez antes de cerrar la solicitud.

4.6. Análisis del modelo de datos y definición

En la elaboración del modelo de datos pertenecientes a la aplicación se crearon un total de veintiocho (31) tablas.

4.6.1. Listado de tablas de la aplicación

A continuación se presenta un listado con todas las tablas de la base de datos perteneciente al sistema desarrollado, junto con una breve descripción para cada una:

- registro_status:** Permite clasificar el estado del usuario (Activo, Inactivo).
- registro_tipousuario:** Permite clasificar a los usuarios por tipo (Administrador, Cliente, Taquilla).
- registro_usuario:** Permite almacenar los usuarios en el sistema.
- racs_accesoriosbase:** Permite almacenar los accesorios básicos de la planilla.
- racs_accesoriosvehiculo:** Tabla utilizada para representar los accesorios base de un vehículo.
- racs_condicionesgeneralesbase:** Permite almacenar las condiciones generales básicas de la planilla.
- racs_condicionesgeneralesvehiculo:** Tabla utilizada para representar las condiciones generales base de un vehículo.
- racs_detalle_datos:** Permite almacenar los detalles y datos de la planilla.

-rcs_documentospresentados: Permite almacenar los documentos presentados básicos de la planilla.

-rcs_documentospresentadosbase: Tabla utilizada para representar los documentos presentados base de un vehículo.

-rcs_estadosolicitud: Permite almacenar los diferentes estado que puede poseer la planilla (Abierta, Por Revisar,Cerrada).

-rcs_estadovehiculo: Permite almacenar los diferentes estado que puede poseer las partes del vehículo (Bueno, Regular,Malo).

-rcs_mecanicabase: Permite almacenar las mecánicas básicas de la planilla.

-rcs_mecanicavehiculo: Tabla utilizada para representar las mecánicas base de un vehículo.

-rcs_motivosolicitud: Permite almacenar los diferentes motivos por lo cual se realiza la solicitud de inspección.

-rcs_solicitudinspeccion: Permite almacenar las solicitudes de inspección realizadas por el sistema.

-rcs_tipomanejo: Permite almacenar los diferentes tipos de manejo (Automático, Sincrónico).

-rcs_tipovehiculo: Permite guardar la información de los diferentes tipos de vehículos (Sedan, Carga, Coupe)

-rcs_titularvehiculo: Permite almacenar la información referente a los titulares de los vehículos.

-rcs_trajovehiculo: Permite almacenar la información de las personas que trajeron el vehículo a inspección en lugar del titular del titular.

-rcs_vehiculo: Permite almacenar la información referente al vehículo.

-rcs_vehiculo_accesorios_vehiculo: Tabla utilizada para representar las relaciones Many-ToMany de rcs_accesoriosvehiculo y vehículo.

-rcs_vehiculo_condiciones_generales_vehiculo: Tabla utilizada para representar las relaciones ManyToMany de rcs_condicionesgeneralesvehiculo y vehículo.

-rcs_vehiculo_detalle_datos: Tabla utilizada para representar las relaciones ManyToMany de rcs_detalle_datos y vehículo.

-rcs_vehiculo_documentos_presentados: Tabla utilizada para representar las relaciones ManyToMany de rcs_documentos_presentados y vehículo.

-rcs_vehiculo_mecanica_vehiculo: Tabla utilizada para representar las relaciones ManyToMany de rcs_mecanica_vehiculo y vehículo.

-rcs__mecanica_vehiculo: Tabla utilizada para representar las relaciones ManyToMany de rcs_mecanica_vehiculo y vehículo.

-rcs__modelovehiculo: Permite guardar la información de los diferentes modelos de vehículos.

-rcs__marcavehiculo: Permite guardar la información de las diferentes marcas de vehículos.

-rcs__parentesco: Permite guardar la información de los diferentes parentescos de la persona que trajo el vehículo a la inspección utilizando el sistema esquimal de parentesco.

-django_migrations: Permite almacenar y llevar un control de las migraciones aplicadas para cambiar la base de datos a lo largo del proceso de desarrollo.

-django_session: Permite almacenar todos los valores usados en la sesión de los usuarios dentro de la aplicación.

4.6.2. Modelo de datos

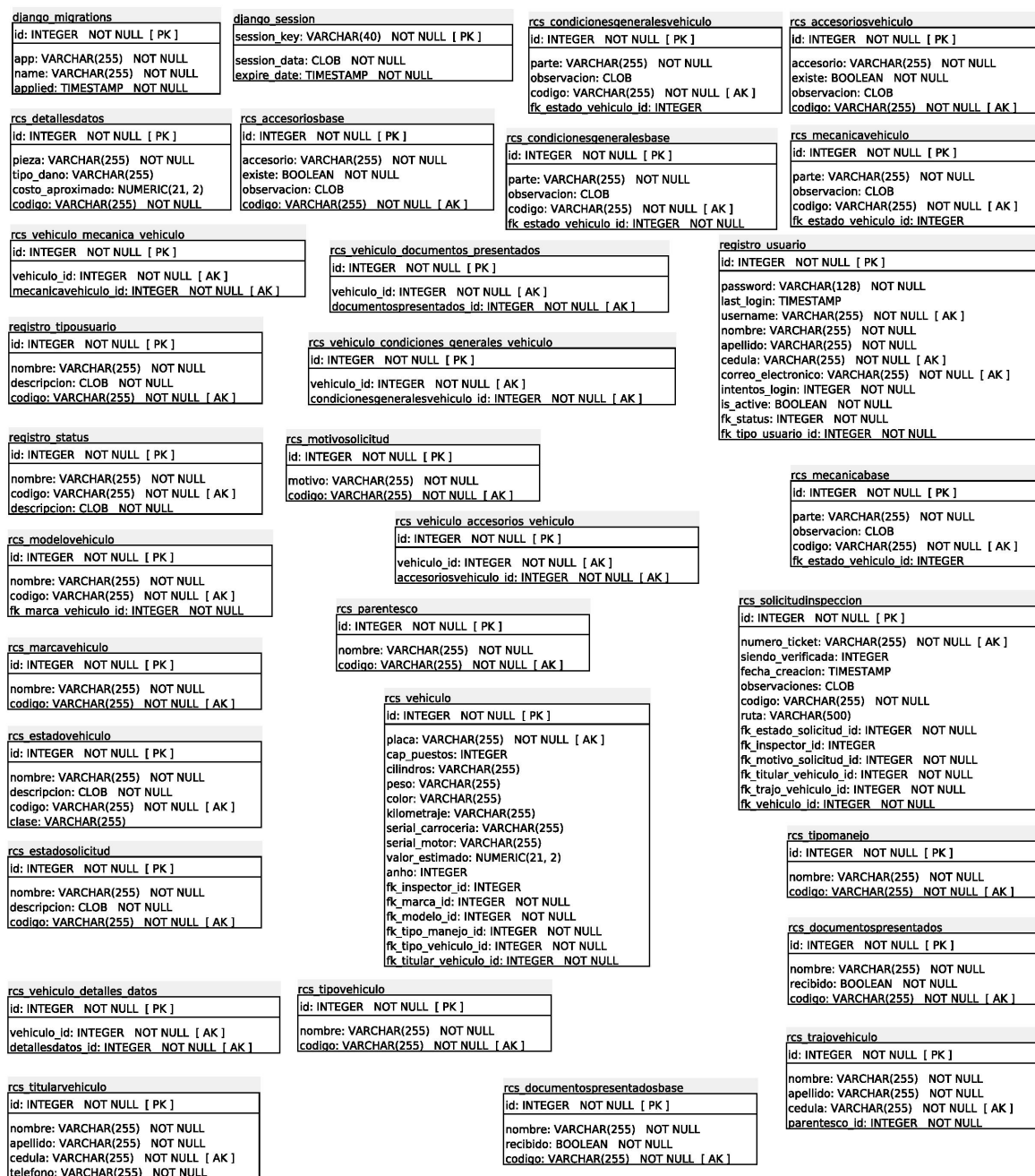


Figura 4.4: Modelo de datos de la aplicación. Fuente:El Autor.

4.7. Descripción de los módulos del sistema y sus interfaces

El sistema web se encuentra dividido en varios módulos. Los cuales están representados en la siguiente Figura.

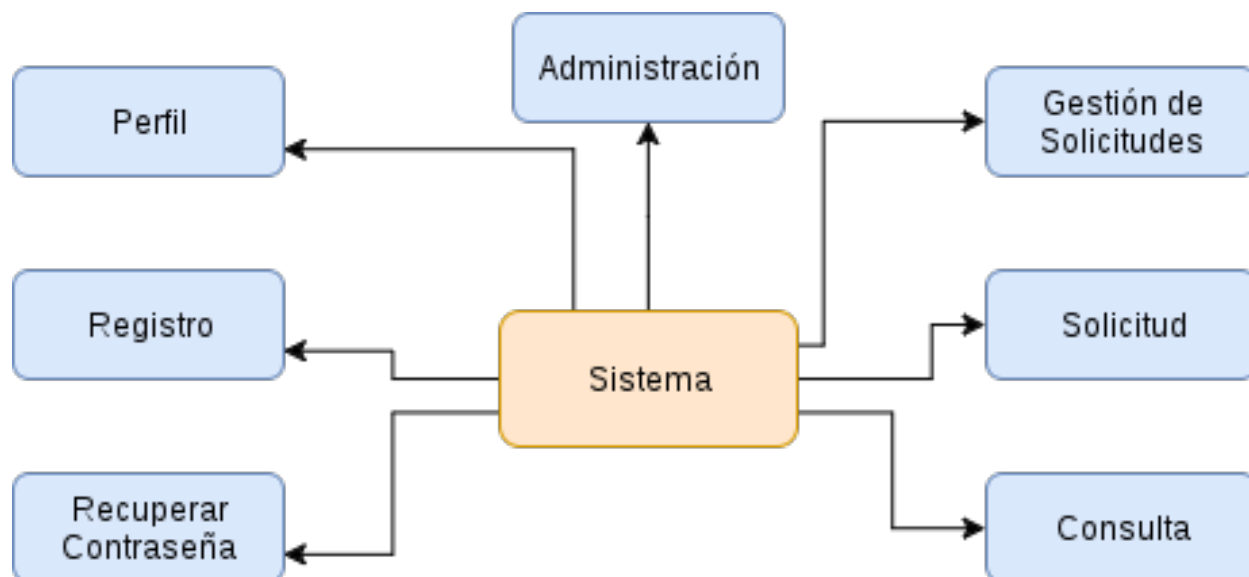


Figura 4.5: Diagrama de módulos del sistema. Fuente:El Autor.

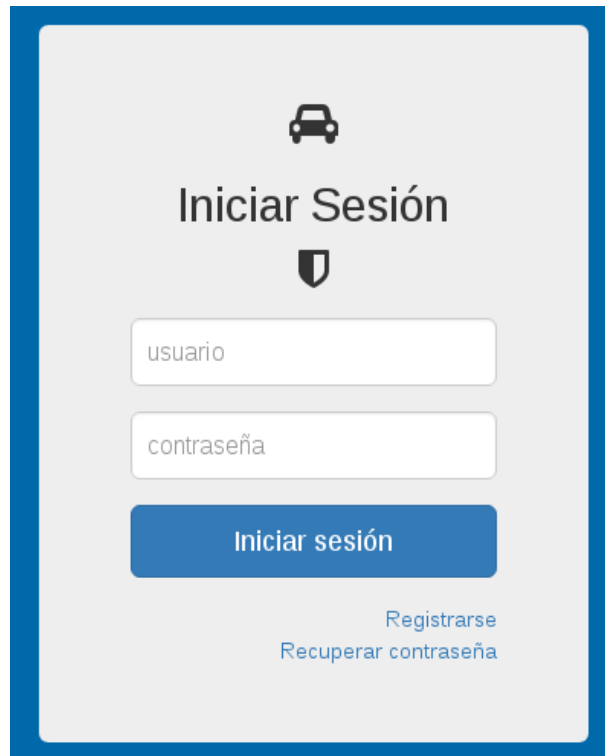
Módulos:

- **Administración:** En este modulo el usuario con perfil administrador puede configurar parámetros referentes a los usuarios del sistema.
- **Registro:** En este modulo los usuarios pueden registrarse en este sistema ya sea como usuario tipo Inspectores o Taquilla.
- **Perfil:** Este modulo le permite al usuario modificar su información de contacto registrada en el sistema.
- **Recuperar Contraseña:** En este modulo los usuarios pueden recuperar su contraseña en caso de olvidarla.
- **Consulta:** Este modulo hace énfasis en los filtros necesarios para la búsqueda de información dentro del sistema ya sea de solicitudes, usuarios o clientes (Aquellos que trajeron el carro a inspección)
- **Solicitud:** En este modulo los tipo de usuario Taquilla, al recibir un cliente, crean una solicitud de inspección.

- **Gestión de Solicitudes:** En este modulo los tipo de usuario Inspector realizan el registro de los datos de la inspección del vehículo y la gestión de las solicitudes de inspección.

4.7.1. Interfaces de la aplicación

-Interfaz de inicio de sesión: La figura a continuación muestra el formulario de inicio de sesión para aquellos usuarios registrados en el sistema.



The image shows a login form titled "Iniciar Sesión" (Login). It features a car icon at the top, followed by the title and a shield icon. Below these are two input fields labeled "usuario" (username) and "contraseña" (password). A blue button labeled "Iniciar sesión" (Login) is positioned below the input fields. At the bottom of the form, there are two links: "Registrarse" (Register) and "Recuperar contraseña" (Reset password).

Figura 4.6: Interfaz de Inicio de Sesión. Fuente:Print Screen.

A partir de esta interfaz los usuarios registrados en el sistema podrán loguearse con su usuario y contraseña, en caso de no estar registrados se presenta la opción para registrarse, además se muestra la opción de recuperar contraseña en caso de que el usuario la haya olvidado

-Interfaz de recuperar contraseña: La figura a continuación muestra el formulario de recuperación de contraseña para aquellos usuarios registrados en el sistema.



Figura 4.7: Interfaz de Recuperar Contraseña. Fuente: Print Screen.

A partir de esta interfaz los usuarios registrados en el sistema podrán recuperar su contraseña mediante el formulario incluyendo su correo y cédula. Se generara una contraseña aleatoria que sera enviada a su correo con la cual podrán acceder al sistema, para luego cambiar su contraseña a una propia.

-Interfaz registro de usuarios: La figura a continuación muestra el formulario de registro de usuarios para aquellos usuarios no registrados en el sistema.

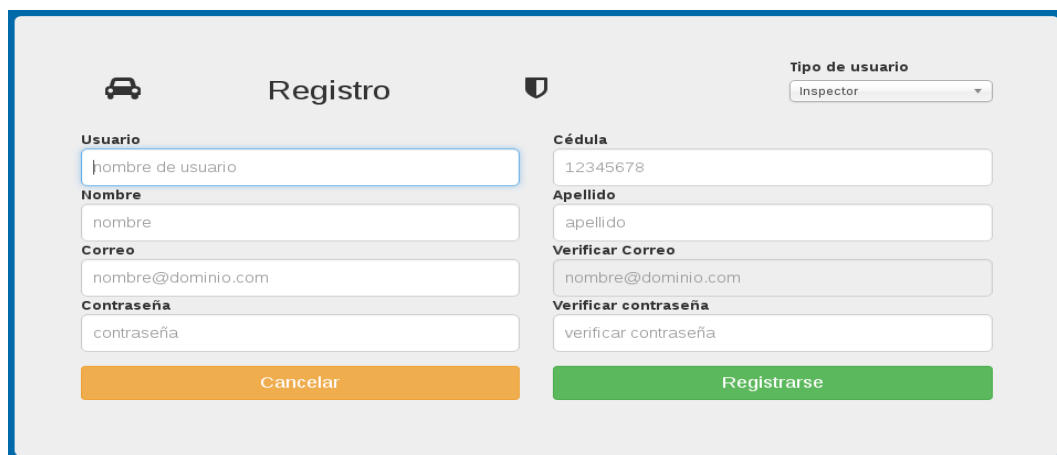


Figura 4.8: Interfaz de Registro de Usuarios. Fuente: Print Screen.

En esta interfaz los usuarios pueden registrarse bien sea como Inspectores o como usuarios Taquilla, ingresando su información personal.

-Interfaz Configuración de Perfil: La figura a continuación muestra el formulario de registro de usuarios para aquellos usuarios no registrados en el sistema.

The image shows a web form titled "Interfaz Configuración de Perfil". It is divided into several sections:

- Información de la Cuenta:** Contains two input fields: "Tipo de usuario" (with the value "Inspector") and "Usuario" (with the value "mcinsp").
- Datos Personales:** Contains three input fields: "Nombre" (with the value "MERCEDES CAROLINA"), "Apellido" (with the value "ARISMENDI"), and "Cédula" (with the value "17293827").
- Cambio de correo:** Contains two input fields: "Nuevo correo" (with the value "usuario.ins.tesis@gmail.com") and "Verificar nuevo correo" (with the value "usuario.ins.tesis@gmail.com").
- Cambio de contraseña:** Contains two input fields: "Nueva contraseña" (with the value "contraseña") and "Verificar nueva contraseña" (with the value "verificar contraseña").

At the bottom of the form, there are two buttons: "Cancelar" (orange) and "Guardar cambios" (green).

Figura 4.9: Interfaz de Configuración del Perfil de Usuario. Fuente: Print Screen.

En esta interfaz los usuarios pueden modificar su información personal ya sea el correo o contraseña.

-Interfaz Administrador bandeja de Usuarios: La figura a continuación muestra el filtro y bandeja de los usuarios registrados en el sistema.

Usuarios

Cedula

Correo

Tipo de Usuario

Nombre

Apellido

USUARIO	CEDULA	NOMBRE	APELLIDO	TIPO USUARIO	CORREO	OPCIONES
mcinsp	17293827	MERCEDES CAROLINA	ARISMENDI	INSPECTOR	usuario.ins.tesis@gmail.com	✎ ⏻ 🗑
egtaqu	1887263	EMILIO GERARDO	RAMIREZ RIVAS	TAQUILLA	tesis.shsa@gmail.com	✎ ⏻ 🗑

Figura 4.10: Interfaz de bandeja edición usuario por Administrador. Fuente: Print Screen.

En esta interfaz los usuarios pueden ser modificados, inactivados y eliminados. Inactivados siempre y cuando no tengan una solicitud de inspección en proceso y eliminado siempre y cuando no tengan ninguna solicitud asociada al mismo.

-Modal Inactivar Usuario: La figura a continuación muestra el modal de confirmación para inactivar un usuario.

CONFIRMACIÓN

Desactivar Usuario

¿Esta seguro que desea desactivar este usuario?

Figura 4.11: Modal Inactivar Usuario. Fuente: Print Screen.

-Modal Eliminar Usuario : La figura a continuación muestra el modal de confirmación para eliminar un usuario.

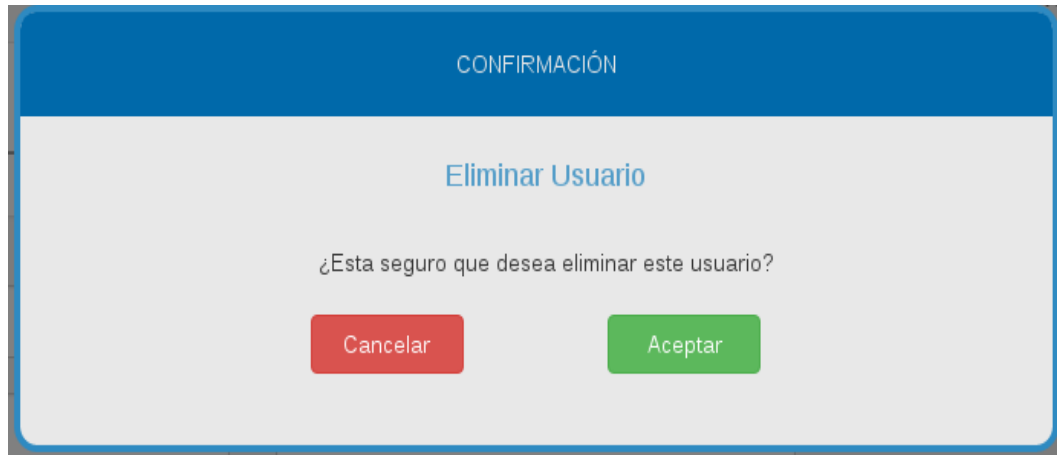


Figura 4.12: Modal Eliminar Usuario. Fuente: Print Screen.

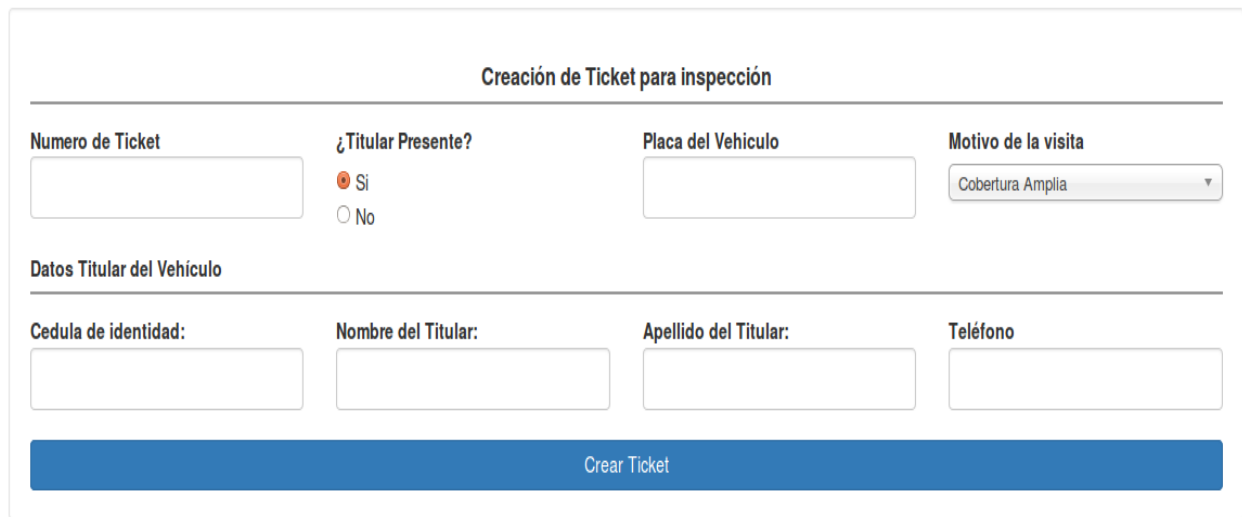
-Interfaz Administrador Edición de Usuarios: La figura a continuación muestra el formulario de edición de usuario por parte del administrador.

A screenshot of a web application form titled 'Datos del Usuario'. The form is organized into two columns. The left column contains three input fields: 'Usuario' with the value 'egtaqu', 'Nombre' with the value 'EMILIO GERARDO', and 'Correo' with the value 'tesis.shsa@gmail.com'. The right column contains three input fields: 'Cédula' with the value '1887263', 'Apellido' with the value 'RAMIREZ RIVAS', and 'Verificar Correo' with the value 'tesis.shsa@gmail.com'. Above the 'Cédula' field, there is a 'Reiniciar Usuario' checkbox and a 'Tipo de usuario' dropdown menu currently set to 'Taquilla'. At the bottom of the form, there are two buttons: a red 'Atrás' button on the left and a blue 'Guardar cambios' button on the right.

Figura 4.13: Interfaz de edición de usuario por Administrador. Fuente: Print Screen.

En esta interfaz el administrador puede cambiar tanto el nombre como el correo, el nombre no puede ser cambiado si el inspector ya proceso alguna solicitud. Además esta la función de reiniciar el usuario, el cual funciona como el recuperar contraseña en caso de que el usuario haya olvidado su correo o usuario registrado.

-Interfaz creación de solicitud de inspección: La figura a continuación muestra el formulario para la creación de una solicitud de inspección.



Creación de Ticket para inspección

Numero de Ticket	¿Titular Presente?	Placa del Vehículo	Motivo de la visita
	☒ Si ☐ No		Cobertura Amplia

Datos Titular del Vehículo

Cedula de identidad:	Nombre del Titular:	Apellido del Titular:	Teléfono

Crear Ticket

Figura 4.14: Interfaz de creación de solicitud de inspección. Fuente: Print Screen.

En esta interfaz el usuario Taquilla puede crear una solicitud de inspección con los datos del cliente dando este el numero de solicitud de inspección, en caso de que el cliente exista, se llenan automáticamente al incluir la cédula. En caso de que no este el Titular se incluyen también los datos de la persona que trajo el vehículo a inspección.

-Modal crear solicitud de inspección : La figura a continuación muestra el modal de confirmación para crear una solicitud de inspección.



Figura 4.15: Modal crear solicitud de inspección. Fuente: Print Screen.

-Interfaz Taquilla bandeja de solicitudes: La figura a continuación muestra el filtro y bandeja de las solicitudes de inspección creadas recientemente.

Tickets Abiertos

Cedula
Placa
Numero Solicitud

NRO. SOLICITUD	NRO. PLACA	CEDULA	NOMBRE	ESTADO	OPCIONES
169	COAIHSOI	1982701	Maria	ABIERTA	✖
128	FAM99P	8177301	Armando	ABIERTA	✖

Figura 4.16: Interfaz de bandeja solicitudes de inspección por Taquilla. Fuente: Print Screen.

En esta interfaz los usuarios taquilla pueden cancelar una solicitud de inspección siempre y cuando esta no este siendo atendida por algún inspector.

-Modal eliminar solicitud de inspección: La figura a continuación muestra el modal de confirmación para eliminar una solicitud de inspección.

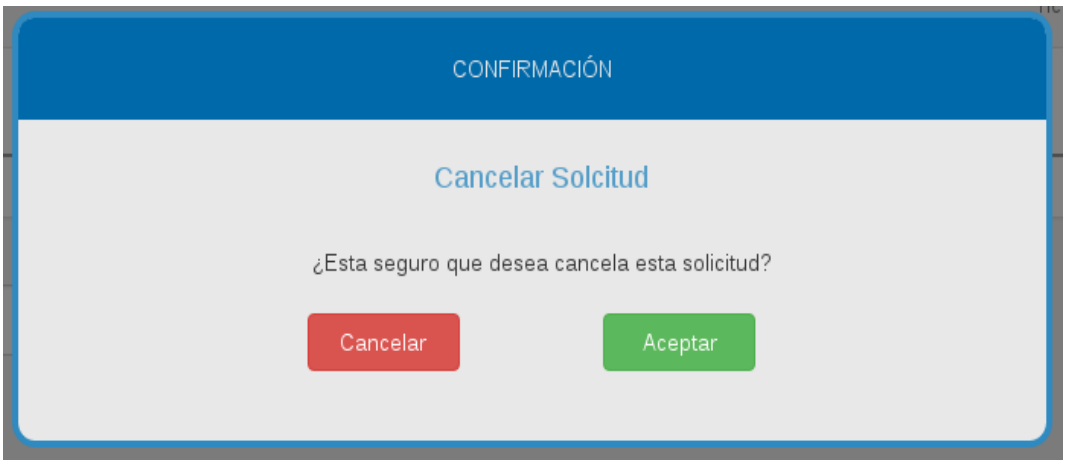


Figura 4.17: Modal eliminar solicitud de inspección. Fuente: Print Screen.

-Interfaz Inspectores bandeja de solicitudes: La figura a continuación muestra el filtro y bandeja de las solicitudes de inspección creadas.

Solicitudes de Inspección

Numero Solicitud

Cedula

Estado Solicitud

Placa

Buscar

NRO. SOLICITUD	NRO. PLACA	CEDULA	NOMBRE	ESTADO	OPCIONES
127	09AJSJ18	1092772	Pedro	POR GESTIONAR	
128	FAM99P	8177301	Armando	ABIERTA	

Figura 4.18: Interfaz de bandeja solicitudes de inspección usuario Inspector. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los datos de la inspección realizada, gestionar la solicitud de inspección y visualizar el documento generado por la solicitud.

-Interfaz registro datos del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los datos del vehículo.

The image shows a web form titled "Datos del Vehículo". It is divided into two main sections: "Datos Titular del Vehículo" and "Vehículo".

Datos Titular del Vehículo

- Nombre y Apellido del Titular:** Input field with "Pedro Villanueva".
- Cedula de identidad:** Input field with "19009172".
- Teléfono:** Input field with "(0424)-962-01-98".

Vehículo

- Tipo Modelo:** Dropdown menu with "Sedan".
- Tipo:** Dropdown menu with "Automatico".
- Placa:** Input field with "10923098".
- Marca:** Dropdown menu with "Abarth".
- Modelo:** Dropdown menu with "124 SPIDER".
- Año:** Input field.
- Canitdad de Puestos:** Input field.
- Cilindros:** Input field.
- Peso:** Input field.
- Color:** Input field.
- Kilometraje:** Input field.
- Serial de carroceria:** Input field.
- Serial del motor:** Input field.
- Valor estimado:** Input field.

At the bottom, there are two buttons: "Atrás" (red) and "Continuar" (blue).

Figura 4.19: Interfaz del formulario de registro de los datos del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los datos del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz condiciones generales del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los datos de condiciones generales del vehículo.

Condiciones Generales del Vehículo								
PART E	ESTADO	OBSERVACIONES	PART E	ESTADO	OBSERVACIONES	PART E	ESTADO	OBSERVACIONES
23. Latonería	☐ B ☐ R ☐ M		24. Planilla O Rejilla Frontal	☐ B ☐ R ☐ M		25. Techo	☐ B ☐ R ☐ M	
26. Pintura	☐ B ☐ R ☐ M		27. Sop O Focos Traseros	☐ B ☐ R ☐ M		28. Compuesta Trasera	☐ B ☐ R ☐ M	
29. Tapicería	☐ B ☐ R ☐ M		30. Cocuyos Frontales	☐ B ☐ R ☐ M		31. Parabrisas	☐ B ☐ R ☐ M	
32. Cauchos	☐ B ☐ R ☐ M		33. Retrovisor Interno	☐ B ☐ R ☐ M		34. Parachoques	☐ B ☐ R ☐ M	
35. Platinas	☐ B ☐ R ☐ M		36. Manillas Y Cerraduras	☐ B ☐ R ☐ M		37. Cometa	☐ B ☐ R ☐ M	
38. Tableros	☐ B ☐ R ☐ M		39. Limpia Parabrisas	☐ B ☐ R ☐ M		40. Butacas Y Asientos	☐ B ☐ R ☐ M	
41. Faros Y Sus Aros	☐ B ☐ R ☐ M		42. Retrovisores Externos	☐ B ☐ R ☐ M		43. Volantes	☐ B ☐ R ☐ M	
44. Emblema De Modelo	☐ B ☐ R ☐ M		45. Guardafangos	☐ B ☐ R ☐ M		46. Cinturones De Seguridad	☐ B ☐ R ☐ M	
47. Estribos	☐ B ☐ R ☐ M		48. Puertas	☐ B ☐ R ☐ M		49. Ventanas	☐ B ☐ R ☐ M	
50. Frontal	☐ B ☐ R ☐ M		51. Capot	☐ B ☐ R ☐ M		52. Maleta Y Tapa Maleta	☐ B ☐ R ☐ M	
53. Modific. Del Modelo	☐ B ☐ R ☐ M		54. Carter Trasero	☐ B ☐ R ☐ M		55. Otro	☐ B ☐ R ☐ M	

Atrás
Continuar

Figura 4.20: Interfaz del formulario de registro de condiciones generales del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los datos de las condiciones generales del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión

de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz mecánica del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los datos de la mecánica vehículo.

Mecánica del Vehículo								
PARTE	ESTADO	OBSERVACIONES	PARTE	ESTADO	OBSERVACIONES	PARTE	ESTADO	OBSERVACIONES
56. Encendido	☐ B ☐ R ☐ M		57. Embrague O Crocht	☐ B ☐ R ☐ M		58. Sistema Electrico	☐ B ☐ R ☐ M	
59. Otros	☐ B ☐ R ☐ M		60. Otros	☐ B ☐ R ☐ M		61. Otros	☐ B ☐ R ☐ M	

Atrás

Continuar

Figura 4.21: Interfaz del formulario de registro de la mecánica del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los datos de la mecánica vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz accesorios del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los datos de los accesorios vehículo.

Accesorios del Vehículo								
ACCESORIOS	POSEE	OBSERVACIONES	ACCESORIOS	POSEE	OBSERVACIONES	ACCESORIOS	POSEE	OBSERVACIONES
62. Radio / Antena	☐ Si ☐ No	Especificar si es radio o antena	63. Foco De Posicion	☐ Si ☐ No		64. Rines Especiales	☐ Si ☐ No	
65. Radio Reproductor	☐ Si ☐ No		66. Aire Acondicionado	☐ Si ☐ No		67. Caucho Respuesto	☐ Si ☐ No	
68. Reproductor de Antena	☐ Si ☐ No		69. Spoiler	☐ Si ☐ No		70. Parachoque Espec. (Mataburro)	☐ Si ☐ No	
71. Cornetas Especiales	☐ Si ☐ No		72. Llave De Cruz	☐ Si ☐ No		73. Rayas Decativas	☐ Si ☐ No	
74. Compact Disc	☐ Si ☐ No		75. Triangulo De Seguridad	☐ Si ☐ No		76. Tazas De Los Cauchos	☐ Si ☐ No	
77. Ecualizados	☐ Si ☐ No		78. Parrilla Techo	☐ Si ☐ No		79. Volante Especial	☐ Si ☐ No	
80. Televisor	☐ Si ☐ No		81. Cabina	☐ Si ☐ No		82. Gato	☐ Si ☐ No	
83. Planta	☐ Si ☐ No		84. Faros Especiales	☐ Si ☐ No		85. Anti-Robo	☐ Si ☐ No	
86. Tranca Pedal	☐ Si ☐ No		87. Themo King	☐ Si ☐ No		88. Otro	☐ Si ☐ No	
89. Cortacorreinte	☐ Si ☐ No		90. Tranca Palanca	☐ Si ☐ No		91. Otro	☐ Si ☐ No	

Atrás
Continuar

Figura 4.22: Interfaz del formulario de registro de los accesorios del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los datos de los accesorios del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz detalles y datos del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los detalles y datos del vehículo.

Detalles de Datos

[Agregar detalle +](#)

CODIGO	92. PIEZA	93. TIPO DE DAÑO	94. COSTO APROX.	
F99182	Condensador	Averiado	120000.00	
M9IAS20	Radiador	Problema sistema de refrigeración	230000.00	

[Atrás](#)
[Continuar](#)

Figura 4.23: Interfaz del formulario de registro de los detalles y datos del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los detalles y datos del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz documentos del vehículo en la solicitud de inspección: La figura a continuación muestra el formulario para el registro de los documentos traídos del vehículo.

Documentos Presentados para Inspección

DOCUMENTO	TRAJO	DOCUMENTO	TRAJO	DOCUMENTO	TRAJO
95. Registro Automotor Permanente (R.P.A)	☐ Si ☒ No	96. Carnet De Circulacion	☐ Si ☒ No	97. Documento De Propiedad (Original/Copia)	☐ Si ☒ No
98. Facutura De Compra	☐ Si ☒ No	99. Planilla Del Senial De Exoneracion De Impuestos	☐ Si ☒ No	100. Certificado De Origen	☐ Si ☒ No

[Atrás](#)
[Continuar](#)

Figura 4.24: Interfaz del formulario de registro de los documentos del vehículo. Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden registrar los documentos que trajeron del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Interfaz vista previa de la solicitud de inspección: La figura a continuación muestra el formulario para el registro de las observaciones referente a la inspección del vehículo y una vista previa de como quedara la solicitud.

LOGO

Bandeja Solicitudes

Vehículo 09AJSJ18

MCINSP

LOGO

INFORME DE INSPECCIÓN AUTOMÓVIL

Fecha de elaboración: 04/07/2017

Hora: 10:13:53

Motivo de la Inspección

Cobertura Amplia

Datos del Vehículo

1. Nombre y apellido del Titular: Pedro Perez

2. Cedula de identidad: 1092772

3. Teléfono: (0424)-881-92-81

4. Nombre y apellido de la persona que trajo el vehículo a la inspección: Juan Perez

5. Cedula de identidad: 23098190

6. Parentesco con el titular: Hijo

7. Marca: Peugeot

8. Modelo: 508

9. Cap. Puestos: 4

10. Cilindros: 6

11. Color: Gris

12. Tipo: Sedan

13. Año: 2013

14. Placa: 09AJSJ18

15. Peso: 1815kg

16. Automatico

17. Kilometraje: 150km

18. Serial del Motor: S17W394W02231131

19. Serial de Carrocería. Año: 2013

20. Valor Estimado: 09AJSJ18

21. Nombre del Inspector: MERCEDES CAROLINA ARISMENDI

22. C.I Nro. 17293827

Condiciones Generales del Vehículo

PARTES	B	R	M	OBSERVACIONES	PARTES	B	R	M	OBSERVACIONES	PARTES	B	R	M	OBSERVACIONES
23. Latoneria	X				24. Planilla O Rejilla Frontal	X				25. Techo			X	
26. Pintura	X				27. Sop O Focos Traseros			X		28. Compuesta Trasera	X			
29. Tapiceria	X				30. Cocuyos Frontales	X				31. Parabrisas	X			
32. Cauchos	X				33. Retrovisor Interno			X		34. Parachoques	X			
35. Platinas		X			36. Manillas Y Cerraduras	X				37. Cometa	X			
38. Tableros	X				39. Limpia Parabrisas	X				40. Butacas Y Asientos	X			
41. Faros Y Sus Aros	X				42. Retrovisores Externos	X				43. Volantes	X			
44. Emblema De Modelo	X				45. Guardafangos	X				46. Cinturones De Seguridad	X			
47. Estribos	X				48. Puertas	X				49. Ventanas		X		
50. Frontal	X				51. Capot			X		52. Maleta Y Tapa Maleta	X			
53. Modific. Del Modelo	X				54. Carter Trasero	X								

Mecánica del Vehículo

PARTES	B	R	M	OBSERVACIONES	PARTES	B	R	M	OBSERVACIONES	PARTES	B	R	M	OBSERVACIONES
56. Encendido	X				57. Embrague O Crocht	X				58. Sistema Electrico	X			
59. Otros	X			Otro 1										

Figura 4.25: Interfaz de la vista previa de los datos del vehículo (Parte 1). Fuente: Print Screen.

66

Accesorios del Vehículo											
ACCESORIOS	SI	NO	OBSERVACIONES	ACCESORIOS	SI	NO	OBSERVACIONES	ACCESORIOS	SI	NO	OBSERVACIONES
62. Radio / Antena	X		Radio	63. Foco De Posicion	X			64. Rines Especiales		X	
65. Radio Reproductor	X			66. Aire Acondicionado	X			67. Caucho Respuesto	X		
68. Reproductor de Antena	X			69. Spoiler		X		70. Parachoque Espec. (Mataburro)		X	
71. Cornetas Especiales	X			72. Llave De Cruz	X			73. Rayas Decativas		X	
74. Compact Disc		X		75. Triangulo De Seguridad	X			76. Tazas De Los Cauchos	X		
77. Ecualizados		X		78. Parrilla Techo		X		79. Volante Especial	X		
80. Televizor		X		81. Cabina	X			82. Gato	X		
83. Planta		X		84. Faros Especiales	X			85. Anti-Robo	X		
86. Tranca Pedal	X			87. Themo King	X			88. Otro	X		Porta Vasos
89. Cortacorreinte		X		90. Tranca Palanca	X						

Detalles de datos

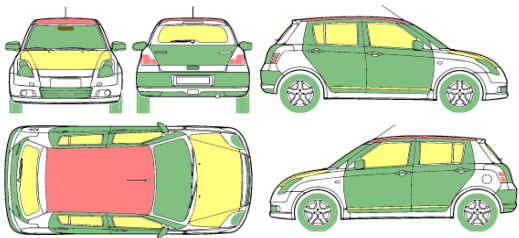
CODIGO	92. PIEZA	93. TIPO DE DAÑO	94. COSTO APROX.
F9129	Microchip	Quemado	350000.00
E398019	Tablero Electrico	Fusil averiado	40000.00
108279	Platina del techo	Abolladuras	150000.00

Documentos Presentados para Inspección

DOCUMENTO	SI	NO
95. Registro Automotor Permanente (R.P.A)	X	
96. Carnet De Circulacion	X	
97. Documento De Propiedad (Original/Copia)	X	
98. Facutura De Compra		X
99. Planilla Del Senial De Exoneracion De Impuestos	X	
100. Certificado De Origen	X	

Observaciones:

No hay observaciones a agregar.



NOTA: La inspeccion del vehículo descrito anteriormente, no implica aceptación del riesgo por par de Horizonte S.A. los riegos que asume la empresa correrán por cuenta de la misma, desde el momento en que el asegurado haya cancelado la prima convenida.

Yo, el inspector declaro que:
 No he omitido y/o simulado, ningún hecho o circunstancia en las respuesta, que puedan modificar las condiciones de riesgo y que los datos indicado fueron tomados del vehículo descrito en el presente formulario y en las fotografías respectivas.

Yo, el Asegurado y/o Solicitante declaro que:
 a) En caso de fraude o declaración falsa por mi parte, la presente inspección es NULA, siendo de igual manera NULO el Contrato que la misma respalda, censando en cuyo caso toda responsabilidad de la empresa.
 b) De igual manera certifico todos y cada uno de los SERIALES y NUMERACIONES Y CONDICIONES indicadas por el inspector, en el presente formulario, como los correspondientes a mi (el) vehículo.

Firma Inspector de riesgos	Firma Asegurado o Solicitante
Firma Intermediario de Seguros	Fecha: Codigo:

Atrás

Continuar

Figura 4.26: Interfaz de la vista previa de los datos del vehículo (Parte 2). Fuente: Print Screen.

En esta interfaz los usuarios inspector pueden verificar todo los datos del vehículo de la inspección realizada. También dicha interfaz sirve para la gestión de la solicitud, mostrando los datos anteriormente cargados para su ultima revisión.

-Modal finalizar registro de inspección: La figura a continuación muestra el modal de confirmación para registrar los datos de una solicitud de inspección.



Figura 4.27: Modal finalizar la solicitud de inspección. Fuente: Print Screen.

-Modal finalizar gestión de inspección: La figura a continuación muestra el modal de confirmación para finalizar la solicitud de inspección de manera definitiva.

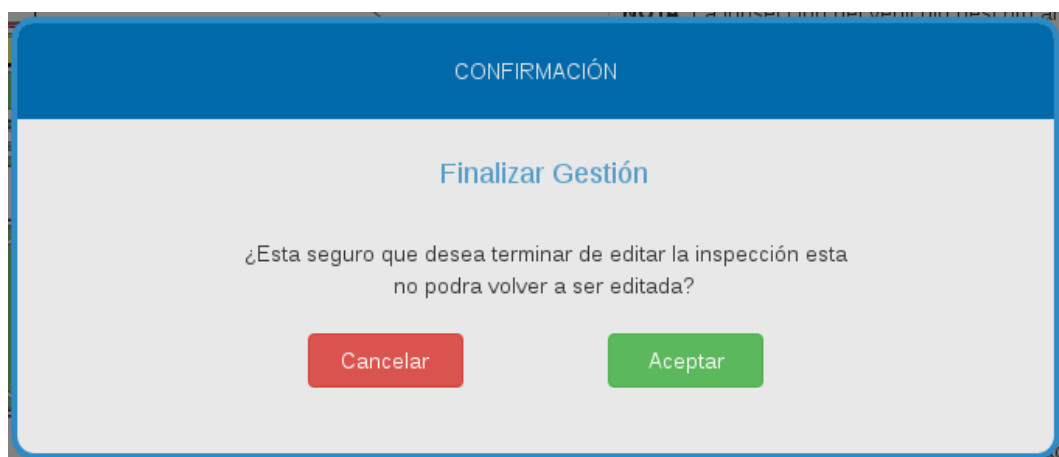


Figura 4.28: Modal finalizar la solicitud de inspección. Fuente: Print Screen.

4.8. Fase de pruebas

Las pruebas realizadas para verificar el correcto funcionamiento del sistema y la aceptación de los usuarios por el mismo se dividieron en dos (2) partes a continuación se explica detalladamente las mismas.

4.8.1. Pruebas funcionales

Para estas pruebas se realizó una tabla mostrada a continuación, en la cual se especifican los módulos pertenecientes al sistema y las funcionalidades que debería tener cada uno de ellos, estas fueron probadas por usuarios especialmente seleccionados que poseyeran un criterio de evaluación acertijo y crítico en cuanto a los resultados.

Modulo	Funcionalidades Probadas	Resultados
Administración.	■ Modificación de Usuarios. ■ Modificación de Clientes. .	.
Registro.	■ Registrar cuenta .	.
Perfil.	■ Editar información del usuario .	.
Recuperar Contra- seña.	■ Recuperar Contraseña .	.

Tabla 4.3: Instrumento evaluación módulos del sistema (Parte 1). Fuente: El autor

Modulo	Funcionalidades Probadas	Resultados
Consulta.	■ Consultar solicitudes creadas. ■ Consultar Usuarios de la aplicación. ■ Consultar titulares de carros. ■ Consultar personas que trajeron el vehículo a la inspección.	.
Solicitud.	■ Realizar solicitud de inspección.	.
Gestión de Solicitudes.	■ Registrar los datos de la solicitud de inspección ■ Gestionar la solicitud de inspección	.

Tabla 4.4: Instrumento evaluación módulos del sistema (Parte 2). Fuente: El autor

Las pruebas presentadas en esta tabla se fueron realizando mediante ciclos hasta lograr la completa aprobación de los módulos.

Usuario.	Ciclo de Prueba		
	1.	2.	3.
1.	■ 3 módulos aprobados ■ 4 módulos no aprobados	■ 5 módulos aprobados ■ 2 módulos no aprobados	■ 7 módulos aprobados.
2.	■ 3 módulos aprobados ■ 4 módulos no aprobados	■ 5 módulos aprobados ■ 2 módulos no aprobados	■ 7 módulos aprobados.
3.	■ 3 módulos aprobados ■ 4 módulos no aprobados	■ 5 módulos aprobados ■ 2 módulos no aprobados	■ 7 módulos aprobados.

Tabla 4.5: Matriz de ciclos de pruebas. Fuente: El autor

-Ciclo 1:

- Usuario 1: de los 7 módulos probados, 4 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - El correo no se enviaba correctamente a los usuarios registrados.
 - Campos numéricos aceptaban caracteres.
 - Datos de las solicitudes de inspección siempre se precargaban los mismos, y cuando se cambia en algún cliente se cambian en todos.
 - Inactivar usuarios.

- Usuario 2: de los 7 módulos probados, 4 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - El correo no se enviaba correctamente a los usuarios registrados.
 - Campos numéricos aceptaban caracteres.
 - Inactivar usuarios.
 - Datos inconsistentes en diferentes solicitudes de inspección.
- Usuario 3: de los 7 módulos probados, 4 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - No llega el correo al registrar el usuario
 - Error en montos al colocar carácter.
 - Datos de las solicitudes de inspección siempre se precargaban los mismos, y cuando se cambia en algún cliente se cambian en todos.
 - Inactivar usuarios.

-Ciclo 2:

- Usuario 1: de los 7 módulos probados, 2 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - Consultar titulares del vehículo no funcionaba consultar por numero.
 - Al terminar la solicitud de inspección, no se muestra en el pdf las observaciones realizadas
- Usuario 2: de los 7 módulos probados, 2 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - Consultar titulares del vehículo no funcionaba consultar por numero.
 - Al terminar la solicitud de inspección, no se muestra en el pdf las observaciones realizadas
- Usuario 3: de los 7 módulos probados, 2 terminaron con el estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:
 - Consultar titulares del vehículo no funcionaba consultar por numero.
 - Al terminar la solicitud de inspección, no se muestra en el pdf las observaciones realizadas

-Ciclo 3: En este ciclo todos los usuarios aprobaron todas las funcionalidades del sistema de todos los módulos.

4.8.2. Pruebas de aceptación

Para comprobar el nivel de aceptación del sistema, se realizó una encuesta correspondiente a las interfaces del sistema orientadas a la usabilidad. Dicha encuesta fue realizada a un grupo de diez personas. De esta manera se pudo obtener un feed back de parte de los usuarios para mejorar las interfaces del sistema.

		1	2	3	4	5
1	La aplicación es intuitiva					
2	Considera usted que los colores crean un buen contraste.					
3	El diseño y estilo de la aplicación es consistente a lo largo de sus interfaces					
4	El lenguaje usado en la aplicación es claro					
5	La aplicación ofrece mensajes acordes a las operaciones realizadas.					
6	Es posible navegar por la aplicación sin perderse					
7	La aplicación es usable					
8	Las acciones realizadas son fáciles de memorizar					
9	La aplicación muestra mensajes de errores explícitos de acuerdo al error ocurrido					

Tabla 4.6: Encuesta de aceptación. Fuente: El Autor

Para el desarrollo de esta encuesta se deben seleccionar valores que van desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo", estableciendo un rango de número del 1 al 5.

Los resultados de las encuestas aplicadas, para evaluar el nivel de aceptación, se pueden apreciar en las siguientes Figuras.

-Resultado de las encuestas:

La aplicación es intuitiva

10 respuestas

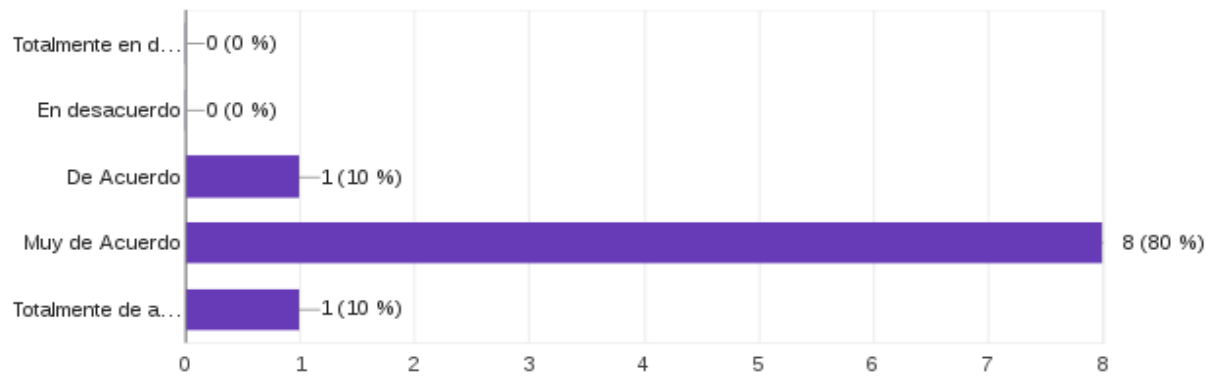


Figura 4.29: Resultados de la pregunta 1. Fuente: El Autor.

Considera usted que los colores crean un buen contraste

10 respuestas

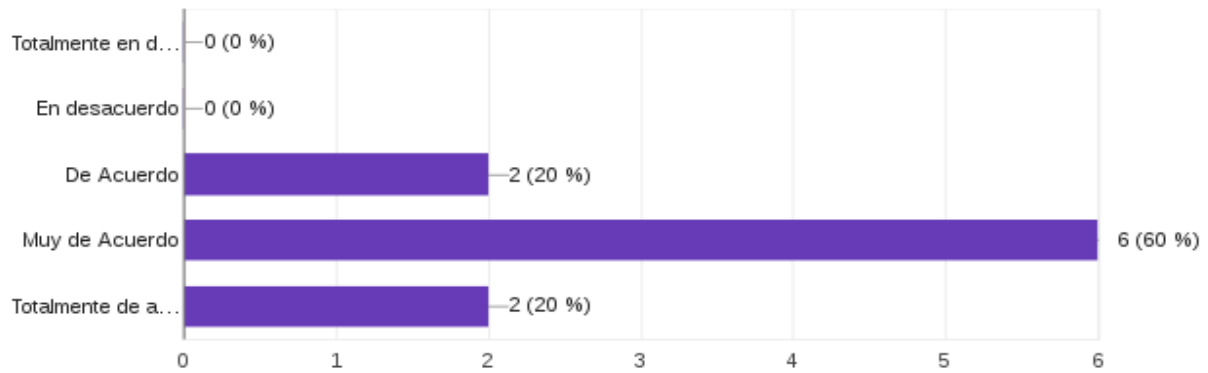


Figura 4.30: Resultados de la pregunta 2. Fuente: El Autor.

El diseño y estilo de la aplicación es consistente a lo largo de sus interfaces.

10 respuestas

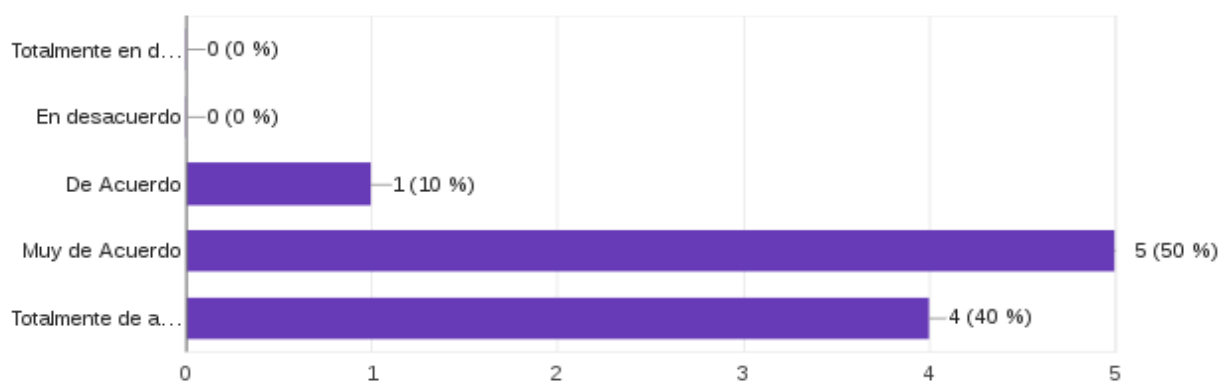


Figura 4.31: Resultados de la pregunta 3. Fuente: El Autor.

El lenguaje usado en la aplicación es claro

10 respuestas

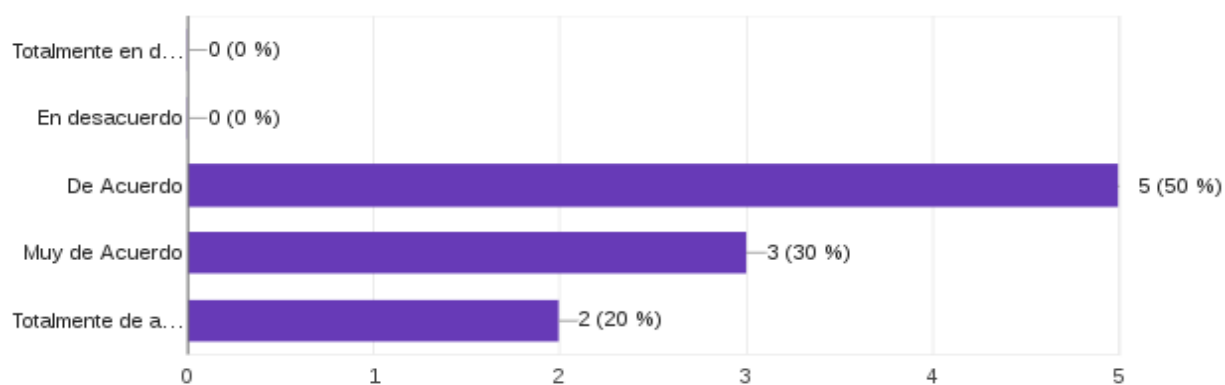


Figura 4.32: Resultados de la pregunta 4. Fuente: El Autor.

La aplicación ofrece mensajes acordes a las operaciones realizadas.

10 respuestas

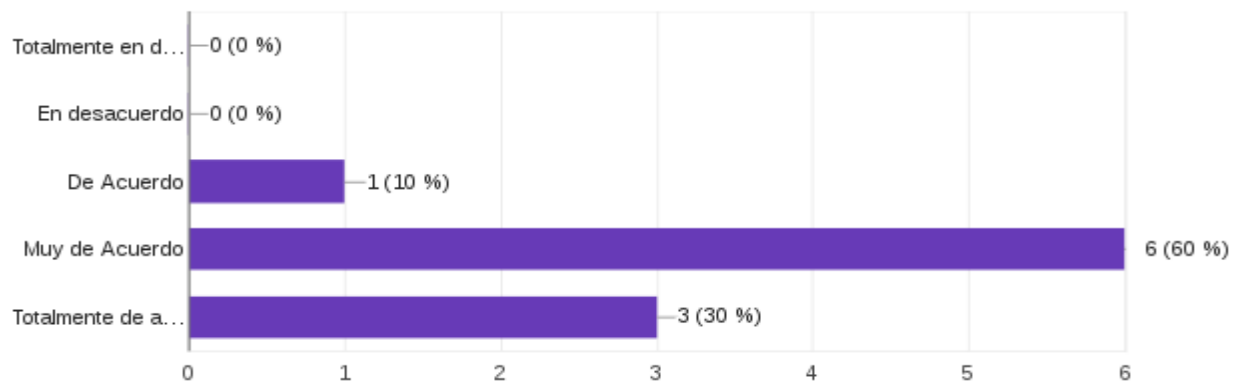


Figura 4.33: Resultados de la pregunta 5. Fuente: El Autor.

Es posible navegar por la aplicación sin perderse

10 respuestas

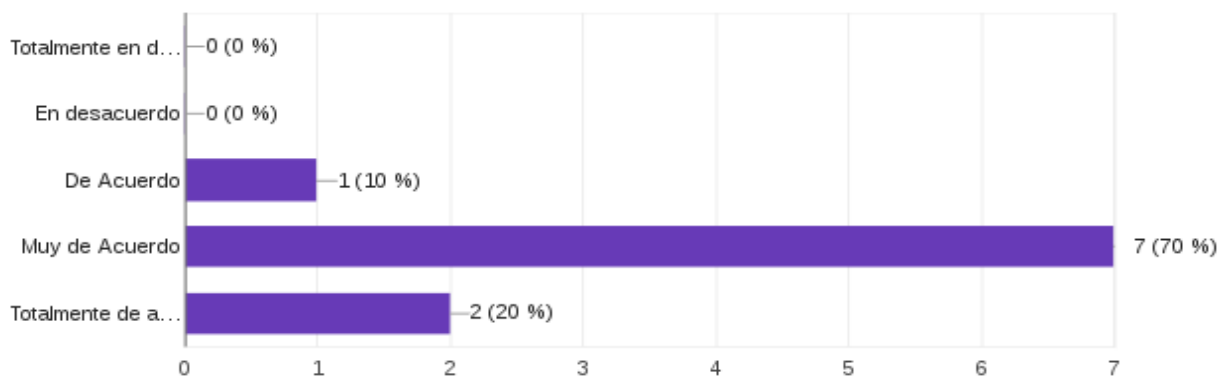


Figura 4.34: Resultados de la pregunta 6. Fuente: El Autor.

La aplicación es usable

10 respuestas

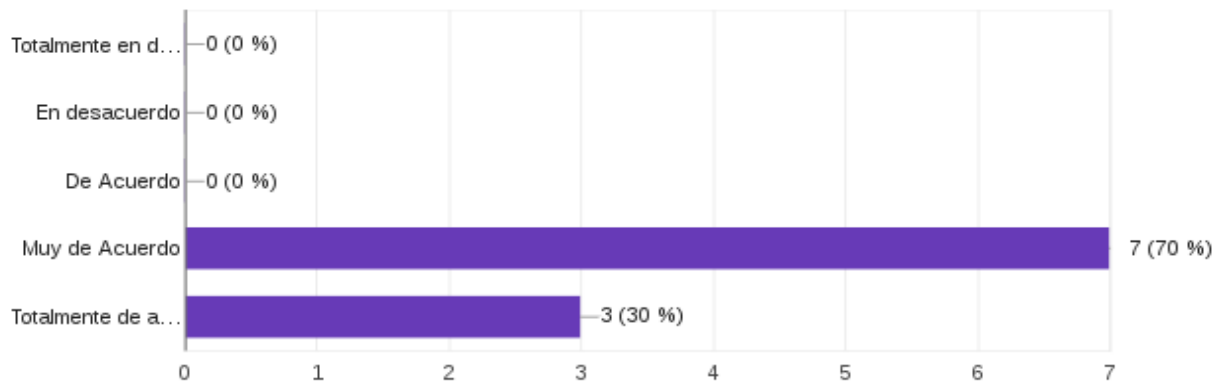


Figura 4.35: Resultados de la pregunta 7. Fuente: El Autor.

Las acciones realizadas son fáciles de memorizar

10 respuestas

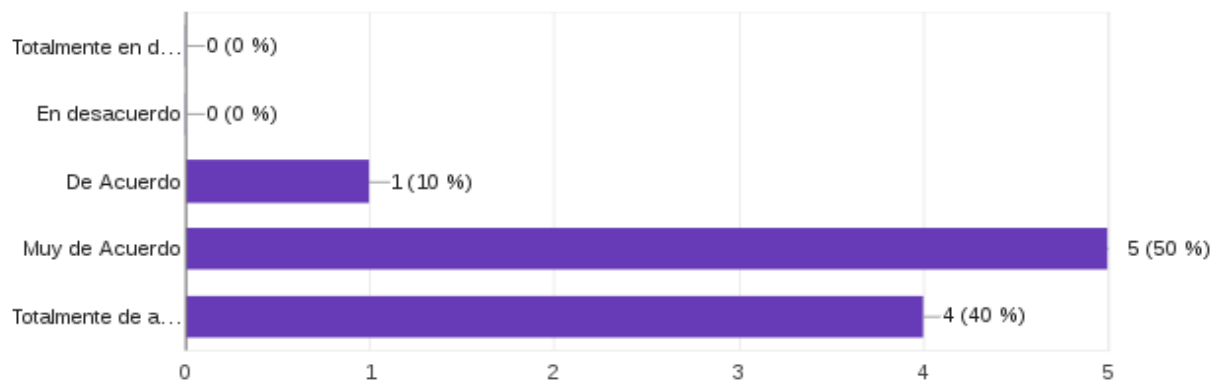


Figura 4.36: Resultados de la pregunta 8. Fuente: El Autor.

La aplicación muestra mensajes de errores explícitos de acuerdo al error ocurrido

10 respuestas

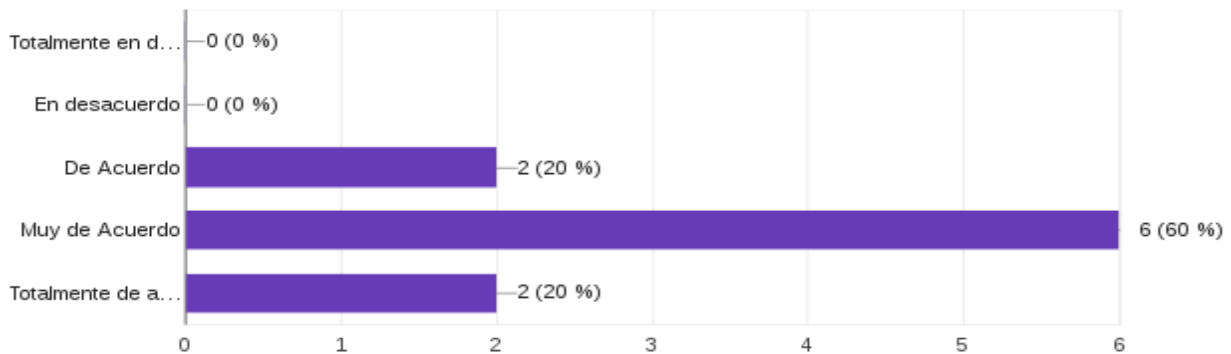


Figura 4.37: Resultados de la pregunta 9. Fuente: El Autor.

Con base en los resultados anteriores se puede concluir que la aplicación es bastante intuitiva, utiliza colores que crean buen contraste y haciendo la aplicación consistente entre sus interfaces, logrando así gran usabilidad. Además se puede apreciar que el lenguaje utilizado en la aplicación tanto como los mensajes de respuesta de la misma son de agrado para los usuarios aunque en parte del lenguaje de la aplicación hay ciertos tecnicismos los cuales le dan un nivel de complejidad a los mismos. Aun así por ser intuitiva, es fácil recordar las acciones realizadas.

CONCLUSIÓN Y RECOMENDACIONES

Se ha podido desarrollar de manera satisfactoria un sistema de información que cumple con los requerimientos planteados para su realización.

En cuanto al análisis de las especificaciones del cliente, se logró un correcto levantamiento de información logrando así, definir un alcance específico de cada uno de los procesos.

A través del alcance definido se pudo generar el modelos de datos, diseño de las interfaces y módulos los cuales fueron certificados por el usuario.

Logrando así desarrollar un sistema capaz de gestionar las planillas de inspección de una compañía de seguros. En este sistema los usuarios podrán gestionar las solicitudes de inspección en conjunto con los datos de los vehículos inspeccionados. Manteniendo así una persistencia digital de la información para un mejor rendimiento y desempeño en el manejo de ella, cumpliendo así con los objetivos previamente definidos.

En cuanto a las pruebas se logró la completa validación de las funcionalidades creadas, realizando pruebas de funcionalidad y de usabilidad cuyos resultados sirvieron de retroalimentación positiva, en pro del desarrollo de un producto de calidad 100 % funcional. Teniendo un producto de software comprobado y con sus funcionalidades

La metodología de desarrollo ágil SCRUM utilizada para el desarrollo de este proyecto, contribuyo de manera significativa en le realización del mismo. Ya que posee una muy buena organización de roles los cuales desempeñan un papel dentro del proyecto para lograr una sinergia entre los actores de SCRUM. Por lo cual fue un gran aporte al desarrollo de la aplicación.

Entre las recomendaciones se sugiere incorporar módulo móvil para la captura de los datos de la inspección, de modo que esta información pueda ser enlazada con el sistema web a través de un web service. También incorporar un módulo de Inteligencia de Negocio de manera que se puedan analizar los datos existentes y crear estrategias enfocadas a la

administración y creación de conocimiento.

Y por ultimo incorporar un módulo de gestión documental de manera que se pueda administrar el flujo de documentos de todo tipo en la organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía.

BIBLIOGRAFÍA

- Acosta, A. E. (2011). Agilus: Construcción ágil de la usabilidad. *Disponible en* http://www.ciens.ucv.ve:8080/genasig/sites/interaccion-humano-comp/archivos/234_CLEI_Acosta_Paper.pdf.
- Albarán, I. (2008). La actividad aseguradora: importancia, revisión e integración de conceptos fundamentales. *Disponible en* <http://eprints.ucm.es/6723/1/0022.pdf>.
- Alvarez, M. A. (2012). Manual de jquery. *Espanha. Retrieved, 4*.
- Asegurándome. (2016). Seguro de automóvil. *Disponible en* <http://asegurandome.com.ve/seguro-de-automovil/>.
- Base de Datos Documento en línea. *Disponible en.* (2016). <http://es.ccm.net/contents/66-introduccion-bases-de-datos>. (Consultado el 17 de octubre de 2017)
- Carlos, H. (2012). El perito, clave en la compañía de seguros. *Disponible en* https://www.fundacionmapfre.org/documentacion/publico/i18n/catalogo_imagenes/grupo.cmd?path=1071550.
- Collison, S. (2007). *Beginning css web development: from novice to professional [inicio del desarrollo web css: desde principiante hasta profesional]*. Apress.
- Documento en línea. *Disponible en.* (2004). <http://www.methodsandtools.com/archive/archive.php?id=18>. (Consultado el 17 de octubre de 2017)
- Documento en línea. *Disponible en.* (2004). <http://xoratom.com/blog/2015/09/09/extreme-programming/>. (Consultado el 20 de agosto de 2016)
- Documento en línea. *Disponible en.* (2013). https://upload.wikimedia.org/wikipedia/commons/thumb/0/01/Modelo_Cascada_Secuencial.jpg/350px-Modelo_Cascada_Secuencial.jpg. (Consultado el 17 de octubre de 2017)
- Ecma, S. (2015). 404: The json data interchange format. *Disponible en* <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> *Cited on, 9*.
- Flanagan, D. (2011). *Javascript: The definitive guide: Activate your web pages*. O'Reilly Media, Inc.
- Golding, D. (2008). *Beginning cakephp: from novice to professional [comenzando cakephp: de principiante a profesional]*. Apress.
- Graham, I. S. (1995). *The html sourcebook [el libro de código html]*. John Wiley & Sons, Inc.
- Guaranty Car, S. (2012). Cómo peritar un coche. *Disponible en* <http://blog.carsycars.com/seguridad-y-tecnologia/como-peritar-un-coche/>.
- Hansson, D. H. (s.f.). Ruby on rails (2013). URL <http://rubyonrails.org>.

- Hinz, S., DuBois, P., y Stephens, J. (2005). *Mysql 5.7 reference manual*. <http://dev.mysql.com/doc/refman/5.7/en/introduction.html>. (Consultado el 17 de octubre de 2017)
- Holovaty, A., y Kaplan-Moss, J. (2009). *The definitive guide to django: Web development done right* [la guía definitiva de django: Desarrollo web hecho correctamente.]. Apress.
- Jacobson, I., Rumbaugh, G., Jacobson, J., Booch, G., y Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software/the unified software development process*. (n.º 004.41). Pearson Educación,.
- Karen, C., y Karen, D. D. C. (1996). *Sistemas de información para la toma de decisiones*.
- Laudon, F., y Laudon, J. (1996). *Sistemas de información*. Editorial Diana, México.
- McLeod, R. (2000). *Sistemas de información gerencial*. Pearson educación.
- Modelo RUP – IBM* Documento en línea. Disponible en. (2016). <https://dtyoc.com/2016/06/07/modelo-rup-ibm/>. (Consultado el 17 de octubre de 2017)
- Oracle SMDDB* Documento en línea. Disponible en. (2016). <https://iessanvicente.com/colaboraciones/oracle.pdf>. (Consultado el 17 de octubre de 2017)
- Otto, M., Thornton, J., y cols. (2015). Bootstrap: the world's most popular mobile-first and responsive front-end framework.'. *Getbootstrap. com*.
- PostgresSql* Documento en línea. Disponible en. (2016). http://www.postgresql.org.es/sobre_postgresql. (Consultado el 17 de octubre de 2017)
- Red, E. (2013). *Ecu red*. Obtenido de http://www.ecured.cu/Modelo_en_cascada.
- Robles, G., y Ferrer, J. (2002). Programación extrema y software libre. *Universidad Politécnica de Madrid. España*.
- Schwaber, K., y Beedle, M. (2002). *Agile software development with scrum* [desarrollo ágil de software con scrum] (Vol. 1). Prentice Hall Upper Saddle River.
- Sistema Manejador de Base de Datos Valor de la Gestion de los Datos* Documento en línea. Disponible en. (2015). <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/406549/Qu-es-el-sistema-manejador-de-bases-de-datos>. (Consultado el 17 de octubre de 2017)
- Wellman, D. (2011). *Jquery ui 1. 8: The user interface library for jquery* [la biblioteca de interfaz de usuario de jquery]. Packt Publishing Ltd.