



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE INVESTIGACIÓN DE SISTEMAS DE INFORMACIÓN

DESARROLLO DE UNA SOLUCIÓN CON TECNOLOGÍA MÓVIL PARA EL PROCESO DE INSPECCIÓN DE VEHÍCULOS EN COMPAÑÍAS DE SEGURO

*Trabajo de Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por el
Br. Reynaldo Trujillo
C.I. V-20978593
Para optar al título de
Licenciado en Computación
Tutor: Prof. Lic. Franky Uzcátegui*

Caracas, Octubre 2018

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 tecnología móvil para el proceso de inspección de vehículos en compañías de seguro" y presentado por el Br. Reynaldo Trujillo Vera C.I. 20.978.593, 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 16 de Octubre de 2018, a las 3:00 p.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 18 puntos.


En fe de lo cual se levanta la presente Acta. en Caracas a los 16 días del mes de Oct del año 2018.



Prof. Franky Uzcátegui
Tutor



Prof. Carlos Acosta
Jurado Principal



Prof. Franklin Sandoval
Jurado Principal

DEDICATORIA

A mis familiares, sobre todo a mis padres
por su apoyo incondicional y siempre creer en mí.

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

Reynaldo Trujillo Vera

AGRADECIMIENTOS

Principalmente agradezco a mi madre Doris Vera, y mi padre Franklin Trujillo por su gran apoyo incondicional a lo largo de todo este proceso, siempre motivándome a ser mejor persona y que todo se puede con perseverancia y dedicación.

A mi hermana Dorys por estar siempre ahí para mí, creer siempre en mí y ayudarme a desconectarme un poco de todo cuando me encontraba muy estresado.

A mi tío Alfonso por su constancia, por su ayuda en tantos momentos, por brindarme buenos consejos cuando los necesitaba. A mi abuela Josefina por estar siempre pendiente de mí y recordarme que siempre que necesite ayuda ella estará ahí.

A todos mis amigos, por tantos buenos momentos, tantos aprendizajes, tantas bromas y tanta ayuda brindada durante todos estos años. A mis hermanos “Los varonsitos” Yldemaro, Kevin, Misael, Félix V., Jorman, Félix I y Alberto, por todo su apoyo y estar conmigo en los peores momentos. A mis amigos de primera por tan buena amistad y trascendencia en tan poco tiempo.

A mis amigas Alison y Sarahy por ser las mejores y estar siempre para mí con su ayuda y juicio totalmente sincero. A mis amigos Oriel y Raffaele por apoyarme tanto durante la carrera y brindarme una muy buena amistad. Y a mi novia Bárbara por su apoyo emocional e insistencia en que culmine esta etapa.

A Virtual Castle por ser la empresa que me ha visto crecer profesionalmente, brindarme tantas oportunidades y darme el ímpetu de seguir aprendiendo y trabajando con diversas tecnologías.

A mi tutor Franky por toda su paciencia, por su consistencia aun cuando yo no me encontraba avanzando en el trabajo. Gracias por enseñarme a ser mejor profesional y siempre exigirme lo mejor de mí.

Al Centro de Investigación en Sistemas de Información (CISI) por permitirme ser parte de él durante tantos años de mi carrera, por darme la oportunidad de convertirme en preparador y descubrir lo mucho que me gusta enseñar.

A la Universidad Central de Venezuela, la casa que vence la sombra, por abrirme sus puertas y permitir mi formación académica y profesional, ser testigo de mi crecimiento personal y darme la oportunidad de formarme como Licenciado de Computación.

¡Gracias a todos!

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

DESARROLLO DE UNA SOLUCIÓN CON TECNOLOGÍA MÓVIL PARA EL PROCESO DE INSPECCIÓN DE VEHÍCULOS EN COMPAÑÍAS DE SEGURO

Autor: Reynaldo Esteban Trujillo Vera, C.I. V-20978593

Tutor: Prof. Franky Uzcátegui Polo

RESUMEN

El objetivo del presente Trabajo Especial de Grado se basa en la creación de un sistema haciendo uso de tecnologías web para realizar una aplicación móvil que apoye en el proceso de inspección de vehículos que deben llevar a cabo los peritos en las compañías de seguros para optar por una suscripción.

Este sistema funciona como complemento de un módulo web previamente desarrollado para llevar gestión de las solicitudes de inspección desde su creación hasta que son atendidas, el uso de una digitalización de estos procesos se traduce en reducción de costos y margen de error humano al automatizar tareas como el vaciado de los valores del registro en el sistema web.

Para el desarrollo del sistema, se aplicó una adaptación de la metodología de desarrollo ágil SCRUM. Se usaron herramientas web como Ionic y Angular para la creación de la aplicación móvil, SQLite para la base de datos móvil lo que permite realizar cambios en las inspecciones sin conexión a internet y un estilo de arquitectura MVC.

El proyecto se dividió en fases, teniendo así la fase de análisis donde se realizó el levantamiento y análisis de requerimientos, fase de desarrollo donde se implementó la solución, fase de pruebas funcionales para comprobar el comportamiento y funcionamiento y por último una fase de pruebas de aceptación para usuarios finales con el fin de tener retroalimentación sobre aspectos básicos de la aplicación.

PALABRAS CLAVE: Aplicación móvil, Ionic, Seguros, Inspección de vehículos, Compañía de seguros, Gestión solicitudes de inspección.

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1 : PROBLEMA DE INVESTIGACIÓN.....	2
1.1 Título	2
1.2 Planteamiento del problema	2
1.3 Justificación	2
1.4 Objetivo General.....	3
1.5 Objetivos Específicos.....	3
1.6 Solución propuesta	3
1.7 Alcance.....	6
CAPÍTULO 2 : MARCO CONCEPTUAL.....	7
2.1. Seguro.....	7
2.1.1. Contrato del seguro	7
2.1.2. Póliza de seguro	7
2.1.3. Seguro de vehículo.....	7
2.1.4. El perito	8
2.2 Sistemas de Información	13
2.2.1 Definición.....	13
2.2.2 Clasificación de los sistemas de información	13
2.2.3 Sistemas de Procesamiento de Transacciones en Línea (OLTP)	15
2.3 Base de Datos	16
2.3.1 Características de una Base de Datos	16
2.4. Sistema manejador de Base de datos	17
2.4.1. PostgreSQL	18
2.5 Base de datos móvil	19
2.5.1 Arquitectura de una base de datos móvil	19
2.5.2 Sistema Manejador de Base de Datos Móvil.....	20
2.6. Computación Móvil	21
2.7. Tecnologías web.....	28
2.7.1. HTML.....	28
2.7.2. CSS	28
2.7.3. JavaScript.....	29
2.8. Servicio web	29
2.8.1. Protocolo SOAP.....	29
2.8.2. Protocolo REST	30
CAPÍTULO 3 : MARCO METODOLÓGICO.....	32

3.1	Metodología tradicional.....	32
3.2	Metodología ágil	33
3.3	Comparación entre las metodologías de desarrollo	34
3.4	SCRUM	34
3.4.1.	Teoría de Scrum	35
3.4.2.	El equipo Scrum	35
3.4.3.	Eventos de Scrum	36
3.4.4.	Artefactos de Scrum	37
3.4.5.	Metodología Scrum.....	38
3.5	Comparación entre metodologías ágiles	39
CAPÍTULO 4 : MARCO APLICATIVO		40
4.1	Descripción general de la solución.....	40
4.2	Aplicación de la metodología SCRUM	41
4.2.1	Lista de Objetivos	41
4.2.2	Lista de tareas de la iteración	43
4.3	Requerimientos del sistema.....	44
4.3.1	Requerimientos funcionales:.....	44
4.3.2	Requerimientos no funcionales:.....	44
4.4	Perfiles de usuario	44
4.5	Descripción del flujo asociado a la solución	46
4.5.1	Proceso de gestión de solicitud de inspección.....	46
4.6	Análisis del modelo de datos	47
4.6.1	Listado de tablas de la aplicación móvil	47
4.6.2	Modelo de datos	48
4.7	Descripción de los módulos del sistema e interfaces	49
4.7.1	Descripción de los módulos del sistema	49
4.7.2	Descripción de las interfaces del sistema	50
4.8	Fase de pruebas.....	69
4.8.1	Pruebas funcionales	69
4.8.2	Pruebas de aceptación	73
CONCLUSIONES.....		80
LIMITACIONES		80
RECOMENDACIONES		81
REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES.....		82

ÍNDICE DE ILUSTRACIONES

Ilustración 1.1: Arquitectura conceptual de la solución.	4
Ilustración 1.2: Arquitectura de la solución tecnológica.	5
Ilustración 2.1: Flujograma del proceso actual de Peritaje Vehicular.	10
Ilustración 2.2: Planilla de inspección vehicular actual.	12
Ilustración 2.3: Tipos de Sistemas de Información.	14
Ilustración 2.4: Arquitectura de una base de datos móvil.	19
Ilustración 2.5: Teléfonos móviles.	23
Ilustración 2.6: Teléfonos inteligentes.	24
Ilustración 2.7: Tableta.	24
Ilustración 2.8: PDA.	25
Ilustración 3.1 : Metodología Scrum.	38
Ilustración 4.1: Arquitectura de la solución tecnológica.	40
Ilustración 4.2: Flujograma del proceso de gestión de solicitud de Peritaje Vehicular.	46
Ilustración 4.3: Modelo de datos para la base de datos móvil.	48
Ilustración 4.4: Módulos involucrados en el sistema.	49
Ilustración 4.5: Interfaz de inicio de sesión.	50
Ilustración 4.6: Interfaz de bandeja de solicitudes pendientes.	51
Ilustración 4.7: Interfaz de bandeja de solicitudes en gestión.	52
Ilustración 4.8: Interfaz de menú lateral.	53
Ilustración 4.9: Interfaz de perfil de usuario inspector.	54
Ilustración 4.10: Interfaz de información básica del desarrollador.	55
Ilustración 4.11: Interfaz de página de inicio de inspección.	56
Ilustración 4.12: Interfaz de página de datos del vehículo en inspección.	57
Ilustración 4.13: Interfaz de página de las condiciones generales del vehículo en inspección.	58
Ilustración 4.14: Interfaz de la página de las mecánicas del vehículo en inspección.	59
Ilustración 4.15: Interfaz de la página de datos de los accesorios del vehículo en inspección.	60
Ilustración 4.16: Interfaz de página de detalles adicionales del vehículo en inspección.	61
Ilustración 4.17: Interfaz de página de los documentos del titular del vehículo en inspección.	62
Ilustración 4.18: Interfaz de la pantalla de archivos multimedia de la inspección.	63
Ilustración 4.19: Interfaz de alerta de confirmación de envío de solicitud a la base de datos local.	64
Ilustración 4.20: Interfaz de la alerta de confirmación de asignación de solicitud a inspector.	65
Ilustración 4.21: Interfaz de la alerta de confirmación de envío de solicitud al servidor.	66
Ilustración 4.22: Interfaz de la alerta de confirmación para salir de la aplicación.	67
Ilustración 4.23: Interfaz de alerta de confirmación para salir del formulario durante inspección.	68
Ilustración 4.24: Resultados de pregunta 1 del instrumento de evaluación.	74

Ilustración 4.25: Resultados de pregunta 2 del instrumento de evaluación.	74
Ilustración 4.26: Resultados de pregunta 3 del instrumento de evaluación.	75
Ilustración 4.27: Resultados de pregunta 4 del instrumento de evaluación.	75
Ilustración 4.28: Resultados de pregunta 5 del instrumento de evaluación.	76
Ilustración 4.29: Resultados de pregunta 6 del instrumento de evaluación.	76
Ilustración 4.30: Resultados de pregunta 7 del instrumento de evaluación.	77
Ilustración 4.31: Resultados de pregunta 8 del instrumento de evaluación.	77
Ilustración 4.32: Resultados de pregunta 9 del instrumento de evaluación.	78
Ilustración 4.33: Resultados de pregunta 10 del instrumento de evaluación.	78

ÍNDICE DE TABLAS

Tabla 2.1: Cuadro comparativo de los Sistemas Manejadores de Base de Datos Móvil.....	21
Tabla 2.2: Cuadro comparativo entre los aportes y las limitaciones de la computación móvil.	22
Tabla 2.3: Cuadro comparativo entre los protocolos SOAP y REST.	31
Tabla 3.1: Metodologías tradicionales y Metodologías ágiles.....	34
Tabla 3.2: Metodologías ágiles SCRUM, XP y Kanban.	39
Tabla 4.1: Lista de objetivos y tareas realizadas (1/2).....	41
Tabla 4.2: Lista de objetivos y tareas realizadas (2/2).....	42
Tabla 4.3: Pruebas funcionales por módulo.	69
Tabla 4.4: Matriz de ciclos de prueba.....	70
Tabla 4.5: Instrumento de evaluación para pruebas de aceptación.....	73

INTRODUCCIÓN

Las compañías aseguradoras que ofrecen servicios de pólizas de seguro a los clientes para la cobertura de riesgos en vehículos, deben realizar un proceso de peritaje mediante un intermediario que toma el rol de perito, el cual se encarga de inspeccionar el vehículo basado en unos estándares para determinar el estado del mismo con respecto a las condiciones y bases establecidas en la póliza contratada, de esta manera, el perito realiza los registros fotográficos y de formularios físicos para confirmar la inspección.

El caso de estudio de este Trabajo Especial de Grado es el proceso de inspección de vehículos llevado a cabo por una compañía aseguradora. La gestión actual de este proceso presenta varios inconvenientes debido a la dependencia de formularios físicos y archivos, ya que diariamente, los inspectores manejan un gran volumen de planillas que están propensas a traspapelarse o extraviarse, sin mencionar el fuerte impacto administrativo, tanto de personal y materiales como de tiempo, estos problemas mencionados buscan ser solventados por la compañía. Para ello, se propone aplicar el uso de la tecnología móvil con la intención de mejorar y agilizar el desempeño de este proceso.

Este documento se estructura en cuatro capítulos, el primer capítulo considerado como el Planteamiento del Problema, se presenta el flujo del proceso actual y se expone el caso de estudio, además se plantean los objetivos y alcances del estudio.

El segundo capítulo, conocido como el Marco Conceptual consta de los conceptos y definiciones de los temas y tecnologías consideradas para el desarrollo de este estudio.

En el tercer capítulo considerado como Marco Metodológico se definen las metodologías ágiles y tradicionales del desarrollo de un producto, exponiendo las ventajas y desventajas de cada una y mediante un cuadro comparativo se determina el uso de una de ellas como propuesta de este estudio.

Por último, el cuarto capítulo, se presenta el Marco Aplicativo, en el cual se explican y describen las fases que se aplicaron para la elaboración del sistema, según una adaptación de la metodología de desarrollo ágil SCRUM.

CAPÍTULO 1 : PROBLEMA DE INVESTIGACIÓN

1.1 Título

Desarrollo de una solución con tecnología móvil para el proceso de inspección de vehículos en compañías de seguro.

1.2 Planteamiento del problema

Actualmente, en el ámbito de aseguradas de Venezuela se sufre de una alta demanda de clientes para la prestación de un servicio eficiente y eficaz, es decir, que se tiene una gran necesidad de adaptarse a las nuevas tecnologías y desarrollar un proceso, optimizado de realizar las inspecciones de vehículos.

El proceso actual de inspección se encuentra realizándose de manera que los documentos generados son físicos y por lo cual su persistencia e integridad se ve afectada por el ambiente en donde se guardan luego de la inspección. Debido a este factor se buscan soluciones y/o alternativas que permitan cumplir el objetivo de una manera automatizada y optimizar el proceso con aplicaciones tecnológicas novedosas que estén al alcance de las personas.

Hoy en día al realizar una inspección de vehículo, el perito utiliza una cámara o celular para el registro fotográfico y una planilla física para registrar los datos de la inspección, luego en la oficina el perito debe almacenar en una carpeta digital en la computadora las fotos tomadas al vehículo e imprimir las más importantes para registrar un expediente físico para la compañía aseguradora.

1.3 Justificación

Digitalizar un proceso requiere un estudio previo del mismo para observar las fallas y las posibles automatizaciones, así como los sub procesos que se pueden eliminar llevando todo a una solución tecnológica. En este caso de estudio en particular, se determinó que una aplicación móvil puede simplificar el proceso en gran medida, adicionalmente, existe una disminución en el margen de error humano al tener un formulario digital con validaciones de entrada previas y sin papeles físicos que se pueden extraviar o confundir.

El uso de un framework de desarrollo multiplataforma como Ionic Cordova, permite aumentar la compatibilidad de la aplicación con los distintos sistemas operativos móviles que existen, para no limitar a la organización a poseer uno en específico, disminuir los tiempos de desarrollo y de mantenimiento posterior ya que al usar tecnologías web tan comunes la curva de aprendizaje es mucho menor que con un enfoque cien por ciento nativo.

En resumen, el uso de una aplicación móvil y dispositivos móviles le genera una reducción de costo a largo plazo a una organización y aporta beneficios al aumentar la productividad por eliminar las tareas rutinarias que se pueden automatizar.

1.4 Objetivo General

Implementar una solución con tecnología móvil para el proceso de Inspección de vehículos en compañías de seguro.

1.5 Objetivos Específicos

- Analizar requerimientos del proceso de Peritaje de Vehículos.
- Diseñar cada uno de los componentes y módulos de la arquitectura de la solución.
- Desarrollar cada uno de los componentes y módulos de la arquitectura de la solución usando Ionic como herramienta tecnológica.
- Aplicar un conjunto de pruebas funcionales, no funcionales y de aceptación a cada uno de los componentes y módulos de la arquitectura de la solución.

1.6 Solución propuesta

Se propone desarrollar una aplicación móvil híbrida a través de la cual se podrá agilizar el proceso de inspección de vehículos llevado a cabo por el perito de la compañía aseguradora. Mediante la aplicación móvil, el inspector podrá autenticarse, obtener una lista de solicitudes de inspección para seleccionar a la cual será atendida y permitir al perito registrar con el uso de una interfaz gráfica todos los datos relacionados con la inspección para su posterior envío desde la aplicación a la base de datos central. Las solicitudes son generadas por la aplicación web previamente desarrollada como el módulo web del proyecto compuesto por una aplicación móvil y una aplicación web que se conectan a una base de datos central, y su finalidad es de automatizar el proceso de inspección antes descrito.

Además, la aplicación propuesta hará uso de las ventajas de los dispositivos móviles, permitiendo a los peritos realizar inspecciones sin la necesidad de llevar formularios o planillas físicamente y realizar el envío de la información en cualquier momento que sea requerido, en caso de no tener conexión a internet, la aplicación móvil guardará una copia de los resultados en la base de datos móvil del dispositivo para su posterior envío en cuanto se cumplan las condiciones necesarias de conexión.

A continuación, la arquitectura conceptual asociada al flujo del proceso propuesto se compone de una aplicación web que posee la administración de usuarios e inspecciones, permitiendo la

solicitud y gestión de las inspecciones de la compañía. El intercambio de información se realiza a través de los servicios web que permiten el envío y recepción de datos asociados a los usuarios o inspecciones registradas, estos servicios permiten la conexión entre los distintos componentes de la arquitectura de la solución propuesta.

Los datos se almacenan en la base de datos central de la compañía que mediante una estructura relacional permite guardar en detalle la información asociada a los usuarios y las inspecciones, permitiendo gestionar las mismas de manera compatible con las funciones establecidas en la página web.

La aplicación móvil contiene un modelo digitalizado de los formularios físicos que utiliza la compañía para registrar la información de las inspecciones, permitiendo la portabilidad de varios formularios mediante un dispositivo móvil y la selección interactiva y sencilla de opciones para el registro de información asociada al vehículo. La información se almacena temporalmente en la base de datos móvil del dispositivo móvil para evitar pérdida de información, así como el almacenamiento en el caso de no tener conexión a internet y así poder ser enviados en otro instante de tiempo (ver Ilustración 1.1).

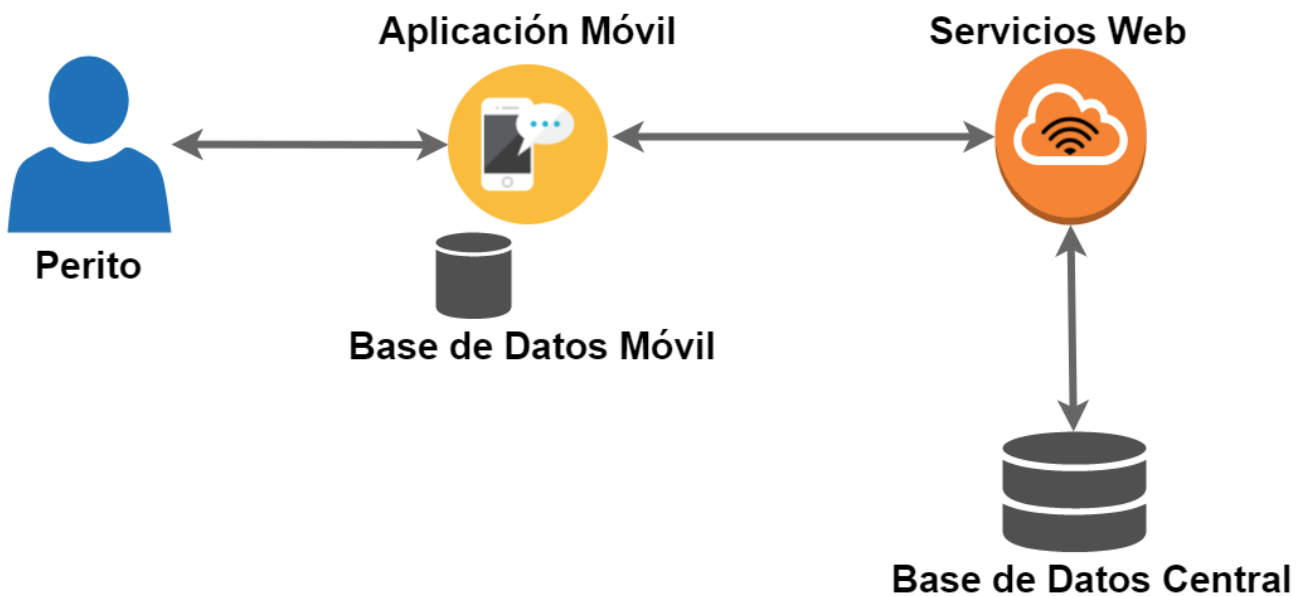


Ilustración 1.1: Arquitectura conceptual de la solución.

La Ilustración 1.2, presenta la arquitectura asociada a la propuesta en el cual se establece el desarrollo de la aplicación móvil, en esta arquitectura se definen las tecnologías a ser utilizadas para el desarrollo de cada uno de los componentes de la solución propuesta.

La aplicación móvil será desarrollada sobre el framework Ionic, el cual permite la portabilidad a diferentes sistemas operativos móviles y al estar hecho con Angular como base, se cuenta con

una comunidad en línea bastante dedicada, lo que facilita el desarrollo al momento de solventar errores que se puedan presentar, la base de datos móvil estará desarrollada en SQLite por su ligereza de almacenamiento, la cual es necesaria para guardar mucha información temporal asociada a una inspección. Los servicios web serán de tipo REST, que permite la compatibilidad con la tecnología que se encontrará en la aplicación móvil además de ser sencillo de desarrollar y mantener a futuro. Como formato de intercambio de información a través de los servicios web se propone el uso de JSON por su simpleza y gran desempeño para el desarrollo de servicios web, más aún, tomando en cuenta la compatibilidad absoluta que tiene con Angular. Finalmente, la base de datos central ya se encuentra previamente desarrollada para el almacenamiento de la información proveniente de la aplicación web, por lo que se trabajará con el sistema manejador de bases de datos PostgreSQL para añadir las estructuras que permitan el almacenamiento de la información generada desde el dispositivo móvil. Como acotación, el desarrollo de la aplicación web que gestiona las solicitudes no se contempla en esta propuesta debido a que ya se encuentra desarrollada previamente.

Para guiar el desarrollo de este proyecto se propone el uso de la metodología de desarrollo ágil SCRUM, por su facilidad de adaptación diseñada para ofrecer un valor significativo de forma rápida en todo proyecto. La misma permite presentar soluciones parciales mediante el uso de iteraciones, lo que hará que existan pruebas que contribuyan con el progreso del desarrollo de la aplicación al poder obtener retroalimentación de parte del cliente.

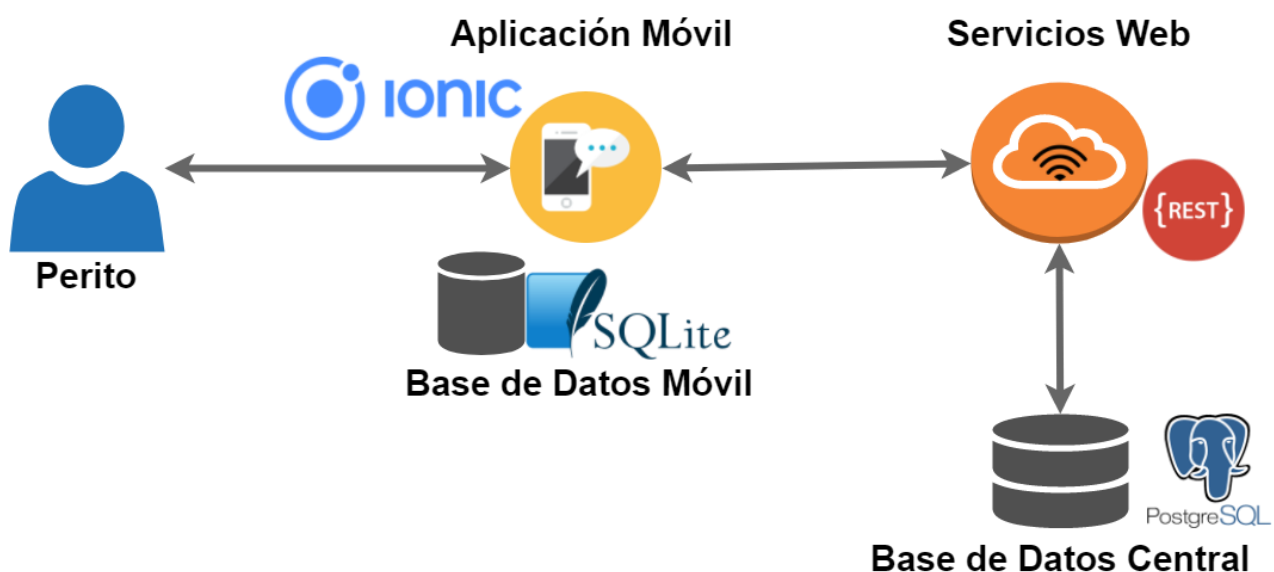


Ilustración 1.2: Arquitectura de la solución tecnológica.

1.7 Alcance

- Empezar el desarrollo de la solución a partir de los módulos y componentes previamente desarrollados, correspondientes a la base de datos central y la aplicación web.
- Recibir las solicitudes de inspección asignadas a cada perito en la aplicación móvil desde la base de datos central.
- Considerar la disponibilidad de conexión a internet o ausencia de la misma para el envío de datos desde la aplicación móvil.
- Contemplar los permisos y roles para la autenticación de usuarios en la aplicación móvil establecidos en la aplicación web.
- Registrar todos los datos que se solicitan en la inspección sincronizando con la base de datos central.
- Implementar la aplicación móvil orientada en el sistema operativo móvil Android 7.1, conocido como Nougat.

CAPÍTULO 2 : MARCO CONCEPTUAL

2.1. Seguro

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

2.1.1. 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 éste 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 pagado cuando ocurra el suceso o riesgo cubierto por el seguro.

2.1.2. Póliza de seguro

Según (Albarán, 2008) 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.

2.1.3. Seguro de vehículo

- **Cobertura amplia:** La aseguradora se compromete a indemnizar el asegurado, en exceso del deducible y hasta la suma asegurada indicada en el cuadro de 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.
- **Cobertura de indemnización por robo del vehículo:** La aseguradora se compromete a indemnizar al asegurado, en caso de queda 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.
- **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.

- **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.
- **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 de 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 encuentre dentro, subiendo o bajando del vehículo asegurado y bajo las condiciones estipuladas en la póliza.

2.1.4. El perito

Es un ente esencial en el engranaje de la compañía de seguros, pero para conocer la verdadera dimensión del trabajo del perito. Esta persona realiza la valoración de los daños mediante la elaboración del peritaje y realizando la propuesta de indemnización a la compañía de seguros, determinando el valor del bien asegurado, realiza los informes de uso interno para la compañía de seguros con la justificación técnica de la ocurrencia del siniestro, revisa los riesgos para la contratación de nuevas pólizas de vehículos de segunda mano con coberturas de daños propios, comprueba la reparación que se ha llevado conforme a la peritación en todas y cada una de las partidas asignadas por el perito.

2.1.4.1. Inspección para asegurar un vehículo

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 para el informe generado a las aseguradoras y/o al cliente y segundo, para evitar futuras reclamaciones al momento de determinar el proceso de inspección.

También se puede 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 está 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

2.1.4.2. Caso de estudio

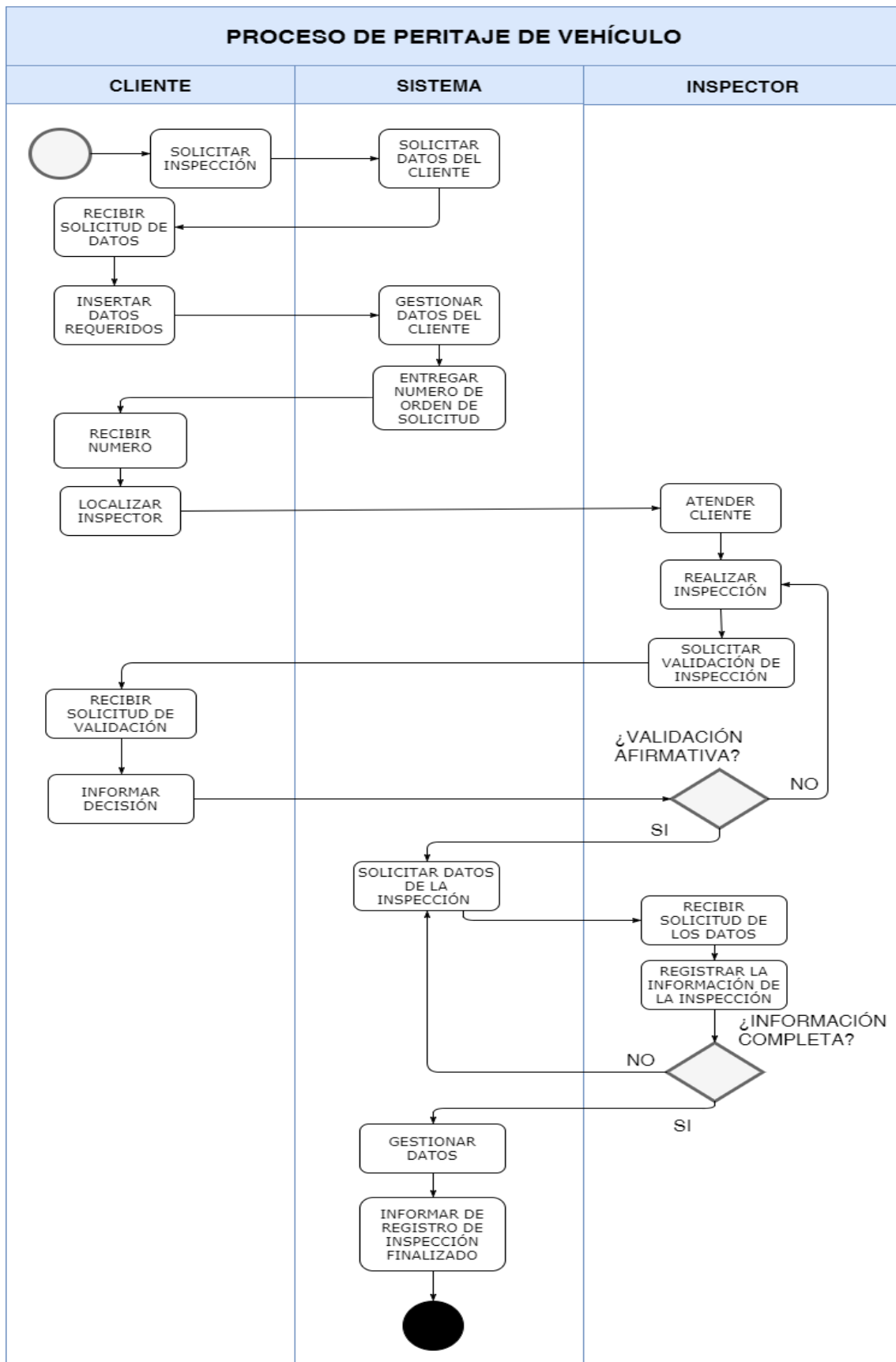


Ilustración 2.1: Flujograma del proceso actual de Peritaje Vehicular.

En la ilustración 2.1, se presenta un flujograma del proceso actual en el cual intervienen el cliente, el sistema o página web y el inspector o perito, este proceso inicia con el cliente que solicita la inspección, el sistema que gestiona las inspecciones y el perito que se encargará de inspeccionar el vehículo. El proceso actual de peritaje vehicular inicia con la solicitud de inspección por parte del cliente a través de la página web, en este momento el sistema solicita los datos del cliente y para registrar la solicitud, finalmente se genera un número de solicitud y se le notifica al inspector para realizar dicha inspección. El inspector realiza el proceso de inspección y registra manualmente en formularios físicos la información asociada a la inspección, para culminar el proceso se le solicita al cliente que valide la información anotada, si es negativa se procede a realizar la inspección nuevamente, en caso de ser afirmativa el inspector debe registrar los datos en la página web para su almacenamiento, esto incluye los registros fotográficos tomados mediante una cámara o dispositivo celular, cuando la información se encuentra completa se finaliza el registro de la inspección y finaliza el proceso.

En la ilustración 2.2, se puede observar la planilla de inspección que se usa actualmente, todos los datos deben ser llenados por el inspector físicamente para luego realizar un vaciado de toda la información en el módulo web previamente desarrollado.

2.2 Sistemas de Información

2.2.1 Definición

Para comprender que es un sistema de información es necesario definir previamente que un sistema es un conjunto de partes coordinadas y en interacción para alcanzar un conjunto de objetivos (Johansen Bertoglio, 1993). La clave está en las relaciones entre las diversas partes del mismo; puede existir un conjunto de objetos, pero si estos no se encuentran relacionados entre sí, no constituyen un sistema.

Luego de haber definido el término anterior, se pueden definir los sistemas de información como un conjunto de componentes organizados, que interactúan entre sí, con el fin de adquirir información a través de un conjunto de datos para lograr un objetivo.

2.2.2 Clasificación de los sistemas de información

Las organizaciones generalmente disponen de más de un tipo de sistema de información, cada uno de ellos con características propias y con un rol fundamental en el logro de las necesidades de información de cada empresa.

Los sistemas de información, de manera general, se pueden clasificar de tres formas según sus propósitos generales: sistemas transaccionales, sistemas de apoyo a la toma de decisiones y sistemas estratégicos. Estos sistemas se encuentran estructurados en una pirámide jerárquica como se muestra en la ilustración 2.3.

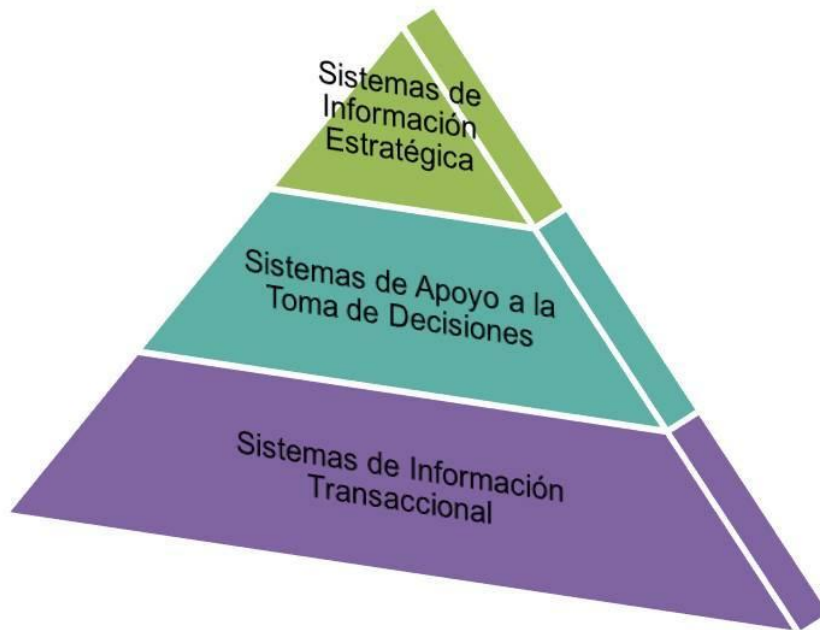


Ilustración 2.3: Tipos de Sistemas de Información.

Fuente: <http://www.udec.cl/dti/node/108>

2.2.2.1 Sistemas de información transaccionales

Son sistemas de información que permiten la automatización de los procesos operativos y/o transaccionales recurrentes de una organización.

La principal característica de estos sistemas es que son capaces de obtener, procesar y mantener los datos producidos por la ejecución de las operaciones básicas y/o rutinarias de las empresas. Como ejemplo de estos podrían ser los sistemas de facturación, sistemas de nóminas, entre otros.

Los sistemas transaccionales se encuentran en el nivel más bajo de la pirámide, y no por ser menos importante, sino porque son la base de los siguientes niveles, sin un sistema operacional, los otros sistemas pierden su sentido.

2.2.2.2 Sistemas de apoyo a la toma de decisiones

Son sistemas de información utilizados normalmente por usuarios de rango organizacional más alto que los anteriores, por ejemplo, gerentes o jefes de departamentos. Estos están diseñados como un conjunto de programas y herramientas que permiten obtener la información que se obtuvo de los sistemas operacionales para ayudar al proceso de toma de decisiones.

Entre las características principales de estos sistemas se tiene que el nivel de cálculos es intenso, la entrada y salida de información es escasa, y la mayoría poseen una interfaz gráfica fácil de usar debido a que va dirigido a un usuario final.

Es así como estos sistemas se incluyen con frecuencia luego de haber implantado los sistemas transaccionales, ya que estos últimos son su principal fuente de información.

2.2.2.3 Sistemas de información estratégica

Estos sistemas representan el nivel más alto de la pirámide, comúnmente utilizados solo por los altos ejecutivos de las organizaciones y suelen ser creados dentro de las organizaciones, resultando complicado adaptarlos de forma sencilla a paquetes disponibles en el mercado.

La principal función de estos sistemas es lograr ventajas competitivas, tales como ventajas en costos y servicios. Adicional a esto, ayudan en el proceso de mejora e innovación de productos y procesos dentro de las organizaciones buscando así ventajas sobre los principales competidores y una forma de hacerlo es la innovación y la creación de nuevos y mejores productos y procesos.

Por ejemplo, el uso de los cajeros automáticos en los bancos es un sistema estratégico, ya que brinda ventajas sobre un banco que no posee tal servicio, por lo tanto, cualquier banco nuevo que desee prestar servicios al público tendrá que brindar este servicio para tener un nivel similar al de sus competidores.

2.2.3 Sistemas de Procesamiento de Transacciones en Línea (OLTP)

Los sistemas de procesamiento de transacciones en línea (OLTP, por sus siglas en inglés) están diseñados para administrar datos que cambian frecuentemente. Soportan concurrencia de usuario, es decir, varios usuarios que realizan transacciones al mismo tiempo que modifican los datos en tiempo real. Los sistemas OLTP están basados en bases de datos relacionales orientadas al procesamiento de transacciones.

2.2.3.1 Transacción

Según (Silberschatz, Korth, & Sudarshan, Fundamentos de Bases de Datos, 2002), una transacción es una unidad de ejecución de un programa que accede y modifica valores de los datos en una base de datos.

Una transacción es una colección de operaciones que se lleva a cabo como una única función lógica en una aplicación de bases de datos. Cada transacción es una unidad de atomicidad y consistencia, se requiere que las transacciones cumplan las restricciones de consistencia de la base de datos.

Las transacciones deben presentar cuatro propiedades conocidas como atomicidad, coherencia, aislamiento y durabilidad (ACID), para ser calificada como una transacción.

- **Atomicidad:** todas las operaciones de la transacción se realizan adecuadamente en la base de datos o ninguna de ellas, es decir, asegurar que la transacción se ha realizado completamente para que no quede a medias en su ejecución y surja algún error.
- **Consistencia:** la ejecución de la transacción debe preservar la consistencia de la base de datos, es decir, asegurar la integridad de la base de datos. Esta propiedad sostiene que cualquier transacción llevara a la base de datos desde un estado válido a otro igualmente válido.
- **Aislamiento:** garantizar que cada transacción ignora al resto de las transacciones que se ejecutan concurrentemente en la base de datos, es decir, asegurar que dos transacciones que afecten el mismo dato (tabla, fila o celda) son independientes y no generan errores ni bloqueo mutuo.
- **Durabilidad:** una vez finalizada exitosamente una transacción se deben reflejar los cambios realizados en la base de datos, es decir, asegurar una vez realizada la transacción, ésta persistirá y no se podrá deshacer, aunque falle el sistema.

2.3 Base de Datos

Según (Silberschatz, Korth, & Sudarshan, 2006) una base de datos es un repositorio centralizado de datos lógicamente relacionados, que permite almacenar y organizar hechos o eventos así como reconstruirlos a demanda de los usuarios para producir información.

Una Base de Datos consolida muchos registros almacenados previamente en archivos independientes, de modo que un cúmulo (pool) de registros sirva como una sola central para muchas aplicaciones de procesos que necesiten este tipo de datos. Está constituida por un conjunto de datos persistentes diseñados para satisfacer las necesidades de información de una organización, por lo que deja de ser propiedad de un departamento y pasa a ser un recurso corporativo compartido.

2.3.1 Características de una Base de Datos

- **Independencia de datos:** implica una separación entre programas y datos.
- **Integridad de la información:** se refiere a la validez y la consistencia de los datos almacenados. Es una de las principales características de una base de datos y tiene como objetivo almacenar únicamente la información correcta.

- **Redundancia mínima:** consiste en utilizar una base de datos como repositorio común para las distintas aplicaciones, minimizando el almacenamiento de un mismo dato varias veces.
- **Globalización de la información:** permite a los usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- **Acceso concurrente por parte de múltiples usuarios:** permite el acceso de varios usuarios a la base de datos, mediante el manejo de protocolos de control de concurrencia basados en técnicas de bloqueo para asegurar la integridad de los datos.
- **Consistencia de datos:** si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a las copias del archivo.
- **Consultas complejas optimizadas:** permitir que las consultas se ejecuten de manera rápida, mediante la optimización de las mismas.
- **Seguridad de acceso y auditoría:** se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. El sistema mantiene el control de acceso a la base de datos, con el objetivo de conocer qué o quién realizó una determinada modificación y en qué momento.
- **Acceso a través de lenguajes de programación estándar:** se refiere a permitir el acceso a la base de datos mediante lenguajes de programación ajenos al sistema de base de datos.

2.4. Sistema manejador de Base de datos

El sistema manejador de base de datos es un conjunto de programas, procedimientos, lenguajes, entre otros. Que suministra a los usuarios no informáticos, analistas, programadores, administradores de base de datos, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, permitiendo mantener su integridad, confidencialidad y seguridad. (Silberschatz, Korth, & Sudarshan, Fundamentos de Bases de Datos, 2002).

Basándonos en (Ospina, 2009), podemos decir que las principales funciones de un sistema manejador de base de datos son las siguientes:

- **Definición de datos:** el SMBD debe incluir componentes procesadores de lenguajes para cada uno de los diversos lenguajes de definición de datos (DDL).
- **Manipulación de datos:** el SMBD posee componentes procesadores de lenguajes para cada uno de los diversos lenguajes de manipulación de datos (DML).

- **Seguridad e integridad de los datos:** el SMBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de control y seguridad definidas por el administrador de la base de datos.
- **Control de concurrencia de los datos:** EL SMBD debe proporcionar un mecanismo para garantizar que la base de datos se actualice correctamente cuando haya varios usuarios actualizando de manera concurrente la base de datos.
- **Recuperación:** EL SMBD debe poder recuperar la base de datos en caso de que resulte dañada de alguna forma.
- **Diccionario de Datos accesible por el usuario:** El SMBD debe incluir un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por parte de los usuarios.
- **Desempeño:** El SMBD debe ejecutar todas las funciones recién identificadas en la forma más eficiente posible.

Todas las funciones anteriormente mencionadas son necesarias para el correcto funcionamiento de una base de datos, el sistema manejador de base de datos facilita el poder llevar a cabo dicha tarea proporcionando un conjunto completo de programas que sirven como interfaz entre uno o varios usuarios y sus diversas aplicaciones.

A continuación, se describe el SMBD usado en la arquitectura del sistema actual.

2.4.1. PostgreSQL

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como Linux, Unix y Windows.

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores (triggers) y procedimientos almacenados (en varios lenguajes). Una base de datos de clase empresarial, cuenta con características avanzadas tales como Multi-Versión Control de concurrencia (MVCC), puntos en tiempo de recuperación, tablespaces, replicación asíncrona, transacciones anidadas (savepoints), respaldos online/hot y un optimizado query planner.

Cabe destacar, que es altamente escalable, tanto en la cantidad de almacenamiento de datos que puede manejar como en el número de usuarios concurrentes que puede administrar. (The PostgreSQL Global Development Group, 2018)

2.5 Base de datos móvil

Es una base de datos portable y físicamente independiente del servidor corporativo de base de datos, pero puede comunicarse con este servidor desde cualquier punto remoto para compartir datos corporativos.

2.5.1 Arquitectura de una base de datos móvil

La arquitectura general de una base de datos móvil normalmente está comprendida por un servidor de base de datos central con un sistema manejador de base de datos que gestiona los datos, unidades móviles o dispositivos móviles con una base de datos local y los enlaces inalámbricos bidireccionales de comunicación entre el sistema manejador central y el sistema manejador móvil, comúnmente esta conexión se hace a través del estándar 802.11 (Wi-Fi).

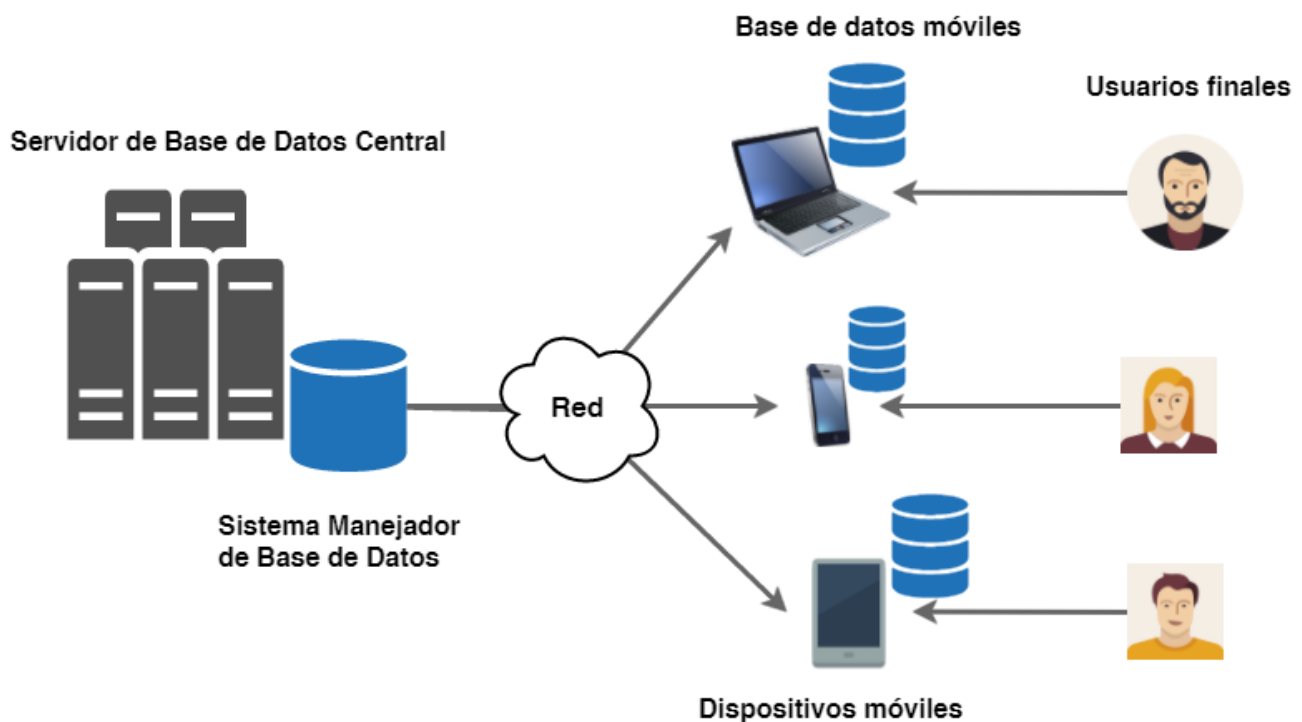


Ilustración 2.4: Arquitectura de una base de datos móvil.

Fuente: (Rivero Hernández, Pérez Vázquez, & Vila Labrada, 2013)

En la ilustración 2.4 se puede observar la arquitectura de una base de datos móvil, propuesta por (Orozco Tapia, 2012) esta arquitectura se compone de un servidor de base de datos central y un sistema manejador de base de datos en el cual se almacenan y manipulan los datos corporativos del negocio, los dispositivos móviles deben contener una base de datos móviles que permita almacenar y manipular un subconjunto de los datos corporativos, la interacción o comunicación entre estos componentes se realiza mediante los enlaces inalámbricos bidireccionales que permitan

la transferencia de los datos entre el servidor central y los dispositivos móviles, normalmente esta comunicación se realiza mediante el estándar 802.11 (WiFi), Bluetooth o comunicaciones satelitales.

- **IEEE 802.11:** Es una norma o protocolo que define el uso de las Wireless LAN, que se conoce como WLAN (WiFi) que permite facilidades no disponibles en los sistemas cableados y formar una red total donde coexisten los dos tipos de sistemas, (López Ortiz, 2010).

2.5.2 Sistema Manejador de Base de Datos Móvil

2.5.2.1 SQLite

Es una base de datos relacional embebida y de código abierto, implementa un motor de base de datos transaccional pequeño, a diferencia de las otras bases de datos, SQLite no tiene un proceso de servidor separado, sino que lee y escribe directamente en archivos de disco. SQLite es una biblioteca compacta y confiable. SQLite es la base de datos más ampliamente implementada (Owens & Allen, 2010)

2.5.2.2 SQLBase

Es una base de datos totalmente relacional, de alto rendimiento e incorporada que permite a las organizaciones administrar los datos y proporciona una arquitectura de base de datos integrada auto-recuperable y sin mantenimiento. Posee soporte de globalización Unicode e integración completa en los principales sistemas de desarrollo, con una estructura robusta, seguridad avanzada, código abierto y flexible (QBS Software Ltd, 2017)

2.5.2.3 SQL Server Compact

Es un motor de base de datos relacional de libre distribución, tanto para dispositivos móviles como para aplicaciones de escritorio. Especialmente orientada a sistemas ocasionalmente conectados, se ejecuta bajo el proceso de la aplicación que la consume (in-process). SQL Server compact posee un motor de base de datos, así como un procesador y un optimizador de consultas especialmente diseñado para entornos móviles. Soporta un subconjunto de tipos de datos y de sentencias T-SQL (Transact-SQL) de SQL Server. (Torres, 2008).

2.5.2.4 Comparación entre los SMBD móviles

A continuación, en la tabla 2.1 se presenta un cuadro comparativo entre las ventajas y desventajas de los Sistemas Manejadores de Base de Datos Móvil antes explicados.

SMBD Móvil	Ventajas	Desventajas
SQLite	<ul style="list-style-type: none"> - Base de datos transaccional pequeña. - Código abierto. - Lectura y escritura directa en los archivos de disco. - Soporta procedimientos T-SQL y condicionales. - Soporta vistas y triggers. - Ilimitado número de conexiones concurrentes. - Soportado por un gran número de lenguajes de programación, incluido JavaScript. 	<ul style="list-style-type: none"> - No permite acceso de data remota (RDA). - No soporta XML.
SQLBase	<ul style="list-style-type: none"> - Base de datos transaccional móvil. - Soporta XML. - Soportado por algunos lenguajes de programación, incluido JavaScript. - Soporta triggers. 	<ul style="list-style-type: none"> - Requiere licencia para su implementación. - Poca documentación.
SQL Server Compact	<ul style="list-style-type: none"> - Base de datos transaccional móvil. - Permite acceso de data remota (RDA). 	<ul style="list-style-type: none"> - No soporta procedimientos T-SQL y condicionales. - No soporta vistas y triggers. - Limitado número de conexiones concurrentes.

Tabla 2.1: Cuadro comparativo de los Sistemas Manejadores de Base de Datos Móvil.

2.6. Computación Móvil

Es el término utilizado para describir el uso de computadoras sin necesidad de estar conectados a una red, es decir, es la computación que permite movilizar el dispositivo sin depender de una conexión cableada.

En la computación móvil es importante destacar que el hardware es pequeño y tiene menos capacidad que una computadora personal, además de que debe garantizar seguridad y facilidad de uso. Estos equipos son denominados dispositivos móviles, también operan con baterías y tienen una velocidad de procesamiento alta ya que las aplicaciones que se manejan se asemejan a un programa en una computadora personal.

El software por su parte debe garantizar una facilidad de uso y ser diseñado especialmente para la computación móvil, se trabaja bajo un sistema operativo diseñado especialmente para este

ambiente móvil. Los dispositivos móviles pueden realizar casi cualquier cosa que una computadora debido a las aplicaciones móviles que son programas especialmente diseñados para estos dispositivos.

La computación móvil se está volviendo día a día un paradigma tecnológico de uso más común y que cambiará la forma en que se realizan las actividades laborales, académicas, de investigación y entretenimiento.

En la tabla 2.2 se presenta un cuadro comparativo entre los aportes y limitaciones que posee la computación móvil.

Aportes	Limitaciones
Se tiene acceso a la información sin depender de una red cableada, usando conectividad inalámbrica. No se está atado al lugar donde se encuentra la información.	La batería de los dispositivos móviles según las aplicaciones que se estén usando puede ser de corta duración.
La información se puede obtener en tiempo real.	La accesibilidad a los dispositivos móviles puede verse limitada por los precios en el mercado de dichos aparatos.
Se facilita la movilidad ya que los dispositivos son livianos y de tamaños considerablemente pequeños.	Algunas funcionalidades presentan debilidades de compatibilidad con el hardware entre versiones de los dispositivos móviles.

Tabla 2.2: Cuadro comparativo entre los aportes y las limitaciones de la computación móvil.

Fuente: Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps. Fling, B. (2009)

2.6.1. Dispositivo móvil

Se puede definir como un aparato electrónico pequeño, con una capacidad de procesamiento no muy alta, normalmente tienen una memoria limitada y mantienen una conexión permanente e intermitente a una red, casi siempre están diseñados para una función en específico, pero se pueden llevar a cabo muchas otras, algunos dispositivos pueden ser desde un GPS, un teléfono hasta una computadora portátil. (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2011)

2.6.1.1. Clasificación de los dispositivos móviles

Según (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2011) los dispositivos móviles se clasifican según su nivel de funcionalidad, por ello en el 2005, T38 y DuPoint Global Mobility Innovation Team propusieron los siguientes estándares para la clasificación de estos dispositivos.

- **Dispositivo Móvil de Datos Limitados:** (*Limited Data Mobile Device*), se caracteriza por tener una pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS y acceso a WAP, en este tipo se tienen a los teléfonos móviles clásicos.
- **Dispositivo Móvil de Datos Básicos:** (*Basic Data Mobile Device*), se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a emails, lista de direcciones, SMS y en algunos casos un navegador web básico. Un típico ejemplo de estos dispositivos son los teléfonos inteligentes (o smartphones).
- **Dispositivo Móvil de Datos Mejorado:** (*Enhanced Data Mobile Device*), se caracterizan por tener pantallas entre mediano y gran tamaño (por encima de los 240 x 120 píxeles), navegación de tipo stylus y que ofrecen las mismas características que el anterior, más aplicaciones nativas como aplicaciones de Microsoft Office Mobile (Word, Excel, PowerPoint) y aplicaciones corporativas usuales, en versión móvil. Este tipo de dispositivos incluyen los sistemas operativos como Windows Mobile.

2.6.1.2. Tipos de dispositivos móviles

- **Teléfono móvil:** Es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio, tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, debido a que la realización de llamadas no depende de un terminal fijo o algún tipo de cableado para llevar a cabo la conexión a la red telefónica. Su principal función es la comunicación de voz, pero su rápido desarrollo ha permitido la incorporación de funciones adicionales como mensajería instantánea, agenda, juegos, cámara fotográfica, reproducción de video, acceso a Internet e incluso GPS, entre otras.



Ilustración 2.5: Teléfonos móviles.

Fuente: https://es.wikipedia.org/wiki/Tel%C3%A9fono_m%C3%B3vil

- **Teléfono inteligente:** Es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un computador personal, permite realizar llamadas y enviar mensajes de texto, pero además incluye características cercanas a un computador personal. Una característica importante es que permiten la instalación de programas que permitan

incrementar el procesamiento de datos y la conectividad, también se pueden descargar e instalar programas adicionales como juegos, agendas, entre otros. Los teléfonos inteligentes se destacan por muchas características según (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, & García Baniello, 2011) entre las que destacan las pantallas táctiles, un sistema operativo particular así como la conectividad a internet y el acceso al correo electrónico, otras aplicaciones suelen estar presentes como las cámaras integradas, la administración de contactos, el software multimedia para reproducción de música y visualización de fotos y videos, algunos cuentan con programas de navegación y con la habilidad de leer documentos de negocios en variedad de formatos como PDF y Microsoft Office.



Ilustración 2.6: Teléfonos inteligentes.

Fuente: <https://www.navarra.es/NR/rdonlyres/48F9746B-080C-4DEA-BD95-A5B6E01797E1/315641/7Usodedispositivosmoviles.pdf>

- **Tableta:** Es una computadora portátil de pantalla táctil normalmente de 7 a 10 pulgadas, que sirve para interactuar con los dedos o lápiz óptico sin necesidad de teclado físico ni ratón, debido a que son reemplazados por un teclado virtual.



Ilustración 2.7: Tableta.

Fuente: <https://www.navarra.es/NR/rdonlyres/48F9746B-080C-4DEA-BD95-A5B6E01797E1/315641/7Usodedispositivosmoviles.pdf>

- **PDA:** (por sus siglas en inglés Personal Digital Assistant), es una computadora de mano diseñada originalmente como una agenda electrónica con un sistema de reconocimiento de

escritura. Inicialmente los PDAs incluían aplicaciones estrictamente relacionadas con su función como agenda electrónica, pero con el paso del tiempo ha ido evolucionando a un rango más extendido agregando aplicaciones, acceso al correo electrónico, navegar por Internet, entre otras cosas. Las características del PDA moderno son pantalla sensible al tacto, conexión a una computadora para sincronización, ranura para tarjeta de memoria y conexión WiFi.



Ilustración 2.8: PDA.

Fuente: <http://edutry.blogspot.com/2014/07/types-of-computer-analogdigitalhybrid.html>

2.6.2. Sistemas operativos móviles

A medida que han ido avanzando los teléfonos inteligentes y las tabletas han surgido diferentes sistemas operativos para estos dispositivos móviles, entre los más importantes se encuentran Android, iOS y Windows Phone.

2.6.2.1. Android

Es un sistema operativo basado en el kernel de Linux, esto lo hace multiplataforma, libre y gratuito, fue diseñado principalmente para dispositivos móviles con pantalla táctil, su desarrollo inició por parte de Android, Inc., aunque en el año 2005 Google compró esta empresa y actualmente es uno de los principales sistemas operativos implementado en la mayoría de los teléfonos inteligentes.

Android tiene una gran comunidad de desarrolladores programando aplicaciones para ampliar la funcionalidad de los dispositivos, la mayoría de estas aplicaciones se encuentran en una tienda virtual oficial de Android llamada Google Play. Este sistema operativo utiliza una base de datos SQLite por ser liviana, también es actualizado cada cierto tiempo para arreglar errores o agregar nuevas funcionalidades, actualmente la última versión 8.1 es llamada Oreo.

2.6.2.2. iOS

Es un sistema operativo móvil desarrollado por la empresa Apple Inc., originalmente fue desarrollado para el iPhone, pero después se extendió su uso a otros dispositivos de la empresa. Apple, Inc. no permite la instalación de este sistema operativo en hardware de terceros, aun así, es el segundo sistema operativo más usado en el mercado.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gesto multi-táctiles, la interacción con el sistema incluye diferentes gestos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz fluida. Tiene su tienda virtual oficial conocida como App Store. Actualmente su última versión es iOS 11.

2.6.2.3. Windows Phone

Es un sistema operativo móvil desarrollado por Microsoft, está enfocado en el mercado de consumo empresarial. Este sistema ofrece una interfaz de usuario que integra varios de sus servicios propios como Skydrive, Skype y Xbox Live en el sistema operativo. Compete directamente contra Android e iOS. Las actualizaciones de software son distribuidas, lo que significa que se pueden instalar directamente desde el terminal, actualmente se encuentra en la versión 8.1.2.

2.6.2.4. Aplicación móvil

En general una aplicación según (Castro Bernal, 2013) es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas. Esto lo diferencia principalmente de otros tipos de programas como los sistemas operativos, las utilidades y los lenguajes de programación.

Una aplicación móvil es un software dedicado a resolver o facilitar alguna tarea en un dispositivo móvil, sea un teléfono móvil, un teléfono inteligente, tableta o PDA, algunas de estas aplicaciones son derivadas desde una aplicación de computadora.

2.6.2.5. Aplicación móvil nativa

Según (Ramírez Vique, 2011), este tipo de aplicaciones están hechas para ejecutarse en un dispositivo y sistema operativo específico, se crean con distintos tipos de lenguajes, pero se encuentran fuertemente relacionados al sistema operativo móvil en el cual serán ejecutados. Estas aplicaciones funcionan de forma más eficiente sobre los dispositivos móviles ya que sus componentes están diseñados de forma específica para un determinado sistema operativo, debido a que estas aplicaciones realizan acceso a sensores o elementos propios del teléfono, como cámara, GPS, agenda, entre otros.

La principal ventaja de usar estas aplicaciones es que se tiene un acceso total a los componentes del sistema operativo del dispositivo y normalmente no requieren de conexión a internet, la desventaja se presenta al momento de que se desee ejecutar esta aplicación en otro sistema operativo, dado que se requiere de una nueva programación bajo otro lenguaje.

2.6.2.6. Aplicación móvil web

De acuerdo con (Ramírez Vique, 2011), este tipo de aplicaciones se ejecutan dentro del navegador web del dispositivo, estas aplicaciones están desarrolladas con HTML, CSS y JavaScript. Dichas aplicaciones se pueden ejecutar en múltiples dispositivos, su principal ventaja es que el proceso de desarrollo es más sencillo ya que se utilizan tecnologías web conocidas, sin embargo, estas aplicaciones no tienen acceso a los elementos del dispositivo o son muy limitados.

2.6.2.7. Aplicación móvil híbrida

Este tipo de aplicaciones unen lo mejor de los modelos anteriores, dado que permiten el uso de tecnologías multiplataforma como HTML, CSS y JavaScript y permiten acceder a un conjunto de los componentes del dispositivo. La infraestructura es tipo web en gran parte y la comunicación con los elementos del dispositivo se hace mediante frameworks, según (Ramírez Vique, 2011).

- **Framework:** se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos de un framework son acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones, (Gutiérrez, 2014).

2.6.2.7.1. Ionic

Ionic es un framework de código abierto para el desarrollo de aplicaciones móviles híbridas, basado en el lenguaje AngularJS y Apache Cordova. Este framework ofrece herramientas y servicios para el desarrollo de aplicaciones móviles híbridas usando tecnologías web conocidas como HTML5, CSS3 y JavaScript, la versatilidad que ofrece Ionic es el desarrollo mediante estas tecnologías y la distribución sobre cualquier sistema operativo móvil.

Ionic se basa en el patrón MVC (Modelo – Vista – Controlador) que permite separar los datos, la lógica y las interfaces de usuarios. Ionic está construido para ser rápido y de alto rendimiento debido a la mínima manipulación del DOM, posee un bonito diseño, limpio, sencillo y funcional para todos los dispositivos móviles y el comando CLI permite crear, construir, probar y compilar las aplicaciones en cualquier plataforma móvil.

2.6.2.7.2. Titanium

Titanium es un framework de código abierto que permite la creación de aplicaciones móviles híbridas mediante un código basado en JavaScript y desarrollado en Appcelerator. Es un framework multiplataforma por el uso de componentes nativos de cada sistema operativo, provee un acceso transparente a las funcionalidades del dispositivo móvil y posee una base de desarrollo con el patrón MCV (Modelo – Vista – Controlador).

2.6.2.7.3. PhoneGap

PhoneGap es un framework para el desarrollo de aplicaciones móviles, PhoneGap principalmente permite a los programadores desarrollar aplicaciones utilizando herramientas genéricas como JavaScript, HTML5 y CSS3, las aplicaciones resultantes son híbridas. Este framework maneja una API que permite el acceso a elementos del dispositivo móvil mediante una interfaz de funciones foráneas en JavaScript.

2.7. Tecnologías web

2.7.1. HTML

Por sus siglas en inglés HyperText Markup Language, que significa Lenguaje de Marcas de Hipertexto se hace relación con la elaboración de páginas web. HTML es un estándar de referencia para el desarrollo de páginas web, como texto, imágenes, videos, juegos, entre otros.

HTML es un estándar el cual está actualmente siendo gestionado por la W3C (World Wide Web Consortium), la cual es una comunidad internacional que se dedica al desarrollo de estándares que aseguran el crecimiento de la Web a largo plazo. Cabe destacar, que HTML es un lenguaje de marcado y permite darle estructura a una página web mediante el uso de etiquetas que pueden servir para colocar hipervínculos, saltos de línea, tablas, estilos, entre otros. Su versión más reciente y la que más uso tiene es HTML5.

2.7.2. CSS

Hojas de Estilo en Cascada, (o por sus siglas en inglés Cascading StyleSheets) es un lenguaje de diseño gráfico para establecer la presentación de un documento, este lenguaje es muy utilizado para la personalización e inclusión de estilos, colores y diseños de una página web. CSS describe la presentación de documentos HTML o XML ofreciendo modularidad al tener una capa separada de la estructura o comportamiento y esté enfocada únicamente en el estilo y presentación de una página web.

2.7.3. JavaScript

Es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje liviano y orientado a eventos con manejo de objetos, es interpretado y débilmente tipado. Oficialmente llamado ECMAScript, es conocido como uno de los lenguajes de programación más practicados del mundo, su uso más común es el de darle comportamiento mediante procesos a un documento HTML. Es interpretado por el navegador y tiene un conjunto de librerías integradas que permiten manejar eventos, modificar dinámicamente el contenido de una página y transferir datos entre el servidor y el cliente haciendo uso de llamadas asíncronas. (Mozilla, 2018).

2.7.3.1. AngularJS

AngularJS es un framework estructural para páginas web dinámicas. Permite usar HTML como lenguaje base para luego extender la sintaxis de HTML y darles más funcionalidad y dinamismo a los componentes de una aplicación web. Inicialmente basado en el patrón MVC (Modelo – Vista – Controlador), usa JavaScript por lo que es ejecutado en el lado del cliente o navegador. (Google, 2018).

2.8. Servicio web

Un servicio web (web service, en inglés) es un estándar de interoperabilidad entre distintas plataformas, utiliza un conjunto de protocolo y estándares que trabajan conjuntamente para intercambiar datos entre aplicaciones, garantizando una comunicación segura. Cualquier aplicación de software desarrollada en un lenguaje de programación particular y ejecutada sobre cualquier plataforma, puede utilizar servicio web para intercambiar datos en una red, como por ejemplo internet. Existen dos protocolos que utilizan los servicios web para la comunicación y que se explica a continuación.

2.8.1. Protocolo SOAP

Por sus siglas en inglés, Simple Object Access Protocol, que significa Protocolo Simple de Acceso a Objetos, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio del intercambio de datos XML, es independiente de la plataforma y del lenguaje. Este protocolo es derivado de uno creado por Dave Winer en 1998, llamado XML-RPC.

Básicamente SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una pila de protocolos de un servicio web, ofreciendo un framework de mensajería básica en el cual los servicios web se pueden construir. Este protocolo está basado en XML y se conforma de tres partes; un sobre que

define qué hay en el mensaje y cómo procesarlo, un conjunto de reglas de codificación para expresar instancias de tipos de datos y una convención para representar llamadas a procedimientos y respuestas. Además, cuenta con tres características principales que son la extensibilidad, neutralidad e independencia.

- **XML:** Lenguaje Extensible de Marcas (eXtensible Markup Language, en inglés) es un meta-lenguaje que permite definir lenguajes de marcas de desarrollo por el World Wide Web Consortium (W3C) utilizado para almacenar datos de forma legible. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande, con unas posibilidades mucho mayores. Este lenguaje es extensible, su analizador es un componente estándar, es fácil de entender su estructura y procesarla y permite transformar datos en información.

2.8.2. Protocolo REST

Por sus siglas en inglés, Representational State Transfer, que significa Transferencia de Estado Representacional, plantea una arquitectura cliente-servidor, en el cual un servicio es visto como un recurso que se identifica a través de una dirección URL, mediante la cual puede ser accedido o consumido. Para acceder a estos servicios web se hace uso de mensajes en formato simple, los cuales se intercambian entre el cliente y el servidor.

El protocolo REST define a partir de HTTP, cuatro métodos; GET, PUT, DELETE y POST. Este protocolo se basa en el envío de mensajes a través de diversos formatos y lenguajes; XML, HTML, JSON, debido a la sencillez de los mensajes JSON este es el tipo de mensaje más utilizado en el protocolo.

Cada mensaje intercambiado contiene la información necesaria para cada funcionamiento de los servicios, de tal forma que para cada servicio el cliente y el servidor conocen el formato o protocolo interno de los mensajes.

- **JSON:** Es un formato ligero basado en texto, con definición de un lenguaje completamente independiente, cuya sintaxis está basada en un subconjunto de funcionalidades y palabras reservadas del lenguaje script web; JavaScript. Un mensaje JSON está constituido por dos estructuras básicas; la primera es una colección de pareja clave-valor, las cuales en varios lenguajes son conocidos como: registros, diccionarios, tablas, listas, y la segunda es una lista ordenada de valores que en la mayoría de los lenguajes es de forma de arreglos o vectores.

Como puede observar en la siguiente tabla 2.3 se establecen las comparaciones entre ambos protocolos, para determinar las ventajas y desventajas de los protocolos explicados anteriormente.

Protocolo SOAP	Protocolo REST
Basado en el intercambio de datos mediante XML.	Basado en el intercambio de mensajes mediante formato JSON, XML, HTML.
Define qué hay en el mensaje y cómo procesarlo, un conjunto de reglas de codificación y una convención para las llamadas a procedimientos y respuestas	Define a partir de HTTP cuatro métodos: GET, PUT, DELETE y POST.
Es un protocolo estandarizado.	Es un estilo de arquitectura y está diseñado para arquitecturas de software basadas en red.
Independiente de lenguaje, plataforma y medio de transferencia.	Requiere el uso de HTTP.
Manejo de errores integrado.	Eficiente ya que puede usar formatos de mensaje más livianos.

Tabla 2.3: Cuadro comparativo entre los protocolos SOAP y REST.

CAPÍTULO 3 : MARCO METODOLÓGICO

El desarrollo de un software implica desde la planificación hasta la producción, mantenimiento y pruebas del mismo, para esto se debe seguir un conjunto de actividades basadas en un orden específico que establece una metodología a seguir.

La metodología permite de una forma sistemática obtener un producto libre de correcciones. En este caso, la metodología se dirige hacia los procesos que intervienen en el proceso de Inteligencia Operacional a desarrollar, utilizando una metodología que permita gestionar el desarrollo mediante fases relacionadas a cada una de las etapas para obtener el resultado esperado.

Los elementos que componen una metodología son:

- Las Fases, donde se establecen las actividades a realizar en cada fase.
- Los métodos, consiste en identificar la forma en se realizará el de desarrollo del producto.
- Técnicas y Herramientas, identificación de cómo resolver cada una de las tareas que componen el proceso y las herramientas a utilizar.
- Documentación, identificar el tipo de documentación a entregar en cada una de las fases.
- Control y Evaluación, se realiza a lo largo del ciclo de vida del proceso de desarrollo, consiste en comprobar, aceptar o rechazar todos los resultados que se obtienen

3.1 Metodología tradicional

Las metodologías tradicionales son el primer tipo de metodología que surge como guía para garantizar el desarrollo de un producto con un alto nivel de calidad.

Estas metodologías son clásicas, los métodos de trabajo son muy formales y conllevan la realización de una gran carga de trabajo para gestionar y generar gran cantidad de documentación. Las metodologías tradicionales se enfocan en la documentación y en la planificación de los procesos.

La base de esta metodología es la gestión predictiva, es decir, se comienza con unos requerimientos iniciales, luego se configura un plan adecuado usando los recursos y el tiempo necesario, durante la fase de desarrollo se comprueba si hay desviaciones con respecto a los objetivos planteados, se realizan las modificaciones necesarias y se continúa el flujo.

A continuación, se mencionan algunas metodologías tradicionales:

- **Metodología RUP (Rational Unified Process):** Se basa en un marco de procesos de trabajo que pueden ser adaptados por parte de las organizaciones que desarrollen el producto. Esta metodología se compone de tres módulos principales: Inicio para establecer el objetivo del sistema e identificar los requerimientos. Elaboración, se genera una estructura arquitectónica que permite diseñar, implementar y probar el sistema. Fase de construcción, a partir de la arquitectura establecida anteriormente se realiza la implementación y el manejo de versiones con respecto a las funcionalidades. Transición, se comprueba que el sistema cumpla con las necesidades.
- **Metodología en Cascada:** El ciclo de vida en cascada se caracteriza porque todas las fases se realizan de forma secuencial, pero una etapa no puede comenzar sin que la etapa anterior esté finalizada. La ventaja de esta metodología es que posee una planificación sencilla, la calidad del producto obtenido es alto y no requiere de un personal altamente calificado. Sin embargo, para el desarrollo es necesario tener todos los requerimientos iniciales.

3.2 Metodología ágil

Las metodologías ágiles surgen como una alternativa a las metodologías tradicionales, actualmente el entorno de desarrollo de un sistema es inestable y cambiante por lo que las metodologías tradicionales no se adaptan a dicho cambios.

El desarrollo ágil se enfoca en la iteración, comunicación y reducción de elementos intermedios, se puede deducir que las metodologías ágiles se adecuan mejor en entornos de cierta incertidumbre o cambiante, según (Palacio, 2014).

A continuación, se mencionan algunas metodologías ágiles:

- **SCRUM:** Esta metodología tiene como base la creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones, pero en Scrum se llaman "Sprints". Los Sprints son la base de la metodología Scrum, este componente se considera un contenedor de eventos que permite obtener una versión estable del producto. Las ventajas de esta metodología es la flexibilidad a cambios entre cada uno de los Sprints, la necesidad de obtener una versión funcional en cada iteración permite obtener un software de alta calidad, la realización de funcionalidades de mayor prioridad permite la reducción de riesgos de manera anticipada.
- **XP- eXtreme Programming:** Es la primera metodología ágil, está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo, promoviendo el trabajo en equipo. XP se basa en la retroalimentación continua entre el cliente y el equipo de

desarrollo y la comunicación fluida entre los participantes, sin embargo, la particularidad de esta metodología es el alto nivel de disciplina de las personas que participan en el proyecto. XP es una metodología muy popular, además es considerada una metodología posmoderna debido a que las capacidades se generan a través de procesos emergentes.

- **Kanban:** Es un popular framework usado para implementar desarrollo de software ágil. Requiere comunicación en tiempo real de la capacidad y completa transparencia del trabajo. Los ítems de trabajo son representados visualmente en un “Tablero Kanban”, permitiendo a los miembros del equipo ver el estado de cada proceso en cualquier momento. Su objetivo es realizar entregas continuamente sin sobrecargar al equipo de desarrollo. (Atlassian, 2018)

3.3 Comparación entre las metodologías de desarrollo

De acuerdo con (Amaro Calderón & Valverde Rebaza, 2007), en la tabla 3.1 se establece un cuadro comparativo entre las metodologías tradicionales y las metodologías ágiles, las diferencias entre estas metodologías afectan no sólo al proceso de desarrollo, sino también al contexto del equipo como una organización.

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Esencialmente preparadas para cambios durante el proyecto.
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Muchos roles.	Pocos roles.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.

Tabla 3.1: Metodologías tradicionales y Metodologías ágiles.

Fuente: Metodologías Ágiles. Amaro, Sarah & Valverde, Jorge (2007)

3.4 SCRUM

Es un marco de trabajo de procesos usado para gestionar el desarrollo de productos complejos, Scrum no es un proceso o una técnica para construir productos sino un marco de trabajo en el cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las

prácticas de gestión de producto y las prácticas de desarrollo de modo que se pueda mejorar, según (Schwaber & Sutherland, 2016). Scrum es liviano, fácil de entender, pero difícil de llegar a dominar

3.4.1. Teoría de Scrum

De acuerdo con (Schwaber & Sutherland, 2016) Scrum se basa en la teoría de control de procesos empírica o empirismo, donde el empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce, mediante un enfoque iterativo o incremental para optimizar la predictibilidad y el control del riesgo.

3.4.2. El equipo Scrum

Los equipos Scrum son auto organizados y multifuncionales, el modelo de equipo está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

3.4.2.1. Dueño de producto (Product Owner)

Es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo, es la única persona responsable de gestionar la lista del producto (Product Backlog). Se encarga de expresar claramente los elementos de la lista del producto, ordenar los elementos de la lista del producto, optimizar el valor del trabajo que el equipo de desarrollo realiza, asegurar que la lista del producto sea visible transparente y clara para todos y asegurar que el equipo de desarrollo entiende los elementos de la lista del producto.

3.4.2.2. Equipo de desarrollo (Development Team)

Consiste en los profesionales que realizan el trabajo de entregar un incremento de producto “terminado” que potencialmente se pueda poner en producción al final de cada Sprint, solo los miembros del equipo de trabajo participan en la creación del incremento.

Los equipos de desarrollo son auto organizados, multifuncionales dado que cuentan con todas las habilidades necesarias para crear un incremento del producto, todos son desarrolladores independientemente del trabajo de cada persona, no hay subequipos en estos equipos in importar los dominios particulares que se requieran a tener en cuenta y la responsabilidad recae en el equipo de desarrollo como un todo.

3.4.2.3. Scrum Master

Es el responsable de asegurar que Scrum se entienda y se adopte, asegurándose que el equipo de Scrum ajuste la teoría, prácticas y reglas de Scrum en su proceso. Es un líder que está al servicio del equipo Scrum ayudando a las personas externas al equipo Scrum a entender qué interacciones con el equipo Scrum pueden ser útiles y cuáles no.

El Scrum Master da servicios al dueño de producto, al equipo de desarrollo y a la organización de distintas maneras y cada uno de sus servicios depende de a qué miembro del equipo Scrum esté interactuando.

3.4.3. Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum, todos los eventos son bloques de tiempo (time-boxes), de modo tal que todos tienen una duración máxima asociada.

3.4.3.1. Sprint

Es la base de Scrum, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “terminado” utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.

Durante el Sprint no se realizan cambios que puedan afectar el objetivo del Sprint (Sprint Goal) y los objetivos de calidad no disminuyen. Cada sprint puede considerar un proyecto con un horizonte no mayor de un mes.

3.4.3.2. Planificación de Sprint (Sprint Planning)

La planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes, se debe realizar un plan mediante el trabajo colaborativo del equipo Scrum completo. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, además de enseñar al equipo Scrum a mantenerse en el bloque de tiempo estipulado. La planificación de Sprint responde a las siguientes preguntas:

- a. **¿Qué puede hacerse en este Sprint?:** El equipo de desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. La entrada de esta reunión está constituida por la lista de producto, el último incremento de producto, la capacidad proyectada del equipo de desarrollo para el sprint y el rendimiento pasado del equipo de desarrollo. Después que el equipo de desarrollo planea los elementos de la lista de producto que entregará en el Sprint, el equipo de desarrollo elabora un objetivo del Sprint (Sprint Goal), el objetivo del Sprint debería lograrse durante el Sprint a través de la implementación de la lista del producto y proporciona una guía al equipo de desarrollo de por qué se está construyendo el incremento.
- b. **¿Cómo se conseguirá completar el trabajo seleccionado?:** El equipo de desarrollo decide cómo construirá esta funcionalidad para formar un incremento de producto “terminado” durante el Sprint, los elementos de la lista de producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de lista de pendientes del Sprint (Sprint Backlog).

Al finalizar la planificación de Sprint, el equipo de desarrollo debería ser capaz de explicar al dueño de producto y al Scrum Master cómo se pretende trabajar como un equipo organizado para lograr el objetivo del sprint y crear el incremento esperado.

3.4.3.3. Objetivo del Sprint (Sprint Goal)

Es una meta establecida para el Sprint que puede lograrse mediante la implementación de la lista de producto, se construye durante la planificación del Sprint, el objetivo del Sprint brinda al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint.

3.4.3.4. Scrum diario (Daily Scrum)

Es una reunión con un bloque de tiempo de 15 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum diario y haciendo una proyección acerca del trabajo que podría contemplarse antes del siguiente. El equipo de desarrollo usa el Scrum diario para evaluar el progreso hacia el objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la lista de pendientes del Sprint.

3.4.3.5. Revisión de Sprint (Sprint Review)

Al final del Sprint se lleva a cabo una revisión para inspeccionar el incremento y adaptar la lista de producto si fuese necesario. Durante la revisión de Sprint, el equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint, basándose en esto y en cualquier cambio a la lista de producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor.

3.4.3.6. Retrospectiva de Sprint (Sprint Retrospective)

Es una oportunidad para el equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint, esta revisión tiene lugar después de la revisión de Sprint y antes de la siguiente planificación de Sprint. El propósito de esta revisión es:

- Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.
- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
- Crear un plan para implementar las mejoras a la forma en la que el equipo Scrum desempeña su trabajo.

3.4.4. Artefactos de Scrum

Representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación, los artefactos están diseñados para maximizar la

transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

3.4.4.1. Lista de producto (Product Backlog)

Es una lista ordenada de todo lo que podría ser necesario en el producto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto, esta lista evoluciona a medida que también el producto y el entorno en que se usará. La lista de producto enumera las características, funcionalidades, requisitos, mejoras y correcciones que construyen cambios a realizarse sobre el producto para entregas futuras.

3.4.4.2. Lista de pendientes del sprint (Sprint Backlog)

Es el conjunto de elementos de la lista de producto seleccionados para el Sprint, más un plan para entregar el incremento de producto y conseguir el objetivo del Sprint, esta lista es una predicción hecha por el equipo de desarrollo acerca de qué funcionalidad formará parte del próximo incremento y del trabajo necesario para entregar esa funcionalidad en un incremento “terminado”.

3.4.5. Metodología Scrum

Según (Schwaber & Sutherland, 2016), la metodología Scrum se encuentra compuesta por diferentes fases o eventos que comprenden el ciclo de desarrollo de un proyecto, en la ilustración 3.1, se presentan las fases de la metodología Scrum.

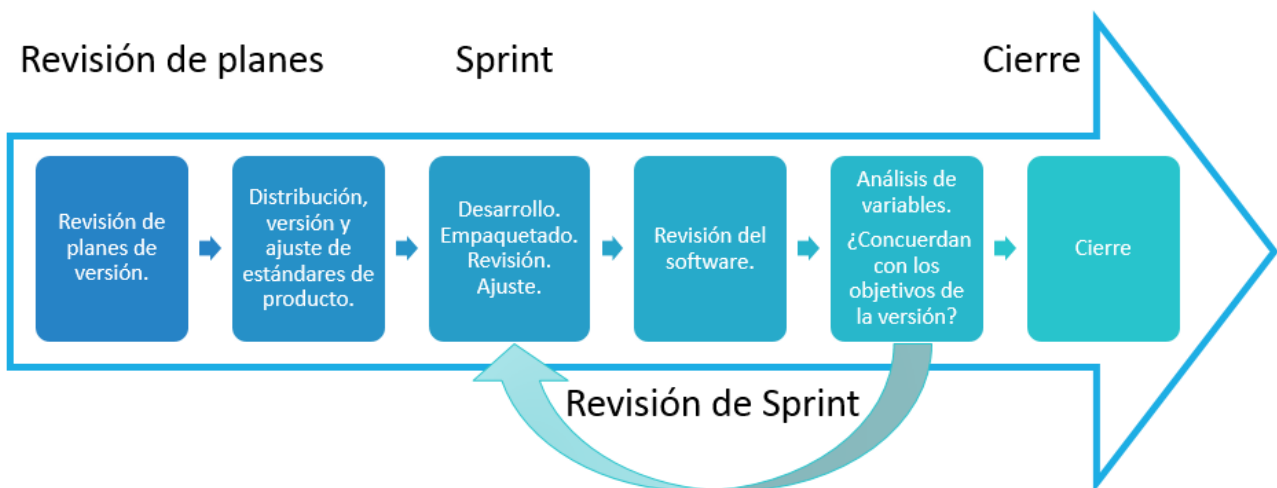


Ilustración 3.1 : Metodología Scrum.

Fuente: La Guía de Scrum. Schwaber, K. & Schutherland J. (2013)

- **Revisión de planes:** Es la fase inicial de la metodología, consiste en examnar las actividades, en qué punto se encuentra la distribución actual y los estándares a utilizar, esta fase es realizada por el equipo de desarrollo.

- **Sprint:** Es la fase de desarrollo iterativo por parte del equipo de desarrollo, el objetivo es implementar un módulo o componente funcional del sistema.
 - **Desarrollo:** Análisis, implementación y pruebas.
 - **Empaquetado:** Generación de paquetes ejecutables.
 - **Revisión:** Resolución de problemas y se añaden nuevos ítems.
 - **Ajuste:** Uso de los ajustes para mejorar el producto.

- **Revisión de Sprint:** Esta fase consiste en la revisión funcional del módulo o componente generado por la fase anterior. Para esta fase se realiza una reunión entre el gestor del proyecto y las personas interesadas por el producto, se revisa el producto del sprint anterior y se pueden añadir nuevos puntos.

- **Cierre:** En esta fase se encuentran las típicas actividades de fin de proyecto, como la generación de una versión distribuible, realización de pruebas, etc.

3.5 Comparación entre metodologías ágiles

En la tabla 3.2 se establece un cuadro comparativo entre las metodologías ágiles SCRUM, Extreme Programming (XP) y Kanban, las diferencias entre estas metodologías afectarán desde la estructura del equipo de desarrollo hasta la frecuencia de las entregas.

SCRUM	XP	Kanban
Sprints e iteraciones con tiempo definido.	Iteraciones con períodos cortos de tiempo.	Entregas continuas con más frecuencia.
Roles definidos como Scrum Master, Product Owner y el team member.	Ciertos roles predeterminados como Customer.	No existen roles predeterminados.
Útil en situaciones donde el trabajo puede ser dividido en lotes independientes	Se enfoca en los ítems siguiendo una estricta prioridad.	Más apropiado en ambientes operacionales donde la prioridad de cada ítem es bastante variable.
No permite cambios a mitad de un Sprint establecido.	Permite cambios en la iteración siempre y cuando el equipo no haya empezado dicho ítem.	Flexible a cambios en cualquier momento.

Tabla 3.2: Metodologías ágiles SCRUM, XP y Kanban.

CAPÍTULO 4 : MARCO APLICATIVO

Seguidamente, para el próximo capítulo se expone una descripción general de la solución en conjunto con las respectivas tecnologías utilizadas para su implementación. Adicionalmente, se especifica el uso de la metodología de desarrollo ágil SCRUM que se aplicó para la elaboración de este proyecto, así como los distintos entes que tienen interacción con el sistema, explicación y descripción de las interfaces de la aplicación y el conjunto de pruebas realizado que apoyaron en gran manera el desarrollo del sistema.

4.1 Descripción general de la solución

La aplicación desarrollada fue implementada haciendo uso de tecnologías web, es una solución híbrida con la que los usuarios finales podrán realizar todo el proceso de evaluación mediante dispositivos móviles para luego ser enviado a un servidor con la base de datos central donde se registrarán los resultados de la inspección. Lo cual permite obtener una persistencia digital de las solicitudes para hacer una mejor gestión de la información.

El desarrollo fue llevado a cabo cumpliendo con la siguiente arquitectura tecnológica propuesta en la ilustración 4.1:

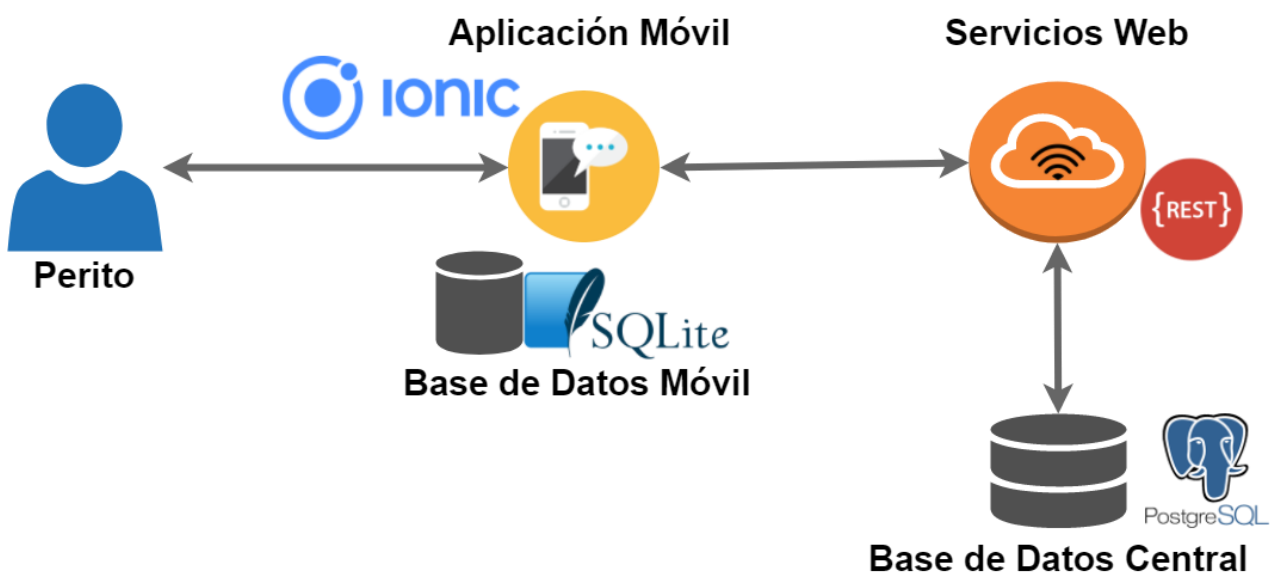


Ilustración 4.1: Arquitectura de la solución tecnológica.

La base de datos central fue previamente desarrollada usando el sistema manejador PostgreSQL, debido a su gran potencia, alto rendimiento, seguridad, compatibilidad y tener licencia de software libre. Los servicios web serán de tipo REST, que permite la compatibilidad con la tecnología que se encontrará en la aplicación móvil además de ser sencillo de desarrollar y mantener a futuro. Como formato de intercambio de información a través de los servicios web se

propone el uso de JSON por su simpleza y gran desempeño para el desarrollo de servicios web, más aún, tomando en cuenta la compatibilidad absoluta que tiene con Angular.

La aplicación móvil será desarrollada sobre el framework Ionic en su versión 3, el cual permite la portabilidad a diferentes sistemas operativos móviles y al estar hecho con Angular como base, se cuenta con una comunidad en línea bastante dedicada, lo que facilita el desarrollo al momento de solventar errores que se puedan presentar, la base de datos móvil estará desarrollada en SQLite por su ligereza de almacenamiento, la cual es necesaria para guardar mucha información temporal asociada a una inspección.

4.2 Aplicación de la metodología SCRUM

A continuación, en la tabla 4.1 se muestran las tareas o actividades programadas para cada Sprint, los cuales tuvieron cada uno un tiempo definido de dos semanas de desarrollo de proyecto.

4.2.1 Lista de Objetivos

Sprint	Actividad	Fecha
1.	<ul style="list-style-type: none"> • Instalación de las herramientas de desarrollo necesarias. • Configuración del entorno de trabajo. • Elaboración de modelo de datos móvil. • Maquetado de interfaces. 	07/02/2018 - 21/02/2018
2.	<ul style="list-style-type: none"> • Desarrollo de módulo de autenticación. • Ejecución de pruebas de funcionalidad. • Maquetado de menú lateral. • Llamadas a servidor correspondientes. 	22/02/2018 - 08/03/2018
3.	<ul style="list-style-type: none"> • Implementación de navegación entre interfaces. • Pase de datos entre interfaces. • Integración de plugins de base de datos y de almacenamiento nativo móvil. 	09/03/2018 - 23/03/2018
4.	<ul style="list-style-type: none"> • Migración de Ionic v1 a Ionic v3. • Reestructuración de código y organización de carpetas. 	30/03/2018 - 13/04/2018

Tabla 4.1: Lista de objetivos y tareas realizadas (1/2)

Sprint	Actividad	Fecha
5.	<ul style="list-style-type: none"> • Inicio de implementación del formulario. • Validaciones de datos de entrada. 	20/04/2018 - 04/05/2018
6.	<ul style="list-style-type: none"> • Implementación de esquema relacional en SQLite. • Página de perfil. • Página de información de desarrollador. 	5/05/2018 - 19/05/2018
7.	<ul style="list-style-type: none"> • Culminación de páginas del formulario. • Pase de data del formulario a base de datos local. • Implementación de funciones de creación, eliminación, actualización, búsqueda e inserción de SQLite. 	19/05/2018 - 02/06/2018
8.	<ul style="list-style-type: none"> • Mensajes al usuario sobre estado actual de la aplicación. • Ejecución de pruebas de funcionalidad. 	03/06/2018 - 17/06/2018
9.	<ul style="list-style-type: none"> • Integración de plugin de cámara. • Llenar formulario con información de base de datos local. • Integración de plugin de sistema de archivos y galería. • Almacenamiento de imágenes capturadas del formulario. 	16/09/2018 - 30/09/2018
10.	<ul style="list-style-type: none"> • Manejo de errores y excepciones. • Desarrollo de estilos en la aplicación. • Ejecución de pruebas de funcionalidad. 	30/09/2018 - 14/09/2018

Tabla 4.2: Lista de objetivos y tareas realizadas (2/2)

4.2.2 Lista de tareas de la iteración

-Sprint 1: Se efectuó la instalación de las tecnologías elegidas para el desarrollo, empezando con Ionic v1 y su plataforma principal Cordova. Ionic realiza la instalación de sus dependencias como AngularJS. Se realizó la creación de los repositorios, del proyecto y estructura de carpetas.

Se llevó a cabo el diseño y maquetado de la aplicación móvil al igual que el modelo de datos que se estaría implementando durante el desarrollo.

-Sprint 2: En esta iteración inició la creación del módulo de autenticación con parte de sus funcionalidades asociadas. Se realizaron las pruebas de funcionalidad necesarias para comprobar el funcionamiento del sistema.

Se hizo un maquetado del menú lateral que posee la aplicación y se hicieron las llamadas al servidor correspondientes para probar la integración correcta del API.

-Sprint 3: Se realizó la navegación entre interfaces siguiendo el flujo de trabajo propuesto y se hizo el pase de datos como parámetros de manera satisfactoria.

Se llevó a cabo la integración de los plugins de base de datos móvil (SQLite) y de almacenamiento local (Local Storage).

-Sprint 4: Se efectuó una migración de framework de Ionic v1 a su versión más estable y reciente Ionic v3 para estar al día con las actualizaciones de seguridad. Para ello se hizo una reestructuración y reescritura de todo el código y se cambió la estructura y organización de las carpetas del proyecto para mejorar la modularidad y mantenibilidad del proyecto.

-Sprint 5: En este Sprint se dio inicio a la implementación del formulario que se llevaría a cabo en más de una iteración por su complejidad. Se realizaron también las validaciones respectivas de los datos de entrada.

-Sprint 6: Se realizó la implementación de todo el esquema de base de datos móvil contemplado en el modelo de datos. Se crearon las páginas o interfaces de perfil de usuario inspector y de información sobre el desarrollador.

-Sprint 7: Se finalizaron las páginas restantes del formulario con sus respectivas validaciones de entrada y se realizó la encapsulación de toda esa data a los respectivos servicios de almacenamiento en base de datos local.

Se desempeñó la tarea de implementar todas las funciones pertinentes a la gestión de la base de datos móvil, como su creación y eliminación de tablas, búsqueda, inserción, actualización y borrado de registros.

-Sprint 8: Se añadieron mensajes con retroalimentación para el usuario final como avisos de que se realizó la solicitud, de que se está esperando respuesta del servidor y la ejecución de las pruebas de funcionalidad pendientes de iteraciones anteriores.

-Sprint 9: Se llevó a cabo la integración del plugin de cámara en la aplicación, para poder tomar fotografías durante la inspección y subirlas junto con el formulario al servidor, o guardarlas en

la base de datos local hasta que sea posible enviarlas. Para ello, se hizo uso de un plugin para manejar archivos del teléfono el cual también fue integrado y probado.

Se comenzó a llenar el formulario con los datos que traiga la base de datos local para permitir al usuario editar sus respuestas antes de enviar al servidor central.

Sprint 10: Finalmente, se realizó el manejo de errores y excepciones respectivo de la aplicación para prevenir o minimizar las fallas por errores comunes. Se dio otro retoque estético a la aplicación para mejorar la usabilidad de la misma y se realizaron las últimas pruebas de funcionalidad correspondientes para finalizar todo el proceso de envío de solicitud.

4.3 Requerimientos del sistema

Una vez realizado el levantamiento de requerimientos definidos por el cliente, haciendo uso del rol de Product Owner, los mismos se distribuyen de la siguiente manera:

4.3.1 Requerimientos funcionales:

- Autenticación con credenciales de la base de datos central.
- Consultar solicitudes de inspección por estado de la solicitud.
- Registro de los datos de inspección del vehículo.
- Almacenamiento de los datos de inspección del vehículo de forma local.
- Página de información o perfil del usuario inspector.

4.3.2 Requerimientos no funcionales:

- Validar la entrada de datos de todas las interfaces para prevenir ataques a la integridad de la base de datos central.
- Mensajes de retroalimentación sobre las acciones del usuario.
- Manejo de errores y excepciones para minimizar fallas.

4.4 Perfiles de usuario

-Inspector: Este usuario es el principal usuario de la aplicación móvil, usando sus mismas credenciales del sistema web puede entrar en la aplicación para realizar el registro de inspecciones, todas sus demás funciones sólo implican lectura de datos.

- Visualización de su información personal.
- Visualización de la lista de solicitudes pendientes.
- Registro y actualización de datos en formulario de solicitud.
- Ocupar o liberar solicitud pendiente.
- Enviar información a base de datos central con la solicitud finalizada.

4.5 Descripción del flujo asociado a la solución

4.5.1 Proceso de gestión de solicitud de inspección

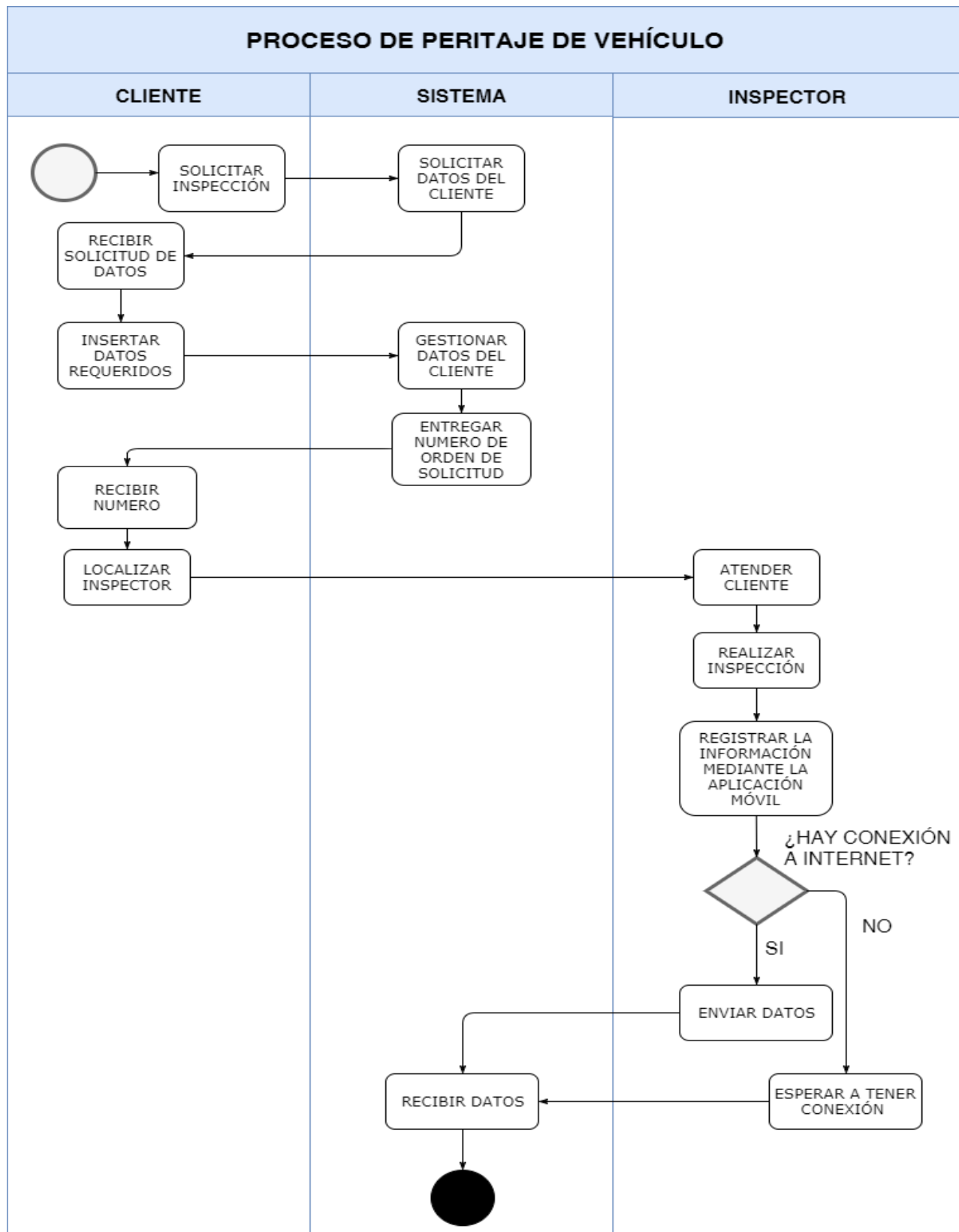


Ilustración 4.2: Flujograma del proceso de gestión de solicitud de Peritaje Vehicular.

Como se puede apreciar en la ilustración 4.2, cuando el cliente realiza su solicitud de inspección vehicular, le es asignado un número de ticket de solicitud desde la base de central. Luego, el inspector puede entrar en la aplicación y revisar la bandeja de solicitudes pendientes para poder ingresar al formulario y registrar los datos correspondientes al peritaje del vehículo en la misma. Una vez presionando “Finalizar”, la solicitud será guardada en la base de datos local del dispositivo móvil para garantizar que no haya pérdidas de información. Posteriormente, el inspector puede entrar en la pestaña de solicitudes con estado “Finalizado” para seguir realizando cambios sobre las mismas y una vez el cliente esté satisfecho, enviar todos los datos al servidor para que sean almacenados en el sistema central.

Listado de excepciones:

- Si el usuario inspector lo desea, puede liberar la solicitud para que la atienda otro inspector, deshaciendo todo cambio hecho.

4.6 Análisis del modelo de datos

El modelo de datos móvil fue elaborado teniendo en cuenta que debe ser eficiente y sólo debe almacenar lo necesario para llevar a cabo la persistencia. Para ello, se crearon un total de diez (10) tablas.

4.6.1 Listado de tablas de la aplicación móvil

- **usuario:** Almacena los valores de los usuarios, que serán los inspectores y actores principales del sistema.
- **titular:** Almacena los datos de los titulares de los vehículos.
- **vehículo:** Almacena los datos del vehículo relacionado en la inspección.
- **solicitud:** Almacena los datos referentes a la solicitud.
- **multimedia:** Almacena los datos de los archivos multimedia pertenecientes a la solicitud.
- **condicion_general_vehiculo:** Almacena los datos del formulario de condiciones generales del vehículo.
- **mecanica_vehiculo:** Almacena los datos del formulario de la mecánica del vehículo.
- **accesorio_vehiculo:** Almacena los datos del formulario de los accesorios del vehículo.
- **detallos_datos:** Almacena los datos del formulario de los detalles de piezas individuales del vehículo.
- **documento_vehiculo:** Almacena los datos del formulario de los documentos del vehículo.

4.6.2 Modelo de datos

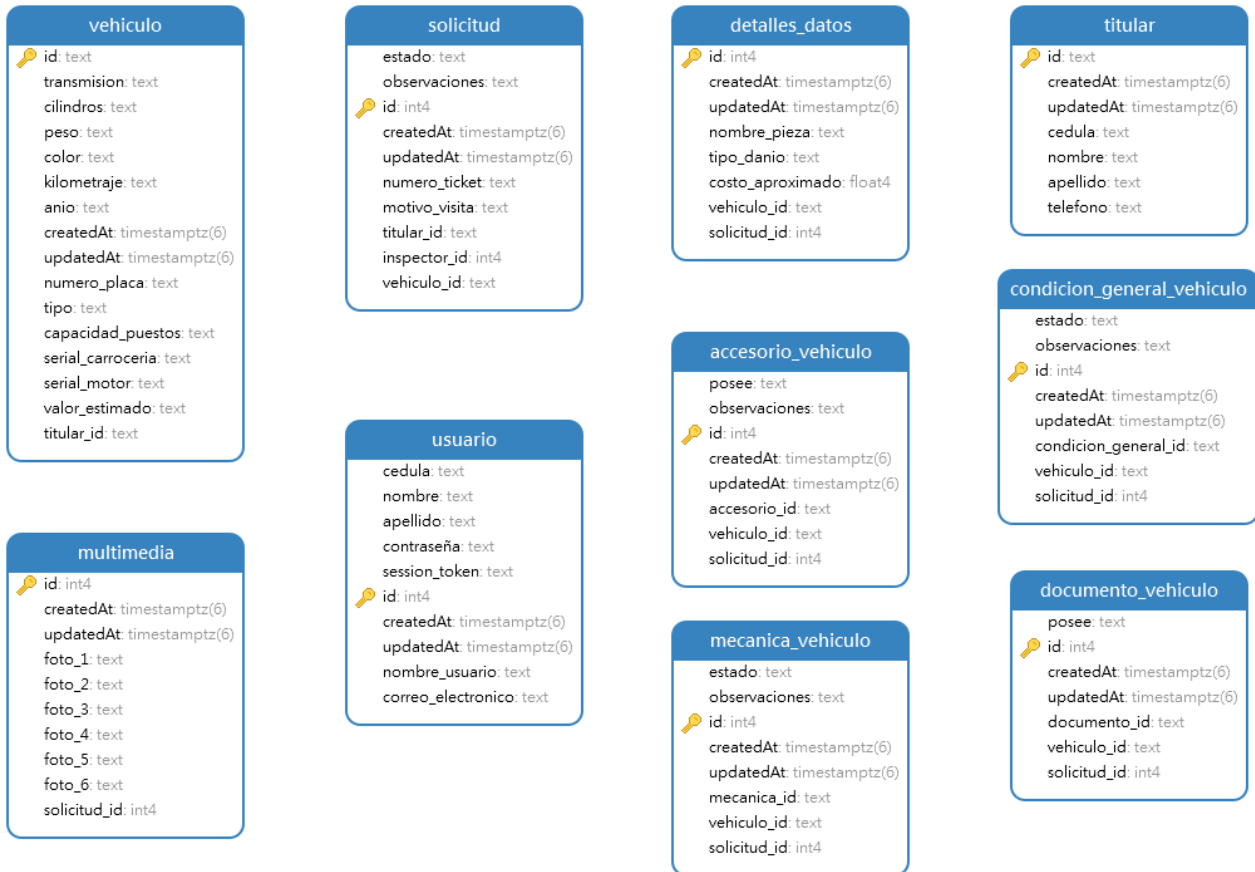


Ilustración 4.3: Modelo de datos para la base de datos móvil.

4.7 Descripción de los módulos del sistema e interfaces

El sistema móvil está constituido de diversos módulos, que pueden ser apreciados en la ilustración 4.4

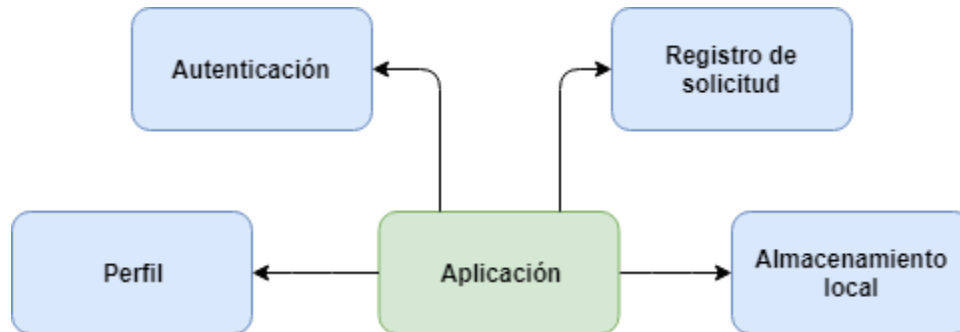


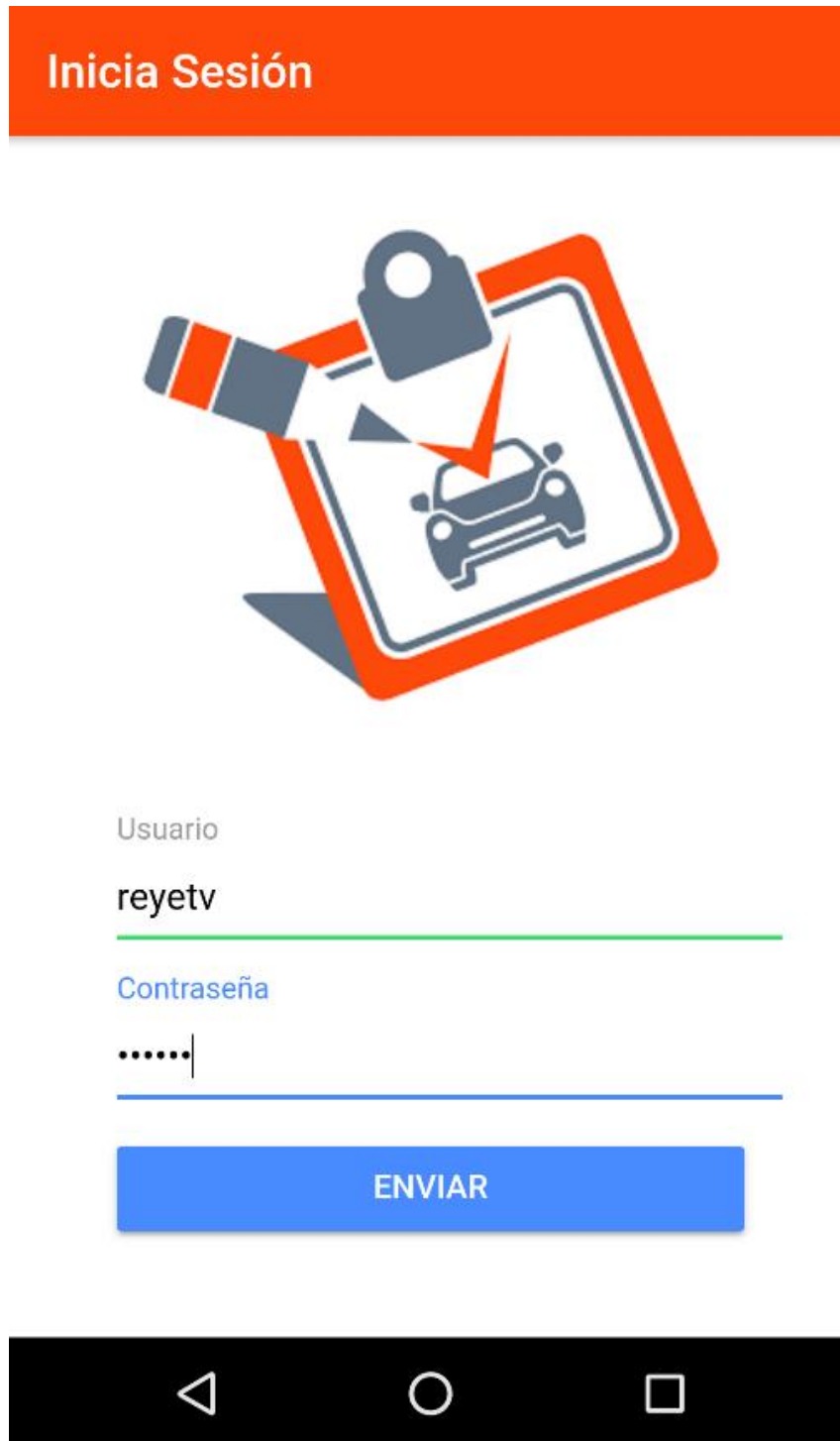
Ilustración 4.4: Módulos involucrados en el sistema.

4.7.1 Descripción de los módulos del sistema

- **Autenticación:** En este módulo, los usuarios realizan el proceso de autenticación con sus credenciales para entrar en el sistema.
- **Perfil:** Mediante este módulo, los usuarios pueden consultar su información personal.
- **Registro de solicitud:** En este módulo, el usuario registra en el formulario de la solicitud de inspección los valores correspondientes y éste es enviado al servidor con la base de datos central.
- **Almacenamiento local:** Por medio de este módulo, se almacena en la base de datos local del dispositivo todos los valores del formulario para que esté disponible sin conexión.

4.7.2 Descripción de las interfaces del sistema

➤ **Inicio de sesión:** Esta interfaz le permite al usuario ingresar en la aplicación móvil con el usuario y clave que posee en la base de datos central. (Ver ilustración 4.5).



Inicia Sesión

Usuario
reyetv

Contraseña
.....

ENVIAR

Ilustración 4.5: Interfaz de inicio de sesión.

➤ **Bandeja de solicitudes de inspección pendientes:** Esta interfaz contiene la lista de solicitudes de inspección con un estado de pendiente por atender. Desde ella se puede navegar a la bandeja de solicitudes de inspección en gestión, también es accesible el menú lateral y se puede actualizar la lista de la bandeja tocando el botón de refrescar. Al seleccionar una, la aplicación navegará a la página del formulario. (Ver ilustración 4.6).

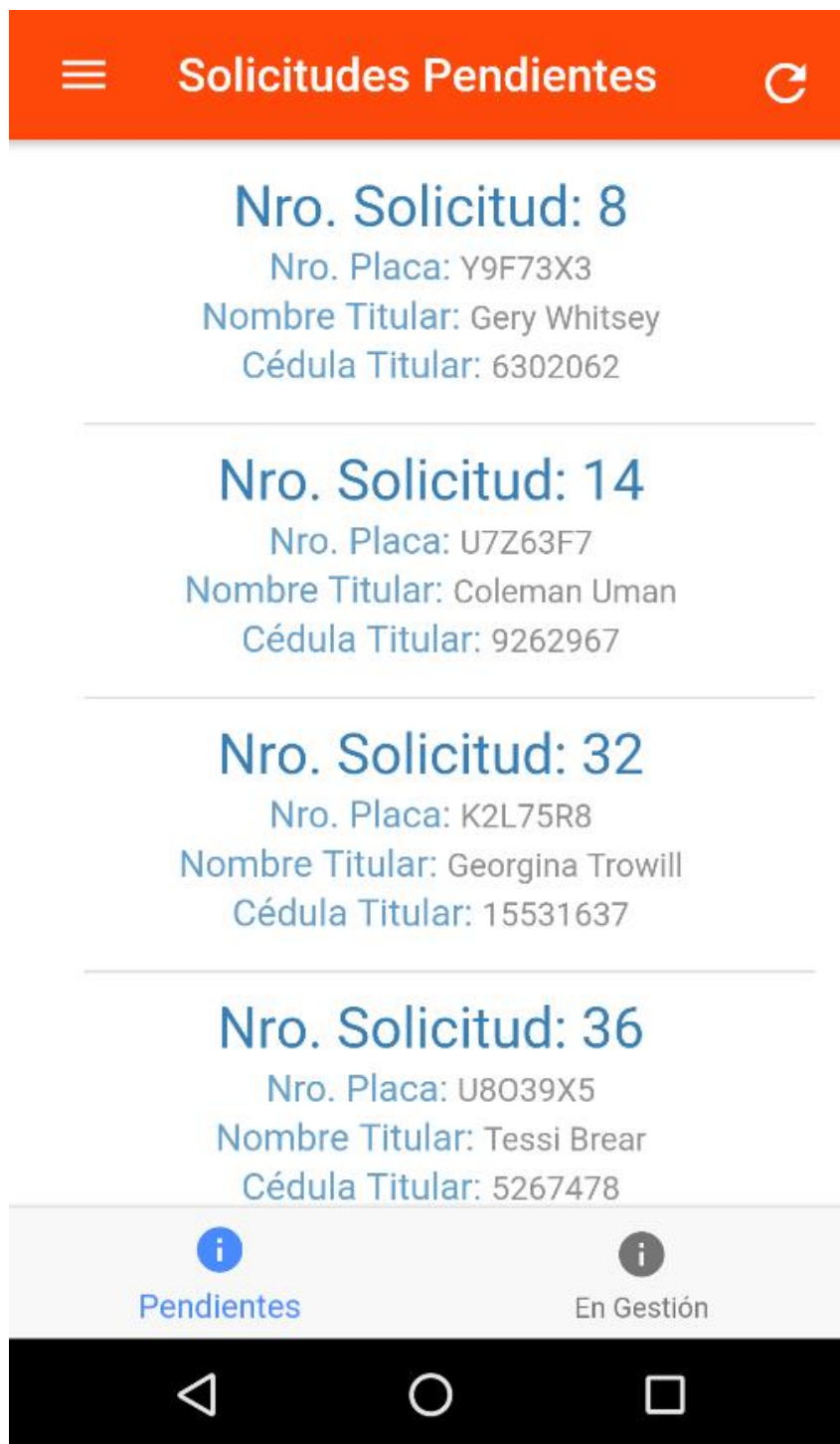


Ilustración 4.6: Interfaz de bandeja de solicitudes pendientes.

➤ **Bandeja de solicitudes de inspección en gestión:** Esta interfaz tiene la lista de solicitudes de inspección en gestión, las cuales abarcan las solicitudes en estado abierto y las solicitudes en estado finalizado. Las solicitudes en estado abierto sólo han sido asignadas a un inspector, pero no han sido completadas y por lo tanto no pueden ser enviadas al servidor. Las solicitudes en estado finalizado ya han sido completadas, pero pueden seguir siendo editadas y tendrán una opción extra para realizar el envío al servidor; luego de haber sido enviada, una solicitud no puede ser editada. (Ver ilustración 4.7).

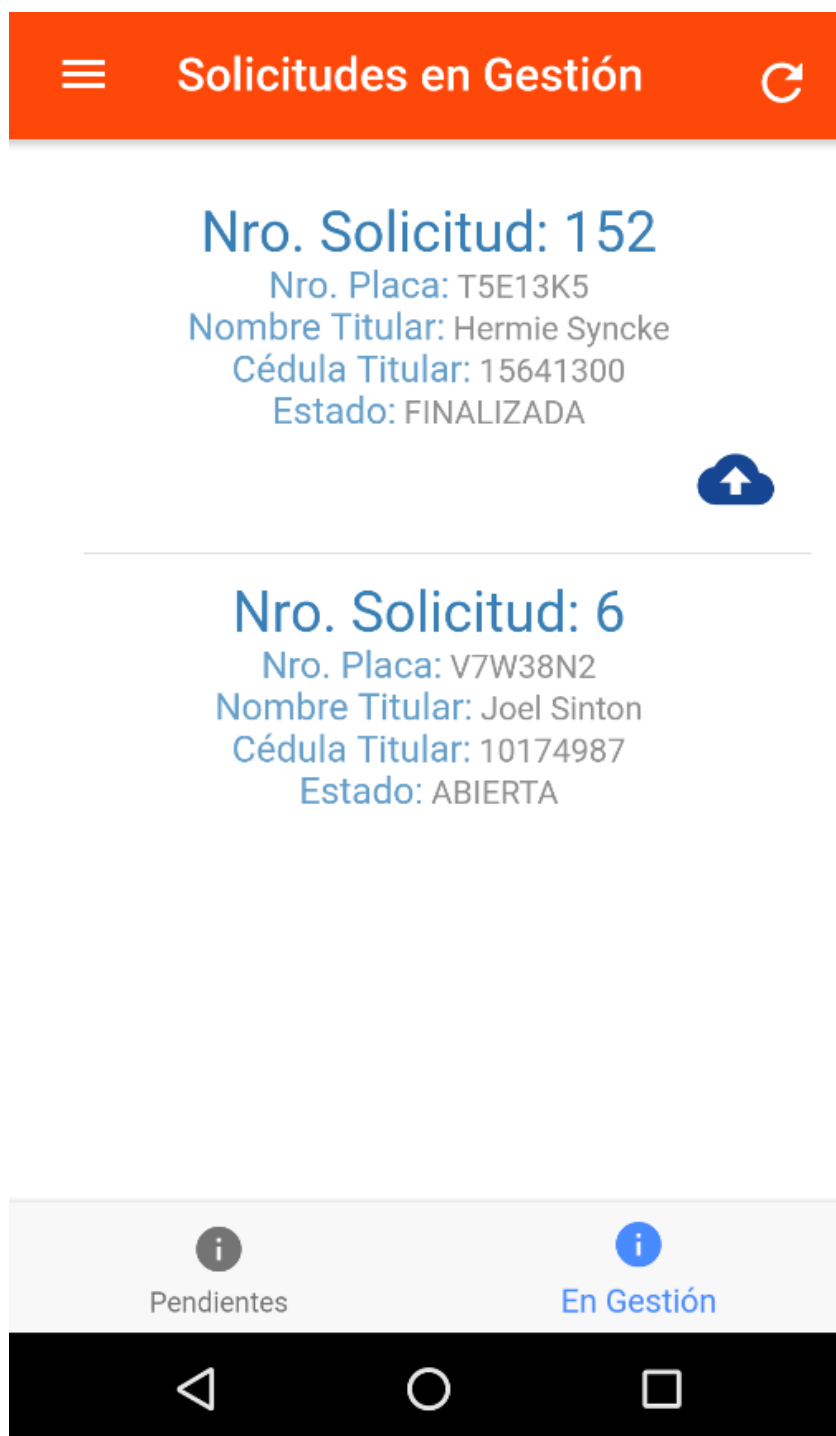


Ilustración 4.7: Interfaz de bandeja de solicitudes en gestión.

➤ **Menú lateral:** Esta interfaz es un menú lateral que aparece al arrastrar el dedo desde la parte izquierda de la pantalla hacia la derecha y desde él se puede acceder a otras interfaces, además de realizar el cierre de sesión actual. El menú no está habilitado hasta que se haya iniciado una sesión activa. (Ver ilustración 4.8).

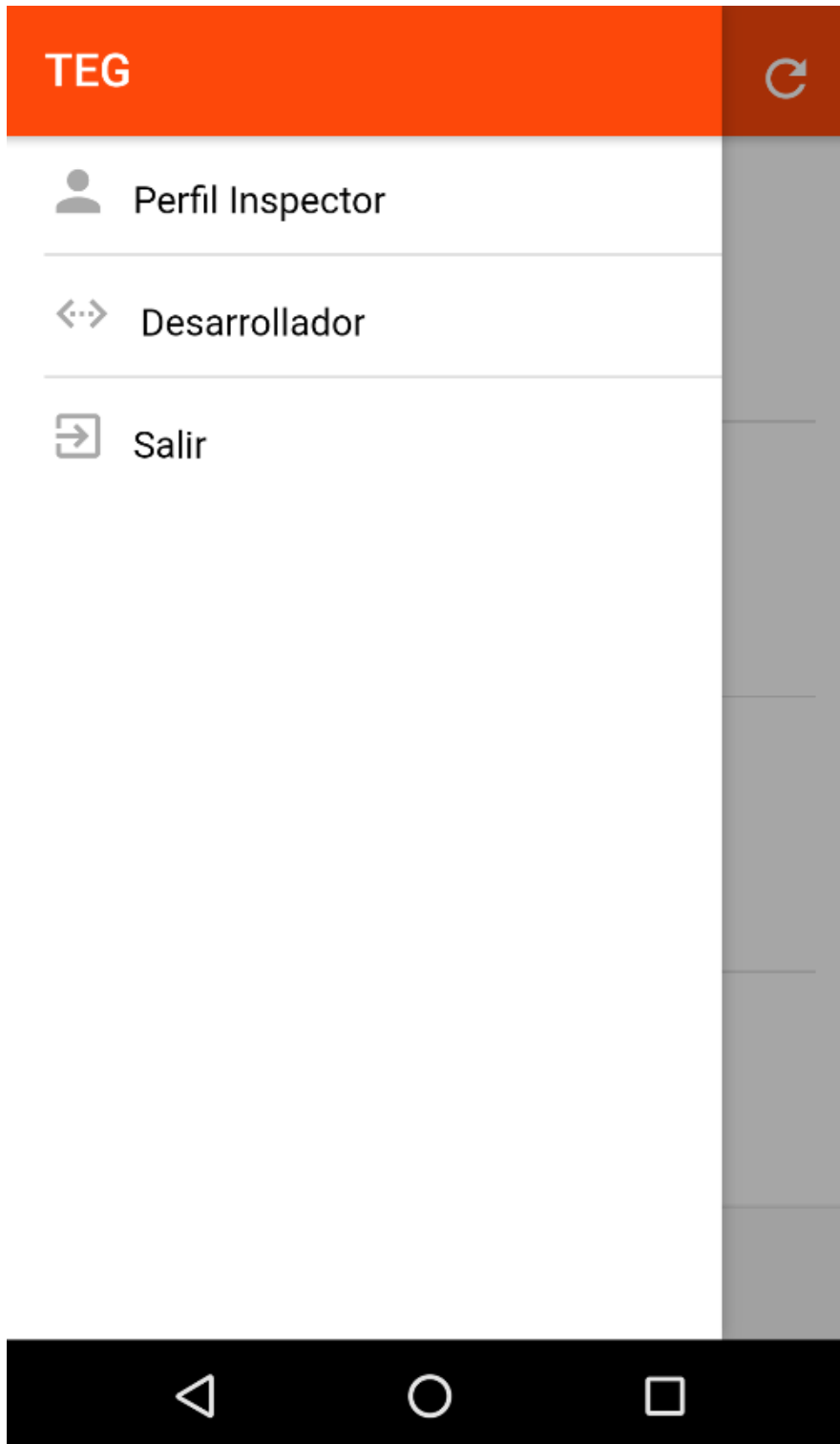


Ilustración 4.8: Interfaz de menú lateral.

➤ **Perfil de usuario inspector:** En esta interfaz el usuario inspector con la sesión activa puede consultar sus datos personales con la información provista de la base de datos central. (Ver ilustración 4.9).



Ilustración 4.9: Interfaz de perfil de usuario inspector.

➤ **Información básica del desarrollador:** En esta interfaz se muestra información básica sobre el desarrollador de la aplicación, además de las tecnologías usadas para realizar la aplicación. (Ver ilustración 4.10).

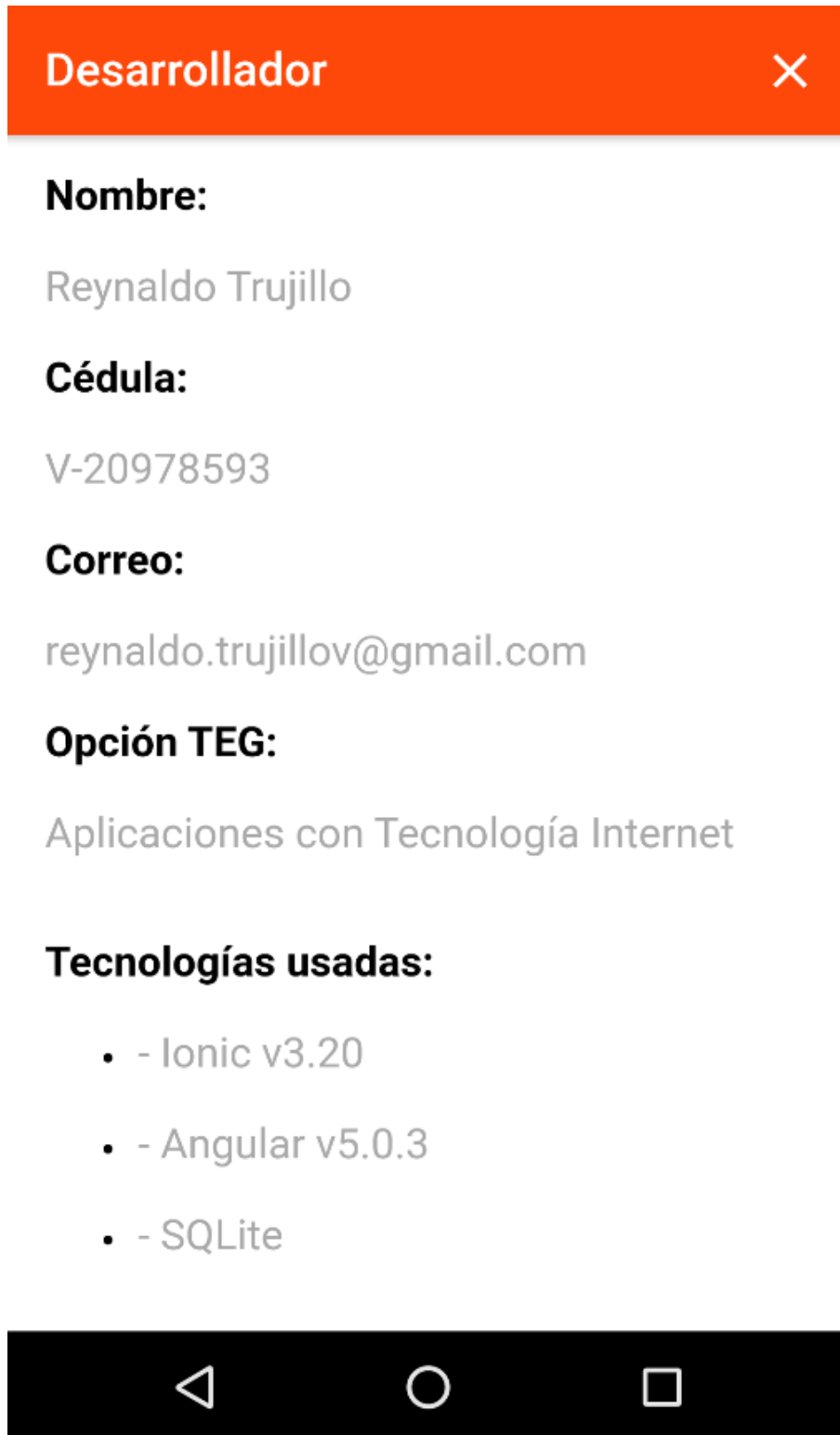


Ilustración 4.10: Interfaz de información básica del desarrollador.

➤ **Formulario de registro:** Esta interfaz se divide en muchas otras interfaces de registro debido a la gran cantidad de campos del formulario. Se listan a continuación.

▪ **Página de inicio de inspección:** Esta es la primera página del formulario, la cual posee los datos del titular provenientes de la base de datos central y de donde el inspector puede decidir si atender la solicitud o no. (Ver ilustración 4.11).

Datos del Vehículo

Datos Titular del Vehículo

Nombre
Joel

Apellido
Sinton

Cédula de Identidad
10174987

Teléfono
118-758-1622

CONTINUAR INSPECCIÓN

VOLVER

● ● ● ● ● ● ● ● ● ●

◀ ○ ◻

Ilustración 4.11: Interfaz de página de inicio de inspección.

▪ **Página de datos del vehículo:** En esta interfaz del formulario, se solicitan datos básicos del vehículo en inspección. (Ver ilustración 4.12).

Datos del Vehículo		Datos del Vehículo		Datos del Vehículo	
Tipo Vehículo	Sedán	Año	2008	Color	Negro
Transmisión	Sincronico	Capacidad de Puestos	5	Kilometraje	70000
Nro. Placa	N5U9215	Cilindros	4	Serial de Carrocería	SLPQ201
Marca	Chevrolet	Peso (kg.)	250	Serial del Motor	GBWQ19
Modelo	Optra	Color	Negro	Valor Estimado	2500
Año	2008	Kilometraje	70000		
Capacidad de Puestos		Serial de Carrocería	SLPQ201		

Ilustración 4.12: Interfaz de página de datos del vehículo en inspección.

▪ **Página de datos de las condiciones generales del vehículo:** En esta interfaz del formulario, se presenta un conjunto de aspectos sobre las condiciones generales del vehículo en inspección y de los que se debe saber su estado actual con opción a colocar observaciones. (Ver ilustración 4.13).

Datos del Vehículo

Condiciones Generales del Vehículo

23. Latonería

Bueno

Regular

Malo

Excelente acabado.

24. Parrilla o Rejilla Central

Ilustración 4.13: Interfaz de página de las condiciones generales del vehículo en inspección.

▪ **Página de datos de la mecánica del vehículo:** En esta interfaz del formulario, se presenta un conjunto de aspectos sobre las mecánicas del vehículo en inspección y de los que se debe saber su estado actual con opción a colocar observaciones. (Ver ilustración 4.14).

The screenshot shows a mobile application interface for vehicle inspection. At the top, there is an orange header bar with the text "Datos del Vehículo" and a camera icon. Below the header, the main title "Mecánica del Vehículo" is displayed. The first section is titled "56. Encendido" in blue. It contains three radio button options: "Bueno" (unselected), "Regular" (selected, highlighted in blue), and "Malo" (unselected). Below these options is a text input field containing the text "Tarda en encender.". The second section is titled "57. Embrague o Crocht" in blue. Below the sections, there is a horizontal row of ten small circles, with the fifth circle from the left being blue and the others grey. At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

Ilustración 4.14: Interfaz de la página de las mecánicas del vehículo en inspección.

▪ **Página de datos de los accesorios del vehículo:** En esta interfaz del formulario, se presenta un conjunto de aspectos sobre los accesorios del vehículo en inspección y de los que se debe saber su estado actual con opción a colocar observaciones. (Ver ilustración 4.15).

The screenshot displays a mobile application interface for vehicle inspection. At the top, there is an orange header bar with the text "Datos del Vehículo" and a camera icon. Below the header, the main content area is titled "Accesorios del Vehículo". It features two sections for data entry:

- 62. Radio / Antena:** A section with a title in blue. It contains a radio button interface with two options: "SI" (selected, indicated by a blue dot) and "NO". Below this is a text input field containing the text "Antena de radio común."
- 63. Foco de Posición:** A section with a title in blue. It contains a radio button interface with two options: "SI" (selected, indicated by a blue dot) and "NO".

At the bottom of the form, there is a horizontal row of ten small grey circles, with the seventh circle from the left being filled with blue, indicating the current page. Below the form is a black navigation bar with three white icons: a back arrow, a home circle, and a square app drawer icon.

Ilustración 4.15: Interfaz de la página de datos de los accesorios del vehículo en inspección.

▪ **Página de datos de detalles adicionales en piezas del vehículo:** En esta interfaz del formulario, se presenta un conjunto de aspectos sobre los detalles adicionales en piezas del vehículo en inspección y de los que se debe saber su estado actual con un valor estimado para solventar la falla o detalle que pueda tener. (Ver ilustración 4.16).

The screenshot shows a mobile application interface for vehicle inspection. At the top, there is an orange header bar with the text "Datos del Vehículo" and a camera icon. Below this is a section titled "Detalles de Datos" which contains three data entry fields:

- 92. Pieza:** A text input field containing "Radiador".
- 93. Tipo de Daño:** A text input field containing "Alto".
- 94. Costo Aproximado:** A text input field containing "4500".

Below the data entry fields, there is a green button labeled "+ AGREGAR DETALLE". Underneath that is a blue button labeled "ANTERIOR". At the bottom of the form area, there is a horizontal row of ten small circles, with the eighth circle from the left being filled with blue, indicating the current step. The entire interface is set against a white background with a black Android-style navigation bar at the very bottom.

Ilustración 4.16: Interfaz de página de detalles adicionales del vehículo en inspección.

▪ **Página de datos de los documentos del titular:** En esta última interfaz del formulario, se presenta un conjunto de aspectos sobre los documentos del titular del vehículo en inspección. (Ver ilustración 4.17).

Datos del Vehículo

Documentos Presentados para Inspección

95. Registro Automotor Pe...

SI

NO

96. Carnet de Circulación

SI

NO

Ilustración 4.17: Interfaz de página de los documentos del titular del vehículo en inspección.

▪ **Pantalla de archivos multimedia:** En esta interfaz se presentan los archivos multimedia asociados a la solicitud de inspección, permite tomar hasta seis (6) fotografías para anexar al formulario. (Ver ilustración 4.18).

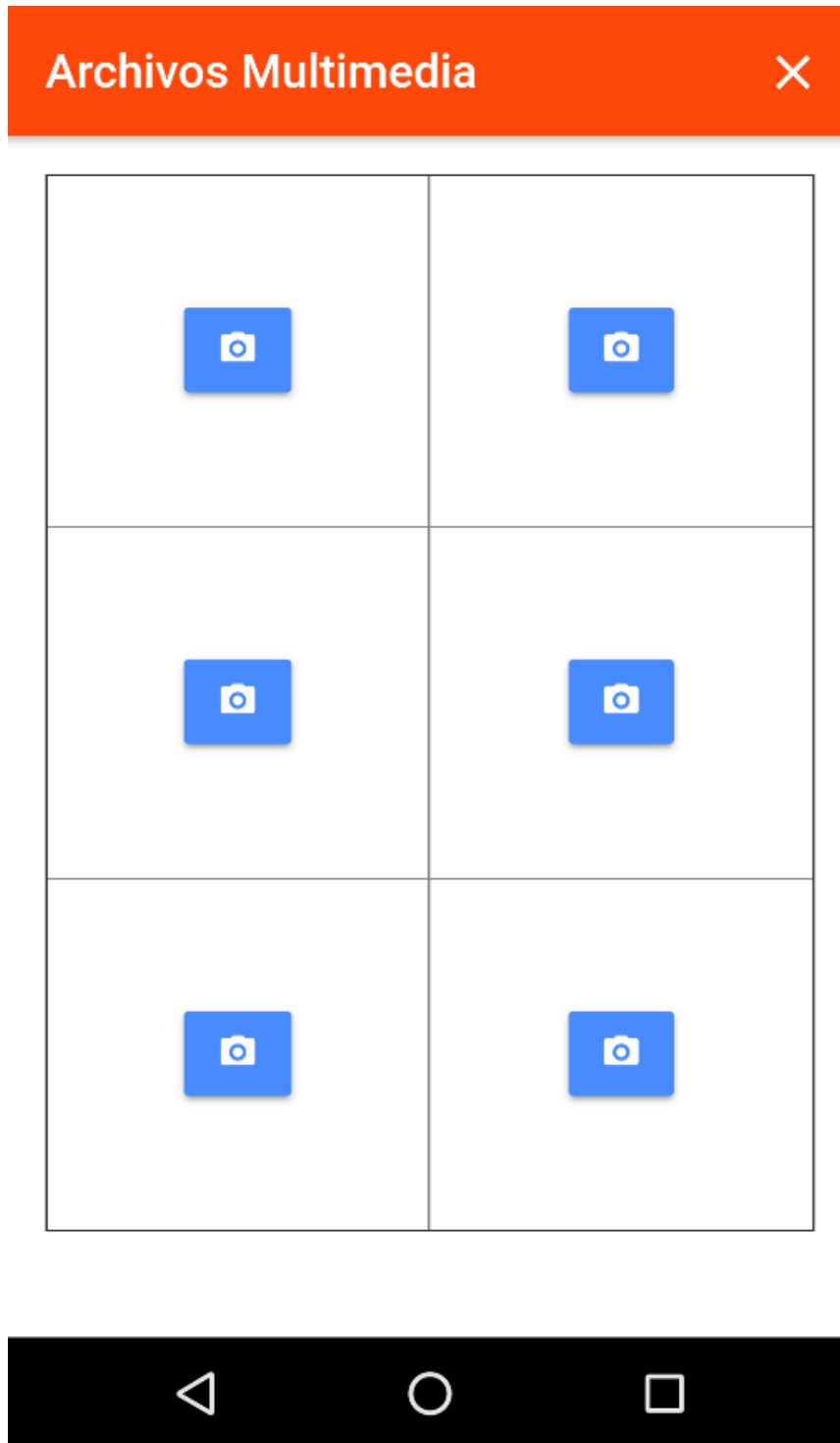


Ilustración 4.18: Interfaz de la pantalla de archivos multimedia de la inspección.

➤ **Confirmación de envío de solicitud a base de datos local:** Esta interfaz es un modal de confirmación de envío de solicitud a la base de datos local que el usuario puede aceptar o no. (Ver ilustración 4.19).

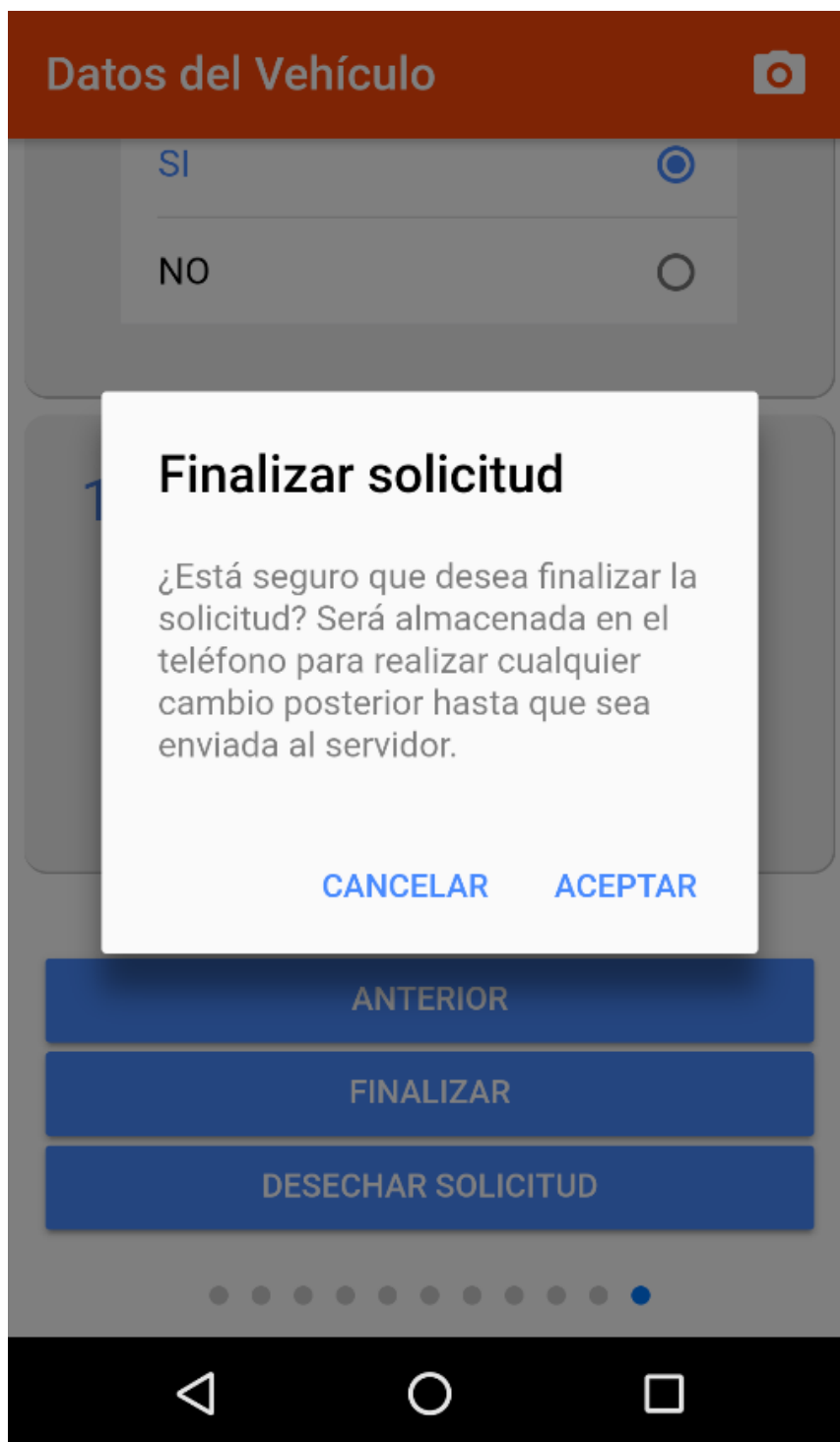


Ilustración 4.19: Interfaz de alerta de confirmación de envío de solicitud a la base de datos local.

➤ **Confirmación de asignación de solicitud a inspector:** Esta interfaz es un modal de confirmación donde el inspector puede decir si desea atender la solicitud en cuestión o no. (Ver ilustración 4.20).

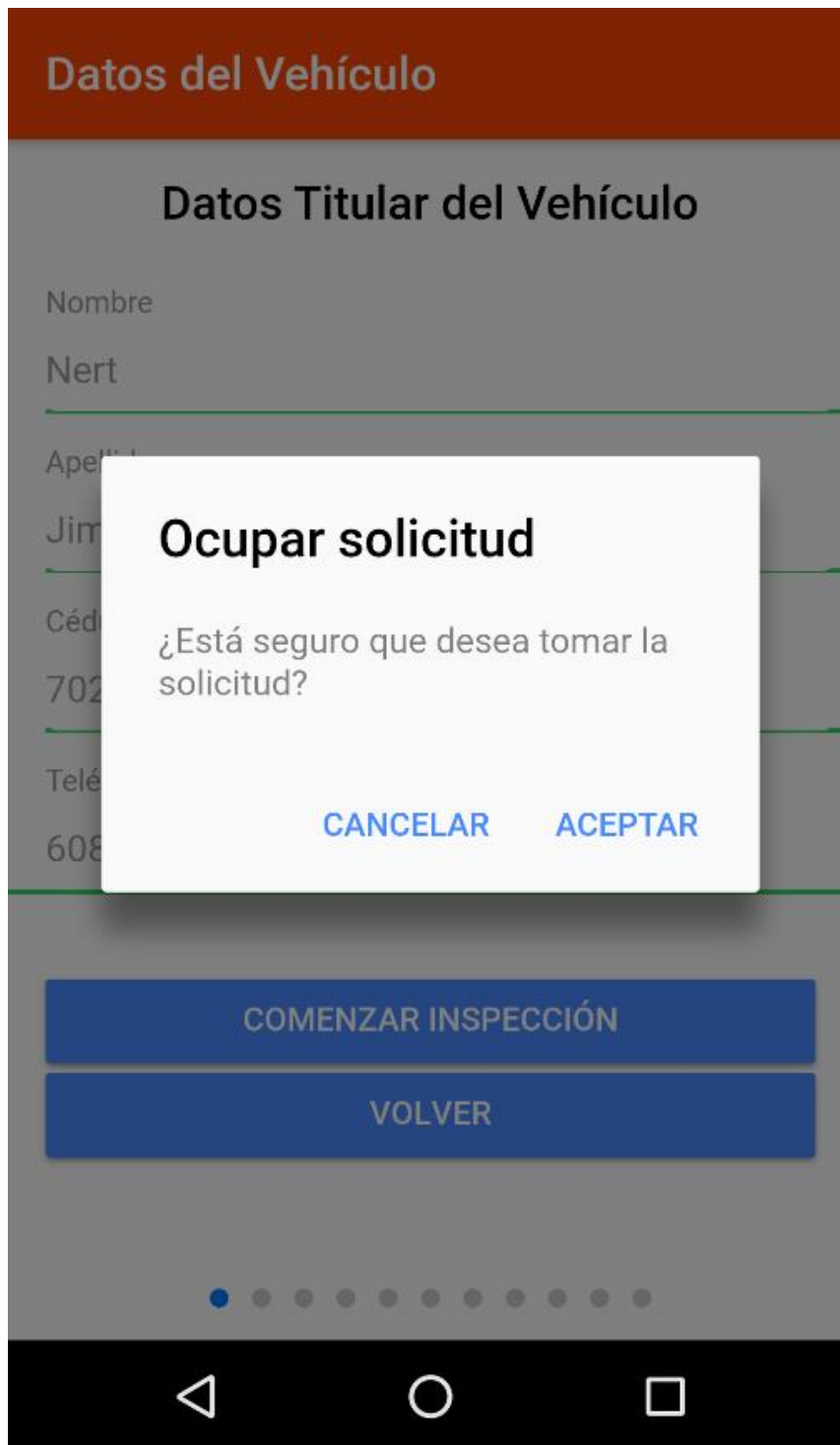


Ilustración 4.20: Interfaz de la alerta de confirmación de asignación de solicitud a inspector.

➤ **Confirmación de envío de solicitud al servidor:** Esta interfaz es un modal de confirmación de envío de solicitud a la base de datos central que el usuario puede aceptar o no. (Ver ilustración 4.21).

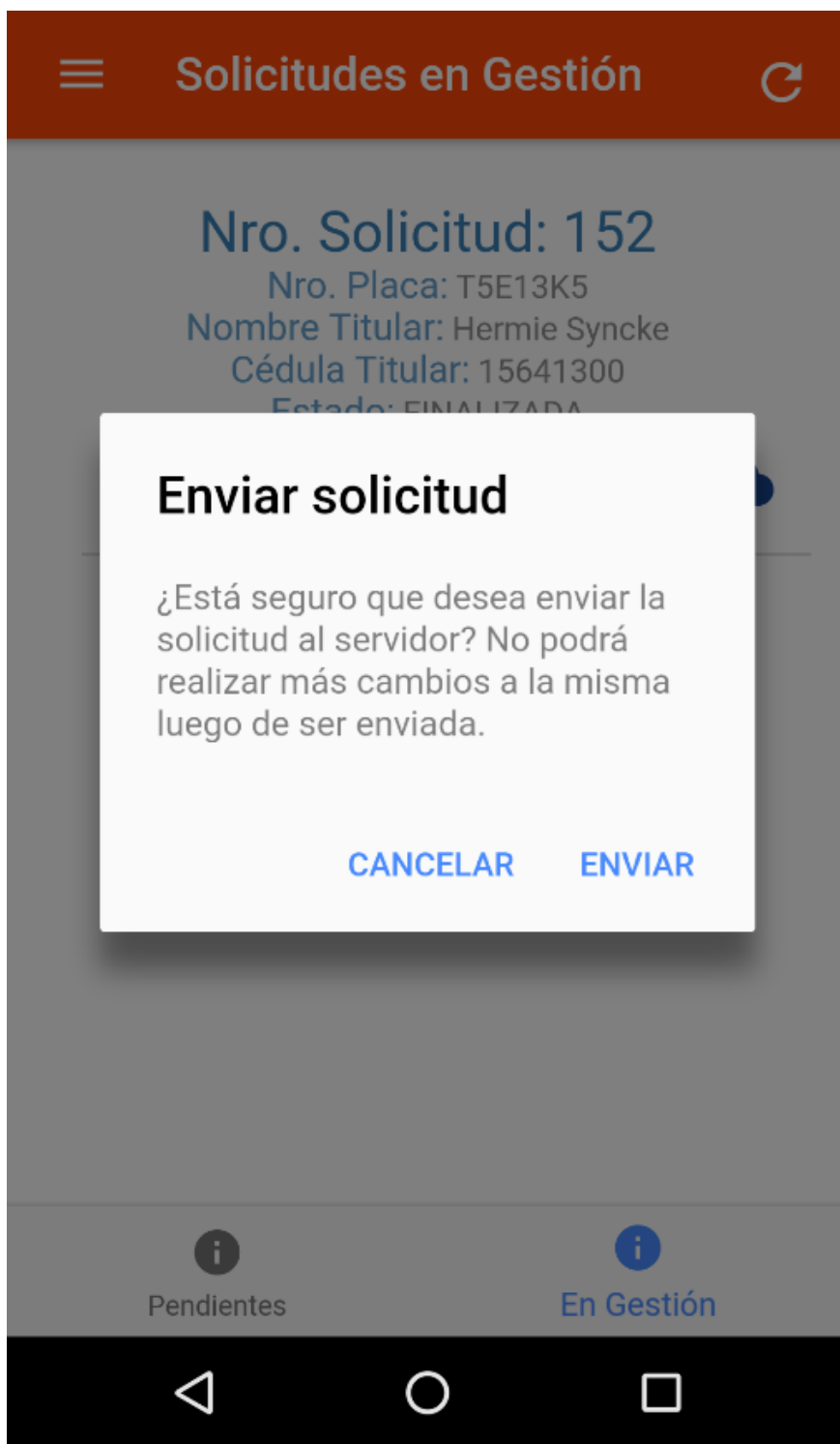


Ilustración 4.21: Interfaz de la alerta de confirmación de envío de solicitud al servidor.

➤ **Confirmación para salir de la aplicación:** Esta interfaz es un modal para alertar al usuario en caso de que quiera salir de la aplicación o no, previniendo cierres accidentales. (Ver ilustración 4.22).

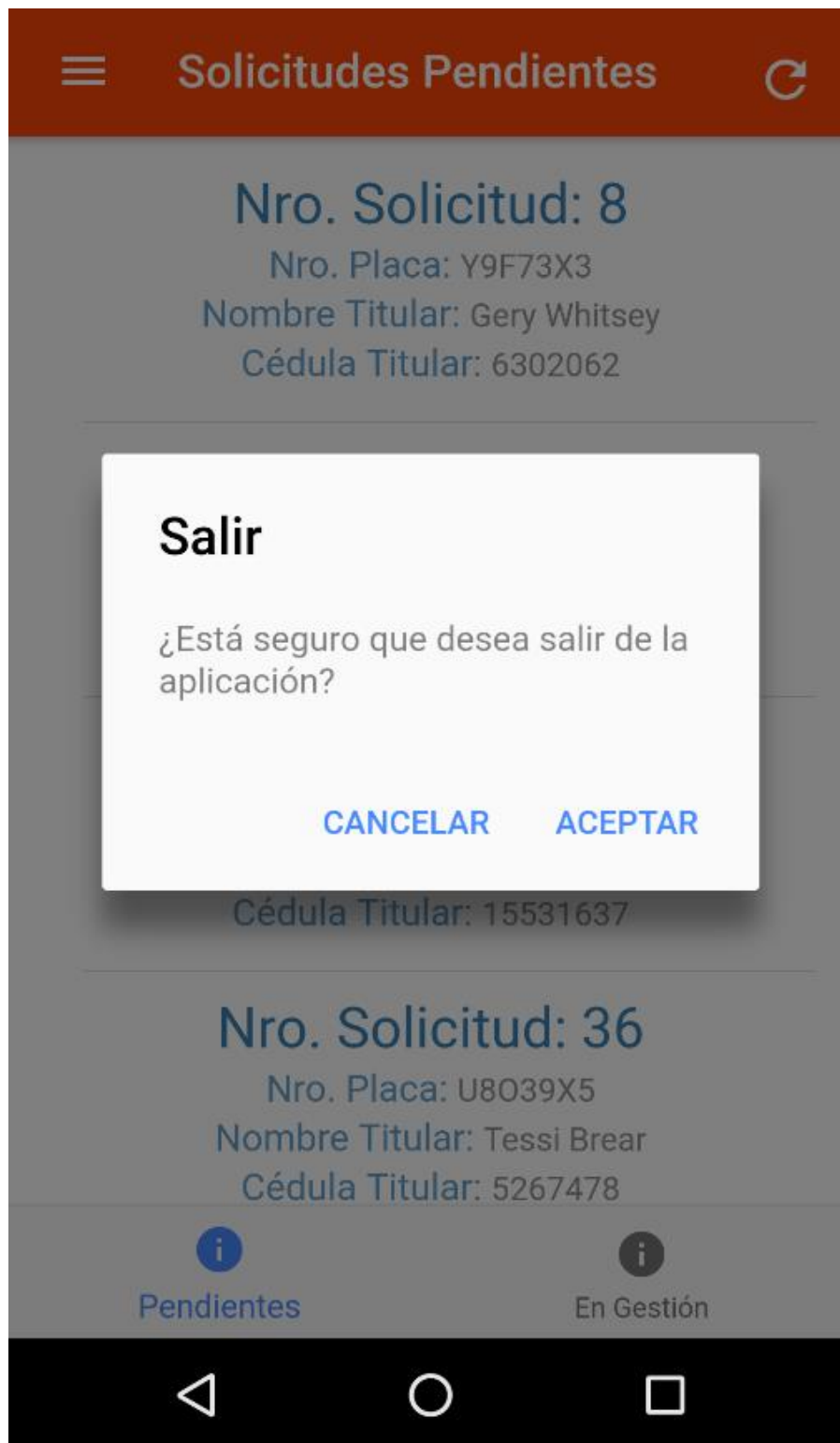


Ilustración 4.22: Interfaz de la alerta de confirmación para salir de la aplicación.

➤ **Confirmación para salir del formulario durante inspección:** Esta interfaz es un modal para alertar al usuario en caso de que quiera salir de la inspección o no, previniendo cierres accidentales teniendo como aviso que los cambios no guardados en la base de datos local se perderán. (Ver ilustración 4.23).

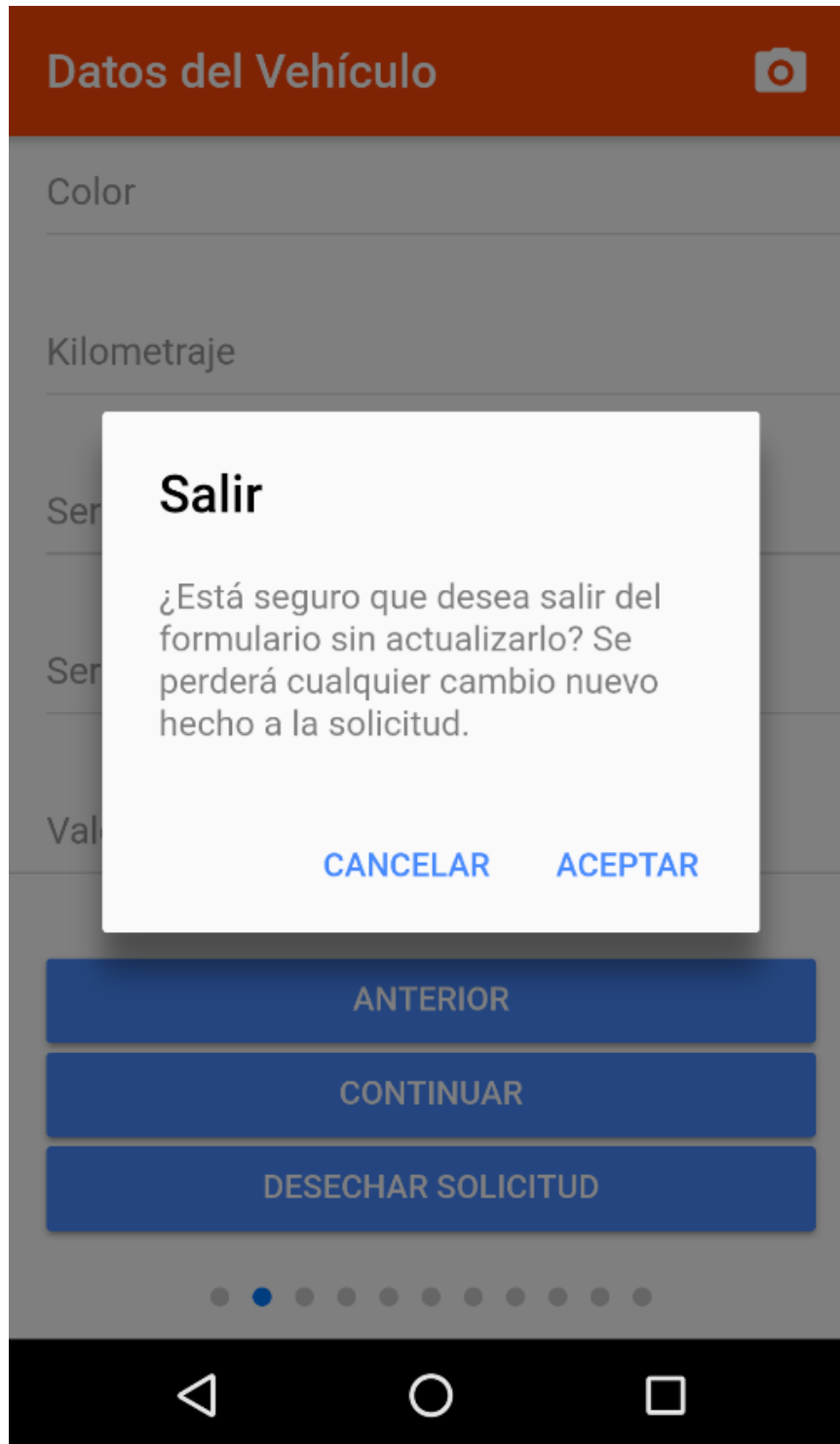


Ilustración 4.23: Interfaz de alerta de confirmación para salir del formulario durante inspección.

4.8 Fase de pruebas

Se realizó un conjunto de pruebas para comprobar el correcto funcionamiento del sistema, además de la aceptación o receptividad de los usuarios con la aplicación. Para ello, se llevaron a cabo dos tipos de pruebas. Seguidamente, una explicación de cada tipo.

4.8.1 Pruebas funcionales

Uno de los componentes más importantes en la metodología SCRUM es el de las pruebas funcionales donde cada módulo del sistema y sus funcionalidades son puestas a prueba.

Módulo	Funcionalidades probadas
Perfil	➤ Consulta de usuario con sesión activa.
Autenticación	➤ Envío de datos al servidor. ➤ Validación de datos de entrada.
Registro de solicitud	➤ Creación de formulario de solicitud. ➤ Validación de datos de entrada. ➤ Toma de fotografías para el formulario. ➤ Envío de datos al servidor.
Almacenamiento local	➤ Creación de esquema de base de datos móvil. ➤ Creación de métodos para gestionar la base de datos móvil.

Tabla 4.3: Pruebas funcionales por módulo.

Las pruebas presentadas en la tabla 4.3 se ejecutaron de manera cíclica hasta lograr una iteración con la aprobación de todos los módulos, dicho proceso es expuesto en la tabla 4.4 a continuación.

Usuario	Iteración de prueba			
	1.	2.	3.	4.
1.	- 2 funcionalidades aprobadas. - 7 funcionalidades no aprobadas.	- 5 funcionalidades aprobadas. - 4 funcionalidades no aprobadas.	- 7 funcionalidades aprobadas. - 2 funcionalidades no aprobadas.	- 9 funcionalidades no aprobadas.
2.	- 2 funcionalidades aprobadas. - 7 funcionalidades no aprobadas.	- 4 funcionalidades aprobadas. - 5 funcionalidades no aprobadas.	- 6 funcionalidades aprobadas. - 3 funcionalidades no aprobadas.	- 9 funcionalidades no aprobadas.
3.	- 2 funcionalidades aprobadas. - 7 funcionalidades no aprobadas.	- 6 funcionalidades aprobadas. - 3 funcionalidades no aprobadas.	- 7 funcionalidades aprobadas. - 2 funcionalidades no aprobadas.	- 9 funcionalidades aprobadas.

Tabla 4.4: Matriz de ciclos de prueba.

➤ **Ciclo 1**

○ **Usuario inspector 1:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas sólo dos (2) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Validación de datos de entrada en la autenticación, acepta cadenas vacías.
- El formulario está incompleto.
- Validación de datos de entrada del formulario.
- Toma fotografías, pero no se pueden acceder de nuevo por falta de identificador.
- No se está construyendo el objeto que se envía al servidor en el formulario.
- La base de datos móvil no aceptó restricciones de integridad.
- Los métodos de gestión de la base de datos móvil no se ejecutaron.

○ **Usuario inspector 2:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas sólo dos (2) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Validación de datos de entrada en la autenticación, acepta cadenas vacías.
- El formulario está incompleto.
- Validación de datos de entrada del formulario.
- Toma fotografías, pero no se pueden acceder de nuevo por falta de identificador.
- No se está construyendo el objeto que se envía al servidor en el formulario.

- La base de datos móvil no aceptó restricciones de integridad.
- Los métodos de gestión de la base de datos móvil no se ejecutaron.

○ **Usuario analista de calidad funcional:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas sólo dos (2) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Validación de datos de entrada en la autenticación, acepta cadenas vacías.
- El formulario está incompleto.
- Validación de datos de entrada del formulario.
- Toma fotografías, pero no se pueden acceder de nuevo por falta de identificador.
- No se está construyendo el objeto que se envía al servidor en el formulario.
- La base de datos móvil no aceptó restricciones de integridad.
- Los métodos de gestión de la base de datos móvil no se ejecutaron.

➤ **Ciclo 2**

○ **Usuario inspector 1:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas cinco (5) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Validación de datos de entrada del formulario.
- Al momento de ver las fotos, la aplicación se detiene.
- No se está cambiando el estado de la solicitud a “FINALIZADA”.
- Los métodos de inserción ejecutaron en un orden incorrecto por lo que arrojaron errores sin excepciones manejadas.

○ **Usuario inspector 2:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas cuatro (4) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Validación de datos de entrada del formulario.
- No hay mensajes al usuario que indiquen si el sistema está trabajando.
- Al momento de ver las fotos, la aplicación se detiene.
- No se está cambiando el estado de la solicitud a “FINALIZADA”.
- Los métodos de inserción ejecutaron en un orden incorrecto por lo que arrojaron errores sin excepciones manejadas.

○ **Usuario analista de calidad funcional:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas seis (6) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Al momento de ver las fotos, la aplicación se detiene.
- No se está cambiando el estado de la solicitud a “FINALIZADA”.
- Los métodos de inserción ejecutaron en un orden incorrecto por lo que arrojaron errores sin excepciones manejadas.

➤ **Ciclo 3**

○ **Usuario inspector 1:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas siete (7) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Las fotografías no se envían de forma correcta al servidor.
- Al querer actualizar un formulario en gestión, su información puede venir errónea de la base de datos local.

○ **Usuario inspector 2:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas seis (6) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Las fotografías no se envían de forma correcta al servidor.
- Al querer actualizar un formulario en gestión, su información puede venir errónea de la base de datos local.
- El botón de atrás del teléfono cierra la aplicación.

○ **Usuario analista de calidad funcional:** se probaron todas las funcionalidades y de las nueve (9), fueron aprobadas siete (7) en esta primera iteración del ciclo de pruebas, siguen pendientes las siguientes funcionalidades:

- Las fotografías no se envían de forma correcta al servidor.
- Al querer actualizar un formulario en gestión, su información puede venir errónea de la base de datos local.

➤ **Ciclo 4**

Durante este último ciclo de pruebas, todos los usuarios aprobaron las funcionalidades del sistema en su totalidad de manera satisfactoria.

4.8.2 Pruebas de aceptación

Finalmente, se realizaron las pruebas de aceptación a usuarios del sistema para solicitudes de inspección. Estas pruebas se hicieron para comprobar la tolerancia a fallas del sistema, y para verificar que cumple adecuadamente con el proceso de solicitudes de inspección vehicular.

Para llevar a cabo estas pruebas, se construyeron un conjunto de instrumentos de evaluación presentados en la tabla 4.5, donde cada instrumento cuenta con cinco posibles respuestas a enunciados que realizan afirmaciones acerca de la aplicación.

		Totalmente de acuerdo	De acuerdo	Neutro	En desacuerdo	Totalmente en desacuerdo
1.	Las funciones del sistema son fáciles de usar					
2.	El lenguaje empleado en los mensajes de usuario es claro					
3.	La aplicación es intuitiva					
4.	El diseño y estilo de la aplicación es consistente					
5.	Las acciones realizadas son fáciles de memorizar					
6.	El sistema apoya al usuario en caso de errores					
7.	La navegación en la aplicación es sencilla					
8.	Los colores del sistema son placenteros a la vista					
9.	La aplicación se siente rápida y fluida					
10.	La aplicación facilita y acelera el proceso de inspección					

Tabla 4.5: Instrumento de evaluación para pruebas de aceptación.

4.8.2.1 Resultados de pruebas de aceptación

A continuación, se muestran los resultados de la aplicación del instrumento de evaluación que se aplicó a 10 usuarios finales distintos.

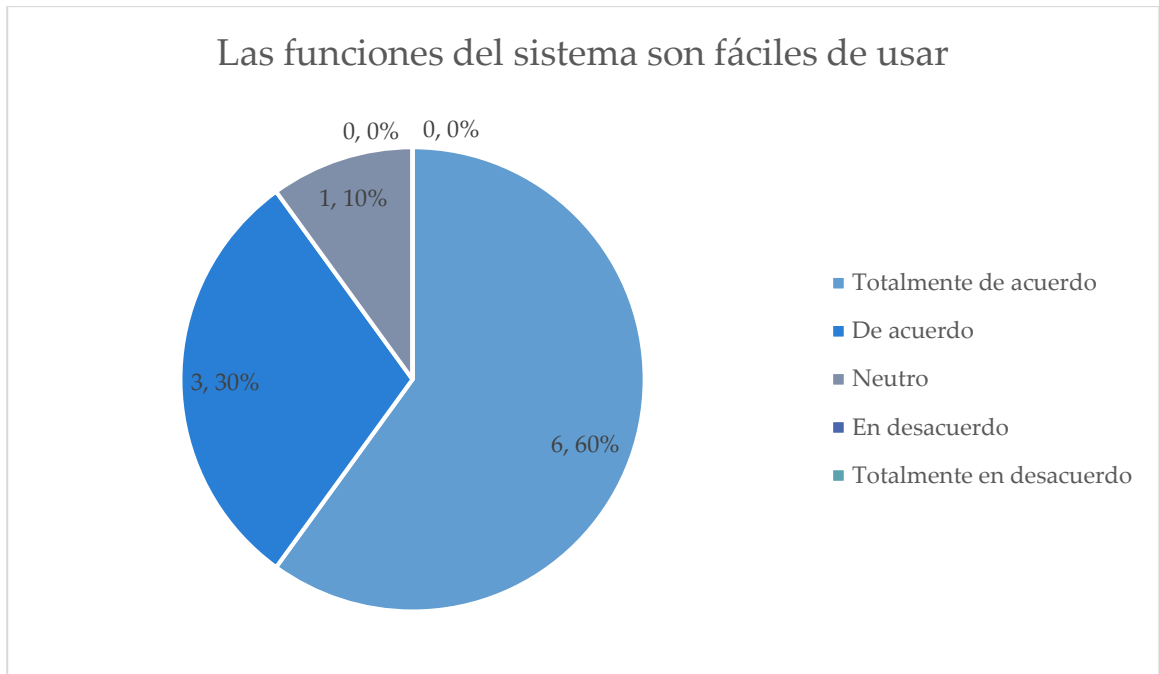


Ilustración 4.24: Resultados de pregunta 1 del instrumento de evaluación.

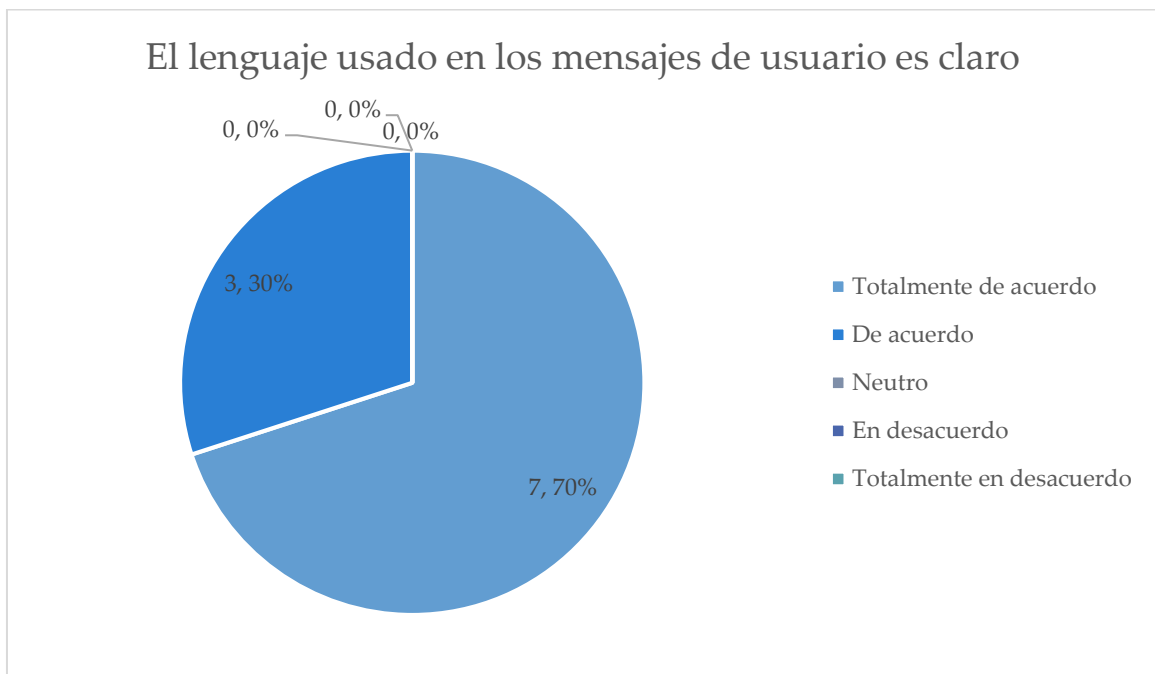


Ilustración 4.25: Resultados de pregunta 2 del instrumento de evaluación.



Ilustración 4.26: Resultados de pregunta 3 del instrumento de evaluación.

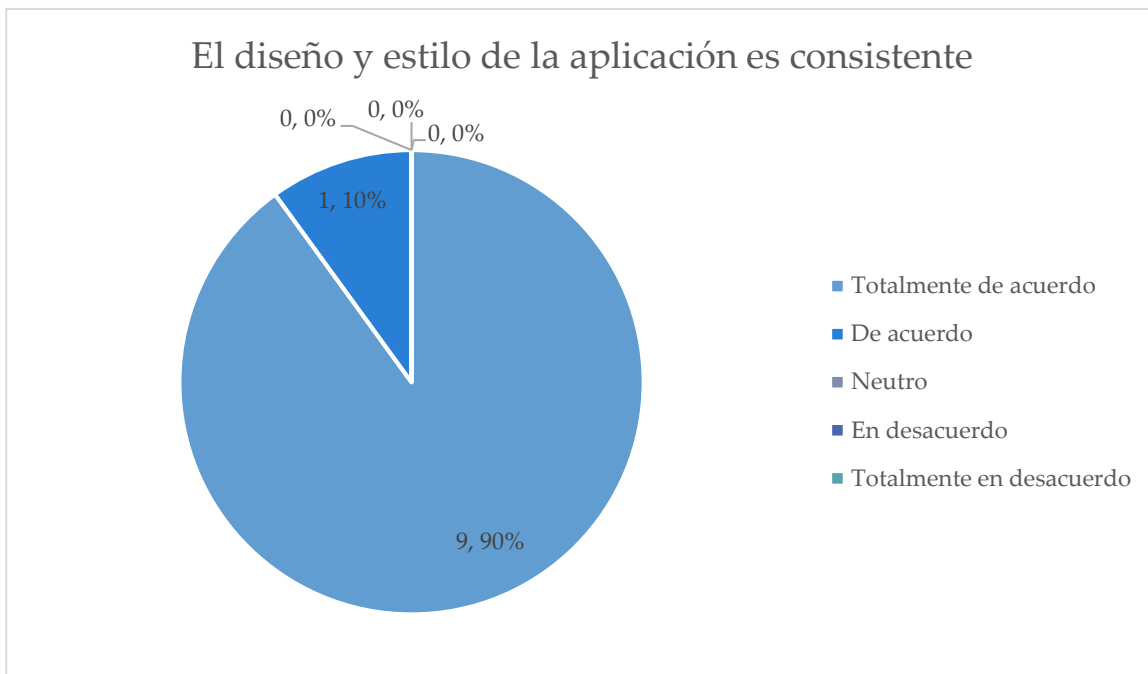


Ilustración 4.27: Resultados de pregunta 4 del instrumento de evaluación.

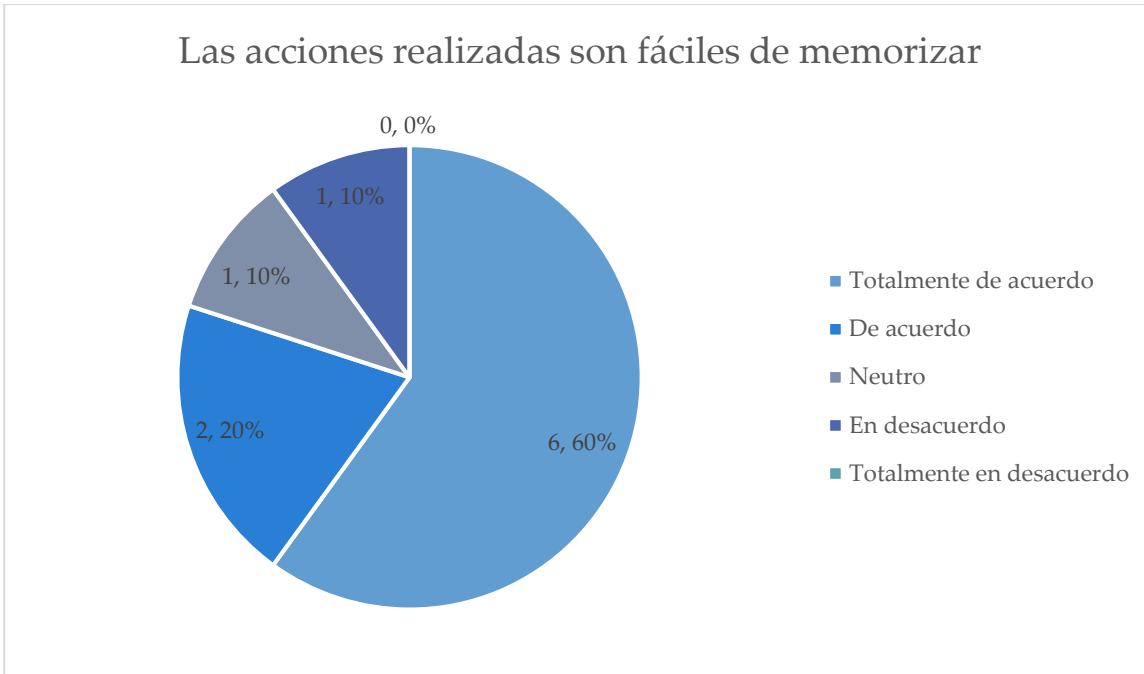


Ilustración 4.28: Resultados de pregunta 5 del instrumento de evaluación.

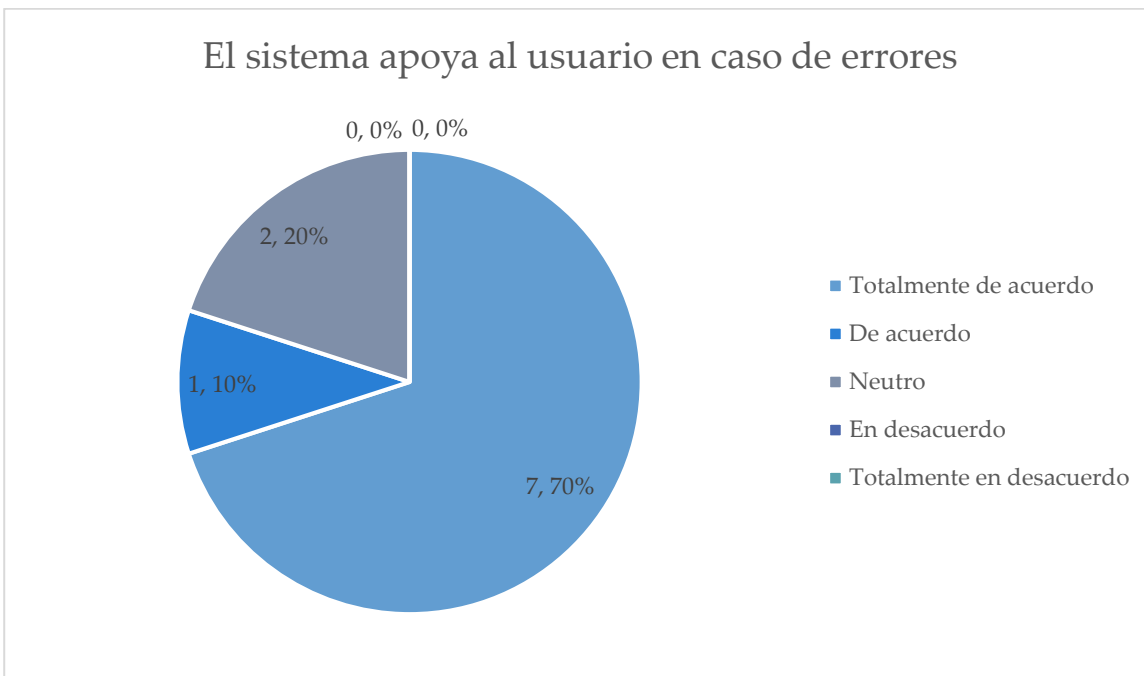


Ilustración 4.29: Resultados de pregunta 6 del instrumento de evaluación.

La navegación en la aplicación es sencilla

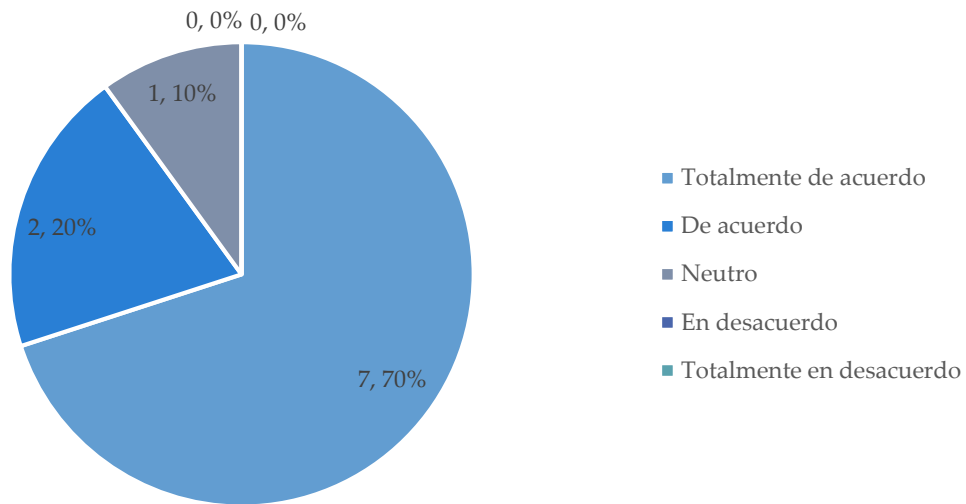


Ilustración 4.30: Resultados de pregunta 7 del instrumento de evaluación.

Los colores del sistema son placenteros a la vista

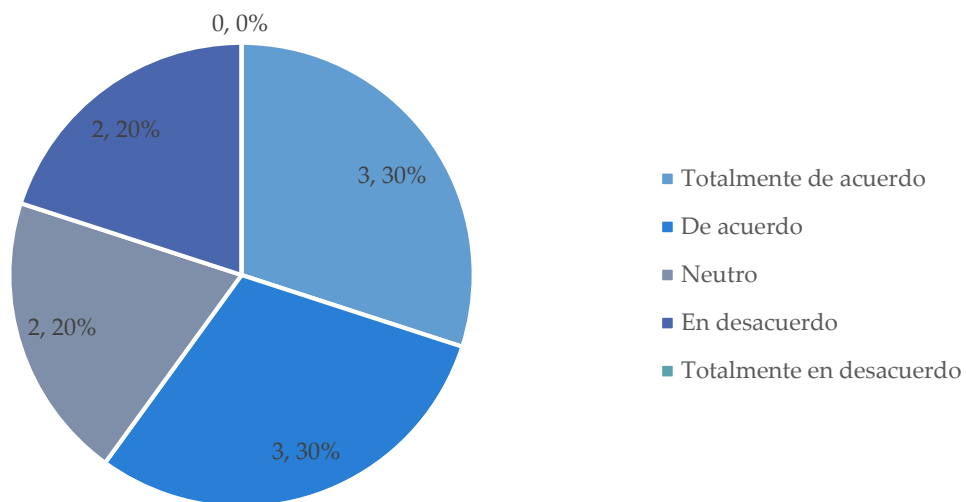


Ilustración 4.31: Resultados de pregunta 8 del instrumento de evaluación.

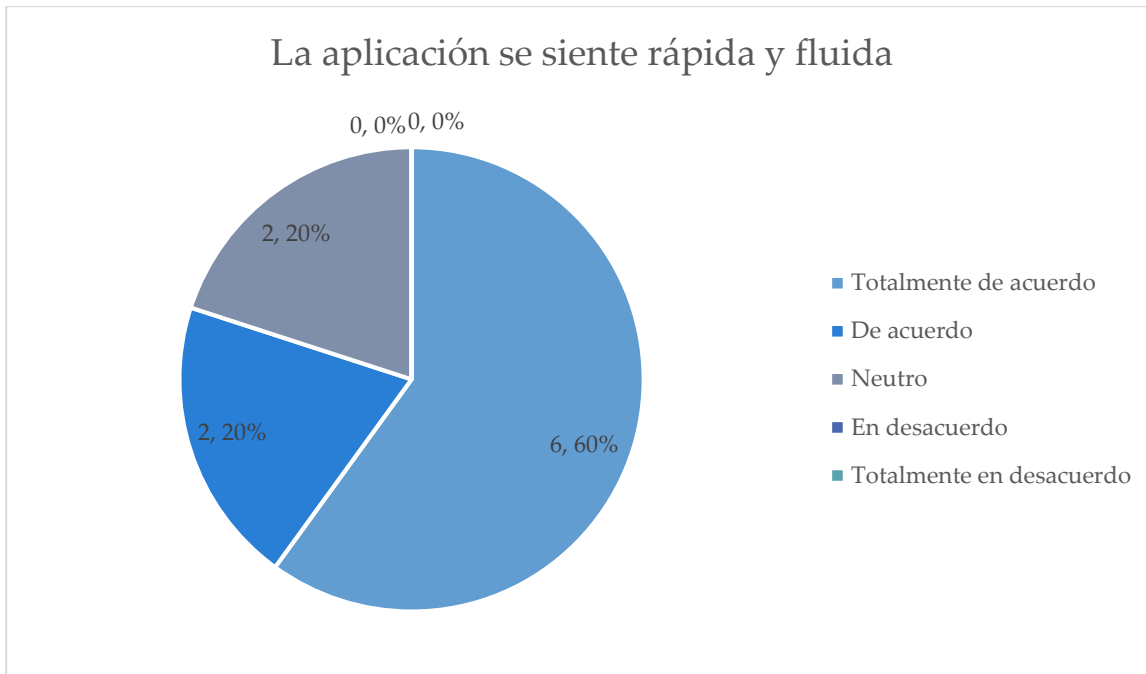


Ilustración 4.32: Resultados de pregunta 9 del instrumento de evaluación.

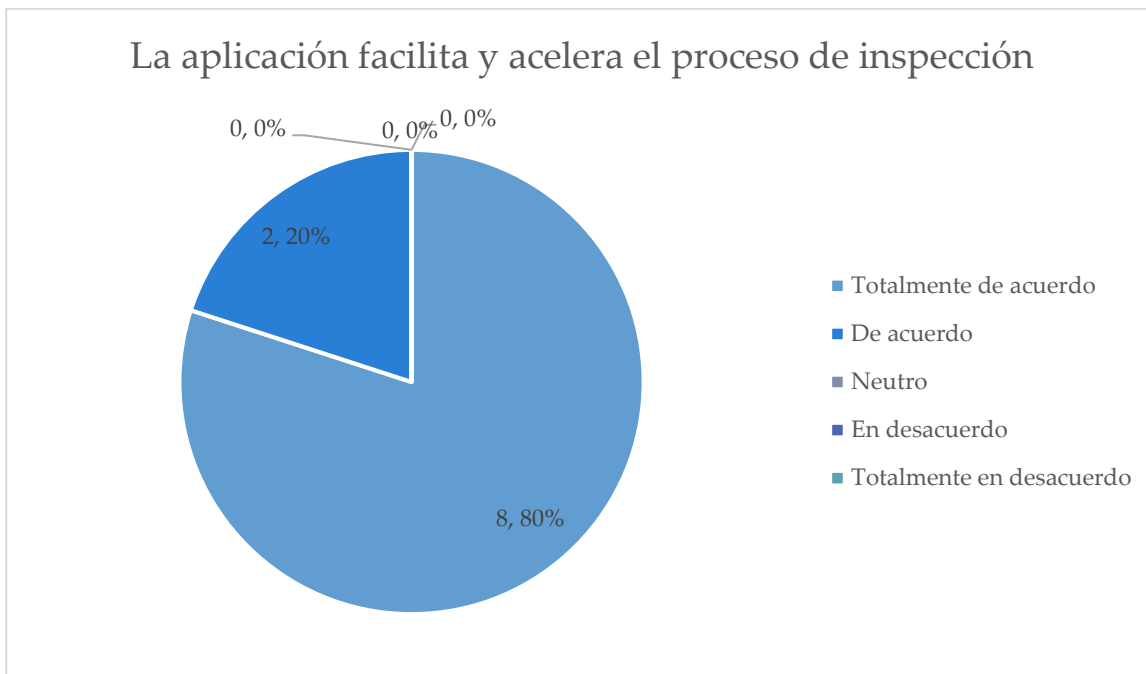


Ilustración 4.33: Resultados de pregunta 10 del instrumento de evaluación.

De este análisis se puede concluir que la aplicación resulta intuitiva y fácil de usar, se mantiene una consistencia en los mensajes y diseños en todas sus pantallas. Por otra parte, también se observa que los colores de la aplicación podrían mejorar, así como su diseño en general.

La aplicación es percibida como rápida, fluida y su navegación entre pantallas es bastante sencilla, lo cual ayuda a memorizar más fácilmente las acciones realizadas en ella. Los mensajes de error y el lenguaje usado para emitir avisos son muy efectivos, previniendo así una cantidad

importante de errores por mal uso de parte de los usuarios finales. En la aplicación no se añadieron tecnicismos en los mensajes para no agregar niveles de complejidad innecesarios.

En definitiva, basado en los resultados obtenidos de las pruebas de aceptación se concluye que la aplicación móvil es apta y beneficia el proceso de inspecciones vehiculares en compañías de seguro.

CONCLUSIONES

Se cumplió satisfactoriamente el objetivo del presente Trabajo Especial de Grado, el cual consistió en el desarrollo de una solución tecnológica móvil para el proceso de inspección de vehículos en compañías de seguros. Se realizó una aplicación móvil multiplataforma, la cual cumple con todos los requisitos planteados al principio del desarrollo.

Se implementó el sistema manejador de base de datos móvil SQLite satisfactoriamente, permitiendo así una persistencia local del dispositivo que beneficia a los usuarios sin conexión. Se ejecutaron todas las pruebas funcionales a cada módulo del sistema comprobando así su óptimo funcionamiento. Por otro lado, con las pruebas de aceptación se obtuvieron resultados que permiten tener una retroalimentación de parte de los usuarios finales muy valiosa, ya que se pueden tomar en cuenta los factores a mejorar en un futuro desarrollo o mantenimiento de la aplicación.

Haciendo uso de la metodología de desarrollo ágil SCRUM se mantuvo una organización en cuanto a las tareas y actividades, así como con la programación de pruebas. Mejorando cada vez con cada iteración en la que participan los respectivos actores para mejorar el proceso de desarrollo del producto final.

Igualmente, el framework Ionic fue de gran ayuda por su compatibilidad multiplataforma y por ser desarrollado en lenguajes web tan conocidos como HTML, CSS y Javascript, por supuesto, para la realización del proyecto en Ionic es fundamental trabajar con Angular, el cual provee un conjunto de librerías bastante útiles y una comunidad en línea que mantiene vivo al framework.

En síntesis, el uso de todas estas tecnologías, estilos de arquitectura y metodologías de desarrollo permitieron cumplir con todos objetivos planteados para este proyecto.

LIMITACIONES

- Las pruebas funcionales de la aplicación móvil se realizaron en un dispositivo móvil corriendo con sistema operativo Android, esto debido a las complicaciones que existen para desplegar la aplicación móvil en otras plataformas. Lo cual fue aclarado en el alcance del proyecto desde su propuesta.

- La comunicación entre la base de datos central y la base de datos móvil se realizó en un ambiente local de prueba, ya que no se contó con un servidor alojado en la web, se utilizó un servidor y una red de comunicación local, de esta manera no intervino el uso de datos del dispositivo móvil.

RECOMENDACIONES

Las siguientes recomendaciones pueden mejorar el rendimiento del sistema, y servir de apoyo o referencia para trabajos futuros.

- Incorporar un módulo de gestión de documentos requeridos por el cliente para tener un mejor control de los documentos que se tienen en la organización, eso permitiría ampliar la aplicación móvil para enviar o consultar documentos en alguna interfaz.
- Ampliar el módulo web para realizar inspecciones de siniestros, digitalizando más procesos y dando oportunidad de desarrollo tanto al módulo web como al módulo móvil.
- Se recomienda desplegar la aplicación móvil en los demás sistemas operativos móviles que son compatibles y se encuentran en el mercado.
- Se recomienda realizar un cambio en el diseño para que se adapte siempre a la organización que quiera hacer uso de ella.

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

- Albarán, I. (2008). *La actividad aseguradora: importancia, revisión e integración de conceptos*. Recuperado el 03 de Noviembre de 2017, de <http://eprints.ucm.es/6723/1/0022.pdf>
- Amaro Calderón, S. D., & Valverde Rebaza, J. C. (2007). *Metologías Ágiles*. Tesis, Escuela de Informática, Trujillo.
- Atlassian. (27 de 1 de 2018). *Kanban - A brief introduction*. Obtenido de Atlassian Agile Couch: <https://www.atlassian.com/agile/kanban>
- Baz Alonso, A., Ferreira Artime, I., Álvarez Rodríguez, M., & García Baniello, R. (2011). *Dispositivos móviles*. Universidad de Oviedo, Ingeniería de Telecomunicación. Recuperado el 13 de Octubre de 2017, de http://isa.uniovi.es/docencia/SIGC/pdf/telefonía_movil.pdf
- Castro Bernal, A. M. (21 de Septiembre de 2013). *Slidesahre*. Recuperado el 23 de Octubre de 2017, de Aplicaciones web: <https://www.slideshare.net/auramariacastro/aplicaciones-web-26419120>
- Dirección General de Política Económica y Empresarial. (s.f.). Uso de dispositivos móviles (teléfonos móviles, "smartphones", "ebooks", GPS y "tablets"). *Acércate @ las TIC*, 29. Recuperado el 2 de Enero de 2018, de <https://www.navarra.es/NR/rdonlyres/48F9746B-080C-4DEA-BD95-A5B6E01797E1/315641/7Usodedispositivosmoviles.pdf>
- Elmasri, R., & Navathe, S. (2007). *Fundamentos de Sistemas de Bases de Datos* (Quinta ed.). Pearson Addison Wesley.
- Fling, B. (2009). *Mobile Design and Development: Practical concepts and techniques for creating mobile sites*. O'Reilly Media. Recuperado el 17 de Diciembre de 2017
- Google. (28 de 01 de 2018). *AngularJS: Development Guide Introduction*. Obtenido de AngularJS: <https://docs.angularjs.org/guide/introduction>
- Gutiérrez, J. (2014). *¿Qué es un framework web?* Recuperado el 26 de Octubre de 2017, de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Johansen Bertoglio, O. (1993). *Introducción a la teoría general de los sistemas* (Primera ed.). México: Limusa S.A.
- López Ortiz, F. (2010). *El estándar IEEE 802.11 Wireless LAN*. Obtenido de <http://www.dit.upm.es/~david/tar/trabajos2002/08-802.11-Francisco-Lopez-Ortiz-res.pdf>
- McWherter, J., & Gowell, S. (2012). *Professional Mobile Application Development*. Wrox. Recuperado el 17 de Diciembre de 2017
- Mozilla. (26 de 01 de 2018). *Introducción - Javascript MDN*. Obtenido de MDN Web Docs: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>
- Orozco Tapia, A. M. (15 de marzo de 2012). *Modelos de BD*. Obtenido de PNFI – IUTEP 2012: <https://modelosbd2012t1.wordpress.com/2012/03/15/base-de-datos-moviles-3/>
- Ospina, M. (2009). *Fundamentos y Conceptos Básicos de Bases de datos*. Centro de Investigación en Sistemas de Información CISI, Caracas.
- Owens, M., & Allen, G. (2010). *The Definitive Guide to SQLite* (Second ed.). Apress. Recuperado el 6 de Octubre de 2017
- Palacio, J. (2014). *Gestión de Proyectos Scrum Manager*. Scrum Manager.
- QBS Software Ltd. (2017). *qbs software*. Recuperado el Octubre de 2017, de http://www.qbssoftware.com/products/SQLBase/overview/_prodsqldb
- Ramírez Vique, R. (2011). *Métodos para el desarrollo de aplicaciones móviles*. Universidad Oberta de Catalunya. Recuperado el 26 de Octubre de 2017, de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)

- Rivero Hernández, D., Pérez Vázquez, R., & Vila Labrada, J. (Diciembre de 2013). Bases de Datos Móviles. *Revista Académica de Investigación*(14), 20. Recuperado el 5 de Octubre de 2017, de <http://www.eumed.net/rev/tlatemoani/14/base-datos-moviles.pdf>
- Schwaber, K., & Sutherland, J. (2016). *La Guía de Scrum*. Recuperado el 24 de Octubre de 2017
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos* (Cuarta ed.). Madrid, España: McGraw-Hill Inc.
- Silberschatz, A., Korth, H., & Sudarshan, S. (3 de Septiembre de 2006). *Fundamentos de Bases de Datos* (Quinta ed.). Madrid: Mc Graw Hill. Recuperado el 25 de Noviembre de 2017, de Portal de sitios web de asignaturas: http://www.ciens.ucv.ve/portalsig/bases_de_datos/1-2015/descarga#
- The PostgreSQL Global Development Group. (2018). *PostgreSQL: About*. Obtenido de <https://www.postgresql.org/about/>
- Torres, J. M. (2008). *SQL Server Compact 2008 SP1 Referencia Completa*. Krasis Press. Recuperado el Octubre de 2017