



**UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA COMPUTACIÓN  
CENTRO DE INVESTIGACION DE SISTEMAS DE INFORMACIÓN**

**DESARROLLO DE UNA SOLUCIÓN  
MÓVIL QUE BRINDE SOPORTE A LA  
GESTIÓN DE ÓRDENES DE SERVICIO  
EN UNA ORGANIZACIÓN**

Trabajo de Especial de Grado presentado ante la ilustre  
Universidad Central de Venezuela por los bachilleres  
**Manuel José Rojas Ramírez C.I. 16.819.812**  
**José Ángel Castro Lapenna C.I. 14.485.049**

Para optar al título de  
Licenciado en Computación

Tutor: Prof. Franklin J. Sandoval S.

Caracas, Mayo del 2018

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

## Acta

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de la Escuela de Computación, para examinar el Trabajo Especial de Grado representado por los Bachilleres: **Manuel José Rojas Ramírez C.I. 16.819.812 y José Ángel Castro Lapenna C.I. 14.485.049**, con el título: “Desarrollo de una solución móvil que brinde soporte a la gestión de órdenes de servicio en una organización”, a los fines de optar al título de **Licenciado en Computación**, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 16 de Mayo de 2018, a las 9:00 Am, para que sus autores lo defendieran en forma pública en la Sala de videoconferencias del Centro de Computación de la Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, lo cual se realizó mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 19 puntos.

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



Prof. Franklin Sandoval  
Tutor



Profa. Mercy Opina.  
Jurado



Profa. Yosly Hernández  
Jurado

## **Dedicatoria**

Dedico este proyecto a Dios por haberme dado la oportunidad de vivir, por llenarme de bendiciones, y darme la fuerza y perseverancia para culminar esta etapa de mi vida.

A mis padres por transmitirme los mejores principios y valores humanos, por su valioso esfuerzo para darme la mejor educación, por haberme brindado su apoyo incondicional en todo lo que me he propuesto conseguir; por estar allí conmigo en los momentos de mayor adversidad.

A mi hermano por su apoyo incondicional e impulsarme a continuar con mi carrera universitaria y a salir adelante en momentos difíciles.

A mi esposa que siempre ha estado conmigo en todo momento, apoyándome en las buenas y en las malas; siempre alentándome a lograr mis metas y a nunca desfallecer en el camino.

A la memoria de mi abuela, que siempre tuvo el deseo de verme formado como una persona íntegra de bien y profesional.

**Manuel J. Rojas R.**

A mi madre quien me dio la vida, educación, apoyo y consejos.

A mis abuelos maternos por ser unos segundos padres para mí.

A mi familia, mis compañeros de estudio, a mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis.

A todos ellos se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

**José A. Castro**

## **Agradecimientos**

Agradezco ante todo a Dios por permitirme llegar a este punto de mi carrera universitaria.

Gracias a mis padres y mi esposa que durante todo este tiempo fueron pilar fundamental con su apoyo moral siempre allí presente.

Gracias al profesor Franklin Sandoval, por ser una excelente persona como amigo y profesional, siendo de una ayuda muy valiosa durante mi carrera y proyecto de grado. Por sus palabras de aliento y motivación para culminar con éxito este emprendimiento.

Gracias a la profesora Brenda López quien siempre me supo orientar al comienzo del proyecto de tesis; y al resto de profesores de la Escuela de Computación de la Facultad de Ciencias de la UCV, porque en un grado mayor o menor fueron de ayuda durante mi trayectoria académica en la universidad. Gracias por su disposición a compartir sus conocimientos de la mejor manera y contribuir a mi formación como profesional.

**Manuel J. Rojas R.**

A Dios por permitirme llegar hasta este punto y darme salud para lograr mis objetivos.

A mi madre María, por haberme apoyado en todo momento, por sus valores, por su sacrificio, constancia y dedicación a sus hijos, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi hermana María Gabriella y mi sobrina Cristina, por formar parte de mi vida y ser fuente de inspiración y sacrificio.

A mi compañera Rosiris, por el apoyo incondicional todos estos años.

A mi tutor Franklin, pocas personas en la vida te tienden la mano realmente y él ha sido una de ellas; gracias por la preocupación y la perseverancia en mi época más difícil.

Gracias a todos los que ayudaron directa o indirectamente en la realización de esta meta.

**José A. Castro**

**Universidad Central de Venezuela.  
Facultad de Ciencias  
Escuela de Computación  
Centro de Investigación en Sistemas de Información**

## **DESARROLLO DE UNA SOLUCIÓN MÓVIL QUE BRINDE SOPORTE A LA GESTIÓN DE ÓRDENES DE SERVICIO EN UNA ORGANIZACIÓN**

**Autores:** Manuel Rojas y José Castro  
**Tutor:** Prof. Franklin Sandoval.

### **RESUMEN**

Este Trabajo Especial de Grado tuvo como objetivo desarrollar una solución de software móvil que funcione como soporte para los procesos de gestión de órdenes de servicio en una organización. La solución se compone de una aplicación móvil híbrida y una aplicación web de apoyo. El componente móvil fue desarrollado haciendo uso de PhoneGap y jQuery Mobile, los cuales son frameworks basados en los estándares web de HTML5, CSS3 y JavaScript. Éstos últimos son utilizados también en conjunto con el lenguaje de programación PHP para el desarrollo de la aplicación web. Para lograr el objetivo, se realizó una adaptación de Scrum como marco de trabajo de la metodología ágil para el control y gestión en el desarrollo de la solución de software propuesta. En el desarrollo del Trabajo Especial de Grado, se provee la documentación necesaria de la solución presentada, desde el levantamiento de requerimientos, diseño, puesta en marcha y pruebas de funcionamiento, dando respuesta a los objetivos planteados. El sistema permite realizar el seguimiento a la gestión de órdenes de servicio en una organización, permitiendo llevar un control más efectivo del proceso.

**Palabras Claves:** aplicación móvil, teletrabajo, multiplataforma, orden de servicio, Scrum, tecnología web, phonegap

## ÍNDICE GENERAL

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPITULO 1 EL PROBLEMA.....</b>	<b>3</b>
1.1.- Situación Actual .....	3
1.2.- Planteamiento del problema .....	5
1.3.- Solución .....	7
1.4.- Objetivos.....	10
1.4.1.- General .....	10
1.4.2.- Específicos.....	10
1.5.- Alcance .....	11
<b>CAPITULO 2 MARCO CONCEPTUAL .....</b>	<b>12</b>
2.1- Orden de servicio.....	12
2.1.1- Definición .....	13
2.1.2.- Elementos de una orden de servicio .....	13
2.1.3.- Ciclo de vida de una orden de servicio.....	15
2.1.4.- Sistema de gestión de órdenes de Servicio .....	18
2.1.5.- Beneficios de los sistemas de órdenes de servicios .....	19
2.1.6.- Tecnología móvil en órdenes de servicio .....	20
2.2.- Sistemas de información.....	22
2.3.- Aplicaciones móviles. ....	23
2.3.1.- Tipos de aplicaciones móviles.....	23
2.3.1.1.- Aplicaciones móviles nativas. ....	24
2.3.1.2.- Aplicaciones web móviles .....	25
2.3.1.3.- Aplicaciones móviles híbridas .....	27
2.4.- Dispositivos móviles .....	29
2.4.1.- Clasificación de los dispositivos móviles .....	29
2.5.- Sistemas operativos móviles .....	32
2.5.1.- Componentes.....	32
2.5.2.- Principales sistemas operativos móviles .....	35
2.6.- PhoneGap.....	41
2.6.1.- Historia .....	42
2.6.2.- Retos en el Desarrollo de Aplicaciones Multiplataforma .....	42
2.6.3.- Cómo funciona PhoneGap .....	44
2.6.4.- Limitaciones de PhoneGap .....	46
2.6.5.- Ventajas de PhoneGap .....	47

<b>CAPITULO 3 MARCO METODOLÓGICO .....</b>	<b>48</b>
3.1.- Scrum .....	50
3.1.1.- Teoría de Scrum.....	52
3.1.2.- Visión general del proceso .....	53
3.1.2.1.- Elementos .....	53
3.1.2.2.- Roles.....	53
3.1.2.3.- Reuniones.....	54
<b>CAPITULO 4 MARCO APLICATIVO .....</b>	<b>56</b>
4.1.- Contexto del Desarrollo .....	56
4.2.- Proyecto.....	56
4.3.- Aplicación de la metodología de desarrollo .....	57
4.3.1.- Definición de Roles .....	57
4.3.2.- Elementos .....	58
4.3.3.- Reuniones y aplicación en el proyecto .....	59
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>123</b>
<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>127</b>

## INDICE DE FIGURAS

Figura 1. Solución propuesta .....	10
Figura 2. Ejemplo de formulario de orden de servicio (Vertext42 LLC,2010)	14
Figura 3. Aplicaciones móviles: nativas vs web (Eternal Software, 2013) ....	27
Figura 4. Arquitectura de una aplicación móvil híbrida (Garzo, 2012) .....	28
Figura 5. Ejemplos de dispositivos móviles: laptop, tableta y teléfono inteligente (Cruz, 2015).....	32
Figura 6. Estructura de un sistema operativo móvil (Android vs iOS, 2013) .	33
Figura 7. Arquitectura del sistema operativo Android (Android OS: Arquitectura, 2012).....	37
Figura 8. Arquitectura del sistema operativo iOS (Android vs iOS, 2013).....	41
Figura 9. Funcionamiento de PhoneGap (Asociación javaHispano, 2012) ...	42
Figura 10. Arquitectura de aplicación PhoneGap (Wargo, 2012).....	46
Figura 11. Núcleo del proceso Scrum. (El método Scrum, 2007) .....	52
Figura 12. Modelo E/R de la base de datos .....	62
Figura 13. Diagrama de Casos de Uso Creación de Órdenes de Servicio en el Sistema Web.....	63
Figura 14. Diagrama de Casos de Uso para la Creación de Órdenes de Servicio en el Sistema Móvil .....	64
Figura 15. Diagrama de Casos de Uso para la Actualización de Órdenes de Servicio .....	64
Figura 16. Diagrama de Casos de Uso para Agregar Cliente .....	65
Figura 17. Diagrama de Casos de Uso para Agregar Partes y Materiales a una Orden de Servicio en el Sistema Móvil .....	65
Figura 18. Diagrama de Casos de Uso para Agregar Partes y Materiales a una Orden de Servicio en el Sistema Web .....	66
Figura 19. Diagrama de Casos de Uso para Agregar Técnico .....	66
Figura 20. Diagrama de Casos de Uso para Cancelar una orden de servicio .....	67

Figura 21. Diagrama de Casos de Uso para Confirmar una Orden de Servicio .....	67
Figura 22. Diagrama de Casos de Uso para Ejecutar una Orden de Servicio .....	68
Figura 23. Diagrama de Casos de Uso para sincronizar Órdenes de Servicio .....	68
Figura 24. Página principal de la aplicación móvil .....	70
Figura 25. Página principal de la aplicación web .....	71
Figura 26. Página de login de la aplicación móvil .....	71
Figura 27. Página de login de la aplicación web .....	72
Figura 28. Conexión a la base de datos móvil .....	73
Figura 29. Página de crear orden de la aplicación móvil .....	77
Figura 30. Página de crear orden de la aplicación web .....	77
Figura 31. Página de lista de órdenes de servicio .....	78
Figura 32. Página de Lista de Clientes .....	79
Figura 33. Página de Lista de Partes/Materiales .....	79
Figura 34. Página de Lista de Técnicos .....	80
Figura 35. Página de Detalle de Partes .....	80
Figura 36. Página de Detalle de Cliente .....	81
Figura 37. Página de Detalle de Técnico .....	82
Figura 38. Página de confirmar orden de servicio de la aplicación web .....	83
Figura 39. Fragmento de código del servicio web.....	84
Figura 40. Página de cancelar orden de servicio .....	89
Figura 41. Página de calendario de órdenes de servicio asignadas a un técnico .....	89
Figura 42. Página de reporte de orden de servicio en la aplicación móvil ....	91
Figura 43. Página de reporte de orden de servicio en la aplicación web .....	92
Figura 44. Página de agregar partes y materiales a una orden de servicio ..	93
Figura 45. Fragmento de código de la conexión al servicio web.....	94

Figura 46. Función para la gestión de contenido de la página principal por perfil de usuario .....	95
Figura 47. Función que implementa la autenticación de usuarios en la aplicación móvil.....	96
Figura 48. Función que implementa la lista de clientes en la aplicación móvil .....	97
Figura 49. Fragmento de código de función para crear orden de servicio ..	103
Figura 50. Fragmento de código de función para listar las órdenes de servicio .....	104
Figura 51. Fragmento de código de función para confirmar una nueva orden de servicio.....	105
Figura 52. Fragmento de código de función para cancelación de una orden de servicio.....	106
Figura 53. Fragmento de código de función para el calendario de órdenes de servicio.....	107
Figura 54. Fragmento de código de función para el reporte de ejecución órdenes de servicio.....	108
Figura 55. Fragmento de código de función para agregar partes y/o materiales a una orden de servicio.....	109
Figura 56. Fragmento de código de función para sincronizar desde el servidor hacia la aplicación móvil .....	110

## INDICE DE TABLAS

Tabla 1. Estados del ciclo de vida de una orden de servicio .....	17
Tabla 2. Lenguajes, Herramientas y Formatos de Paquetes de las principales plataformas móviles .....	43
Tabla 3. Actores de Scrum en el proyecto .....	58
Tabla 4. Pila del Sprint.....	58
Tabla 5. Pila del Producto .....	59
Tabla 6. Estado inicial de la pila del producto .....	60
Tabla 7. Pila del Sprint 1 .....	61
Tabla 8. Prueba de aceptación de control de acceso a la aplicación.....	74
Tabla 9. Prueba de aceptación acceso página principal de la aplicación .....	74
Tabla 10. Estado de la pila del producto luego del Sprint 1 .....	75
Tabla 11. Pila del Sprint 2.....	76
Tabla 12. Prueba aceptación de acceso Página de crear nuevas órdenes ..	85
Tabla 13. Prueba aceptación de acceso a la página de lista de órdenes .....	85
Tabla 14. Prueba aceptación de acceso a la página de lista de clientes .....	86
Tabla 15. Prueba aceptación de acceso página confirmación nueva orden .	86
Tabla 16. Estado de la pila del producto luego del Sprint 2 .....	87
Tabla 17. Pila del Sprint 3.....	88
Tabla 18. Prueba acceso página de cancelación de órdenes de servicio.....	98
Tabla 19. Prueba acceso a la página de calendario de órdenes de servicio	99
Tabla 20. Prueba acceso a la página de reporte de órdenes de servicio .....	99
Tabla 21. Prueba acceso a la página de agregar partes y materiales .....	100
Tabla 22. Estado de la pila del producto luego del Sprint 3 .....	101
Tabla 23. Pila del Sprint 4 .....	102
Tabla 24. Prueba de creación de órdenes de servicio .....	111
Tabla 25. Prueba de mostrar lista de órdenes de servicio .....	111
Tabla 26. Prueba de confirmación de órdenes de servicio .....	112
Tabla 27. Prueba de cancelación de órdenes de servicio.....	112

Tabla 28. Prueba de mostrar calendario de órdenes de servicio ..... 113

Tabla 29. Prueba de reporte de ejecución de órdenes de servicio ..... 113

Tabla 30. Prueba agregar partes y/o materiales a una orden de servicio... 114

Tabla 31. Prueba de sincronización de órdenes de servicio ..... 114

Tabla 32. Estado de la pila del producto luego del Sprint 4 ..... 115

## INTRODUCCIÓN

El uso acelerado de tecnologías móviles ha impulsado una revolución en la manera de llevar a cabo las tareas de trabajo, estudio o simplemente entretenimiento. Hoy en día es común observar a las personas en los restaurantes, transporte público, parques, supermercados, entre otros., llevar a cabo tareas tan simples como revisar una lista de pendientes o tan complejas como realizar una reunión de trabajo virtual.

En el ámbito de las organizaciones, se hace imprescindible contar con herramientas que faciliten o ayuden a cumplir con las tareas del negocio y, en consecuencia, generen un impacto positivo en términos de productividad. La gestión de órdenes de servicio es un pilar fundamental para el proceso de negocio de casi cualquier organización, pero no siempre se cuenta con el enfoque más idóneo para realizar dicha gestión de la manera más eficiente.

El desarrollo de aplicaciones móviles es un mercado enorme en la actualidad. Hay muchas empresas que invierten mucho dinero para desarrollar exitosas y rentables aplicaciones. El problema aumenta cuando se trata de desarrollar una aplicación para ser utilizada por todos los usuarios independientemente de la plataforma o sistema que estén operando. La forma tradicional de enfrentar este problema ha sido siempre la de contratar a más desarrolladores y dividir el proceso de desarrollo entre las distintas plataformas. Sin embargo, esta solución puede parecer ineficiente para una pequeña empresa que no puede permitirse contratar a todos los recursos que hagan falta. Por esta razón, en los últimos años han aparecido varias soluciones basadas en marcos de trabajo que permiten el desarrollo utilizando tecnología web o de capa media, lo cual ha facilitado la creación de aplicaciones que se ejecutan en múltiples plataformas.

El objetivo del presente trabajo de investigación es el desarrollo de una solución móvil que sirva de soporte a la gestión de órdenes de servicio dentro de una organización.

El trabajo está estructurado por capítulos. A continuación se presenta un resumen de cada uno de ellos.

Capítulo 1: En este capítulo se plantea la propuesta de trabajo especial de grado, donde se describe el problema, objetivos, una propuesta de solución y el alcance para el desarrollo de la solución móvil que soporte la gestión de órdenes de servicio.

Capítulo 2: En este capítulo se describen fundamentos teóricos, donde se definen los distintos elementos que componen una solución móvil. Se realiza una descripción del tema órdenes de servicio, de conceptos inmersos en el ambiente de las tecnologías móviles y de algunas herramientas que facilitan la construcción de soluciones móviles.

Capítulo 3: En este capítulo, se presenta el marco metodológico, donde se analiza Scrum como metodología de desarrollo de la solución móvil, explicando en detalle los conceptos teóricos y las etapas del proceso.

Capítulo 4: En este capítulo, se presenta el marco aplicativo, se describe el contexto de desarrollo del proyecto y la adaptación y aplicación de Scrum en el proceso de desarrollo de la solución móvil. Por último se presentan las conclusiones, recomendaciones y referencias bibliográficas de la presente investigación

# CAPITULO 1

## PROBLEMA DE INVESTIGACIÓN

En este capítulo se describe el contexto del problema, luego se expone el objetivo general y los objetivos específicos, así como la justificación de la realización del presente Trabajo de Especial de Grado.

### 1.1.- Situación Actual

Cada día la tecnología cambia la forma de entender el mundo y con ello la manera de llevar a cabo la ejecución de cualquier tarea. La evolución de los sistemas tecnológicos ejerce una influencia directa en el futuro de las relaciones económicas, sociales y culturales de las sociedades, donde las personas y las organizaciones tienen que adaptarse a los cambios producidos por dicha evolución. Es por ello que los profesionales que desean prosperar deben enfrentar cada vez mayores exigencias, para incrementar la eficiencia y productividad, y por consiguiente, el desarrollo profesional, mediante el uso optimizado de los recursos. Los avances tecnológicos han configurado un nuevo escenario tanto en los sistemas informáticos como en las redes de telecomunicaciones, telemática, los teléfonos celulares, la televisión, la radio, los periódicos digitales, faxes, dispositivos portátiles, entre otros.

La movilidad se puede considerar como la innovación de la tecnología más disruptiva que enfrentan las empresas hoy en día. En los últimos años, los dispositivos móviles han evolucionado a partir de limitarse a proporcionar correo electrónico de la empresa y los datos sobre la marcha a la introducción de aplicaciones y servicios que tienen el potencial para cambiar la forma de

hacer negocios. La movilidad puede ayudar a aprovechar los datos contextuales de las empresas para llegar a sus clientes más eficazmente.

Las tecnologías móviles hoy en día permiten a las empresas realizar negocios sin importar su ubicación fija, desbloqueando un inmenso valor y desatando soluciones innovadoras para ayudar más a las empresas. La movilidad es una pieza clave dentro de la tecnología de la información; el auge del mercado móvil, la demanda del usuario por teléfonos inteligentes y la rápida innovación en los dispositivos están dirigiendo el futuro de la plataforma de usuarios finales. El uso de tecnología móvil provee oportunidades para la innovación, agilidad y flexibilidad en el lugar de trabajo.

Aunque para el manejo de operaciones diarias y gestión del proceso de órdenes de servicio las empresas utilizan varios sistemas de software, el personal técnico tiene que llevar consigo siempre formularios en papel para reportar el trabajo realizado.

El proceso tradicional de gestión de órdenes de servicio comienza cuando el cliente comunica a la empresa, bien sea de manera oral o escrita, su necesidad de obtener determinado servicio. Un personal calificado se encarga de tomar nota de la solicitud y registrarla de manera formal manualmente. Se crea entonces una orden de servicio en papel y se coloca en algún tipo de sistema de archivo en espera por aprobación y asignación a un técnico de servicio. El personal técnico es contactado para que pase a retirar la orden en la oficina y acuda luego al sitio señalado para ejecutar el servicio descrito en la orden. Luego de terminar el trabajo, el técnico de servicio debe estar atento y completar toda la información requerida en la orden de manera correcta.

La orden de servicio es regresada luego a la oficina, donde es sometida a revisión y aprobación para luego generar la correspondiente factura que será enviada al cliente. Se debe actualizar también el inventario de partes y

materiales utilizados en caso de ser necesario. Finalmente, una copia de la factura se adjunta a la orden de servicio y se guarda en un historial de archivos.

## **1.2.- Planteamiento del problema**

El uso de órdenes de servicio en papel requiere un consumo de tiempo considerable en el ingreso de los datos y a menudo esto conduce a pérdida o errores en la información. A eso se suma el tiempo invertido por el personal técnico en ir a la oficina a retirar la orden y traerla de vuelta una vez culminado el trabajo.

Durante el ciclo de vida de la orden de servicio en papel existen muchos puntos de contacto. La orden de servicio pasa por la persona que la registra, la persona encargada de aprobar su ejecución y el personal técnico. Luego va de regreso a la persona que la aprueba para, finalmente, llegar hasta el departamento de contabilidad. Durante este proceso, existe el riesgo de que se extravíe la orden o de que ocurran errores.

Estos hechos impactan negativamente en los costos y tiempos de respuesta, conduciendo a un nivel poco óptimo de productividad de la empresa y calidad del servicio entregado al cliente.

Buscando sortear tales problemas, las organizaciones muchas veces optan por implementar complejos sistemas web con registro de información en línea. Los técnicos de servicio son equipados con laptops robustas para acceder a la información en tiempo real desde el sitio de trabajo. Pero mantener las laptops actualizadas consume recursos y las aplicaciones pueden no ser amigables. Al mismo tiempo se hace necesario contar con una conexión permanente a Internet lo cual ocasiona una movilidad reducida.

Siendo la movilidad un punto crítico en las empresas de servicio, sobre todo en aquellas orientadas exclusivamente al servicio en campo, surge entonces

la necesidad de adoptar estrategias que promuevan el aprovechamiento de las tecnologías móviles, integrándolas dentro de los procesos de gestión de órdenes de servicio.

Muchas organizaciones que dan los primeros pasos para implementar una estrategia móvil están enfrentando una decisión importante que influirá en los resultados de esta iniciativa. El proceso de escoger una estrategia de desarrollo para una aplicación móvil, llámese nativa, web o híbrida, implica muchos parámetros tales como presupuesto, ventana de tiempo del proyecto, audiencia objetivo y funcionalidad de la aplicación, por nombrar solo algunas. Cada estrategia conlleva beneficios inherentes y limitaciones, y encontrar la que mejor se adapte a las necesidades de la organización podría ser una tarea desafiante.

Al mismo tiempo, la portabilidad es el problema número uno cuando se desarrollan aplicaciones móviles para múltiples plataformas. Esto es debido a varias razones, pero las dos más prominentes son el reto que impone la diversidad de dispositivos móviles que existen en el mercado y el hecho de que el rendimiento es una de las mayores prioridades cuando se desarrollan aplicaciones móviles.

Las aplicaciones destinadas a las plataformas iOS y Android están escritas en lenguajes completamente diferentes. Las aplicaciones para iOS están escritas en Objective-C o Swift mientras que aquellas para dispositivos Android están escritas en Java. La diferencia más obvia entre las plataformas es el lenguaje en el cual están escritas las aplicaciones. Adicionalmente, los desarrolladores deberían ser conscientes de las distintas capacidades de hardware tales como la tarjeta SD externa, el GPS y la cámara.

Con el propósito de mantener relevancia en el mercado de aplicaciones actual, los desarrolladores deben adoptar los conceptos de desarrollo multiplataforma para asegurar que las aplicaciones sean destinadas a tantas plataformas

diferentes como sea posible. En tal sentido, se hace necesario que cualquier aplicación desarrollada para una plataforma esté también disponible para otras existentes con la habilidad de ser portada a futuras plataformas.

Cada plataforma consiste de perfiles de hardware separados, incluyendo procesador y memoria, así como también el tamaño de la pantalla y otras opciones tales como cámaras y Bluetooth. Estos perfiles de hardware causan que los API (Interfaces de Programación de Aplicación) específicos de la plataforma difieran entre cada una de ellas.

Estos factores resultan en un incremento de costos en términos de tiempo y dinero invertido en el proceso de re-diseño y el costo de la oportunidad de ese tiempo de desarrollo al no ser invertido en el diseño de una nueva aplicación o, al menos, en mantener y mejorar la aplicación ya entregada. Adicionalmente, el segundo proceso de diseño a menudo resulta en un código de base muy distinto, que incrementa los costos de mantenimiento y puede conducir a aplicaciones con características diferentes. En otros casos, simplemente, se tiende a desarrollar una única versión de la aplicación y perder una cuota de potenciales usuarios importante con tal de reducir gastos.

### **1.3.- Solución**

Con el fin de abordar el problema antes descrito, se propone una solución que implemente una aplicación móvil híbrida multiplataforma que brinde soporte a los procesos inherentes a la gestión de órdenes de servicio.

Se propone una arquitectura lo suficientemente flexible que permita desarrollar la aplicación en un lenguaje de programación diferente al que se utilizaría nativamente en las diferentes plataformas y con este código en lo que se podría llamar un “lenguaje neutro”, llevar a cabo compilaciones específicas sobre cada una de las plataformas móviles.

Para lograr el objetivo multiplataforma, se plantea el uso de PhoneGap. Éste no es un framework pensado para desarrollar aplicaciones que exploten al 100% las capacidades de un dispositivo, sino que más bien se centra en el contenido y estructuración de la información, algo que en definitiva se ajusta a las necesidades del proyecto actual. Además, se eligió PhoneGap por encima de herramientas similares como Apache Cordova o Ionic en primer lugar porque Adobe permite la integración de PhoneGap con sus servicios de compilación (Adobe PhoneGap Build), lo que ahorra el trabajo de descargar e instalar los SDKs de cada uno de los entornos como sucede con Apache Cordova, y en segundo lugar porque PhoneGap es más básico permitiendo un desarrollo más rápido que utilizando Ionic, el utiliza AngularJS y hubiese requerido un tiempo mayor de adiestramiento a dicha herramienta.

Se propone el desarrollo de una aplicación móvil apoyada en un sistema web para la creación, asignación y gestión de las órdenes de servicio. Esta solución contempla la definición de una estructura de datos que permita almacenar y manipular de forma detallada la información correspondiente a las órdenes de servicio y los usuarios del sistema.

Existen tres perfiles de usuarios:

- Despachador: encargado de recibir las solicitudes de órdenes de servicio, procesarlas y transformarlas en una orden de servicio real, en caso de ser necesario.
- Gerente: revisa, complementa y valida las órdenes de servicios creadas para luego asignarlas al personal técnico encargado de su ejecución.
- Técnico: ejecuta las tareas correspondientes a las órdenes de servicio que le han sido asignadas.

En la solución planteada, se asume que las solicitudes de servicio son recibidas vía email, fax o telefónica.

Cuando el usuario emite una solicitud de orden de servicio, ésta es recibida y analizada por el despachador. Una vez verificada y validada la existencia real del problema, el despachador procede a convertir la solicitud de servicio en una orden de servicio. Esto se puede lograr desde la aplicación móvil o desde el sistema web de apoyo. Sólo el despachador y el gerente tienen privilegios suficientes para la creación de órdenes de servicio.

Una vez creadas, las órdenes de servicio son revisadas y validadas por el Gerente antes de ser asignadas al personal técnico correspondiente. Si es necesario, en este punto se pueden modificar o añadir nuevos datos a la orden de servicio.

El perfil de Gerente comprende el mayor rango de privilegios dentro la solución móvil. Puede crear, modificar, cancelar o consultar todas las órdenes de servicio en el sistema, bien sea a través de la aplicación móvil o desde el sistema web de apoyo. El técnico sólo está autorizado para consultar y modificar (con ciertas restricciones) las órdenes de servicio que tiene asignadas a través del dispositivo móvil.

Una vez el técnico ha completado la orden de servicio, la misma queda sujeta a revisión y aprobación de parte del gerente. Al ser aprobada, se emite la correspondiente factura para ser enviada al cliente. Ver Figura 1.

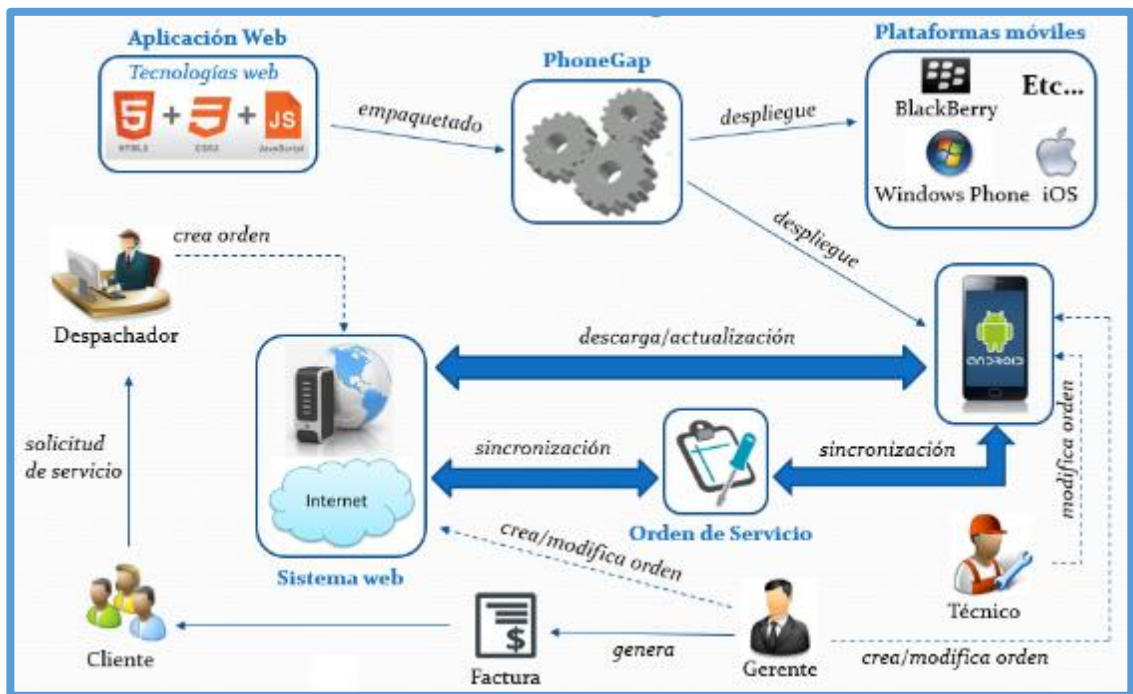


Figura 1. Solución propuesta

#### 1.4.- Objetivos de la investigación

En las siguientes secciones se presentaran los objetivos que se desean alcanzar con el trabajo especial de grado.

##### 1.4.1.- General

Desarrollar una solución móvil que brinde soporte a los procesos inherentes a la gestión de órdenes de servicio en una organización.

##### 1.4.2.- Específicos

Para cumplir con el objetivo general, a continuación se describen los objetivos específicos de la propuesta

- Analizar información recolectada sobre el proceso gestión de órdenes de servicio y las tecnologías móviles.
- Definir una arquitectura de solución que comprenda la aplicación móvil y el sistema web.

- Desarrollar la aplicación móvil híbrida que soporte la gestión de órdenes de servicio.
- Desarrollar la aplicación web que soporte la gestión de órdenes de servicio.
- Realizar pruebas de aceptación y usabilidad.

### **1.5.- Alcance**

El desarrollo de la solución móvil se limitará al diseño e implementación de un sistema móvil para la gestión de órdenes de servicio, comprendiendo sólo el proceso correspondiente al ciclo de vida de la orden. Para el sistema móvil se utilizará tecnología web móvil: HTML5, CSS3 y Javascript, junto con el framework PhoneGap. Para el sistema web de apoyo se utilizarán los lenguajes de desarrollo web PHP, HTML, CSS y Javascript junto con el sistema manejador de base de datos MySQL.

## **CAPITULO 2**

### **MARCO CONCEPTUAL**

A continuación, se describen fundamentos teóricos, donde se definen los distintos elementos, componentes y términos, que componen una solución móvil. También se describen algunas herramientas que facilitan la construcción de soluciones móviles. Previamente, se realiza un análisis y descripción detallados del tema de órdenes de servicio.

Antes de profundizar en el concepto de orden de servicio, es necesario conocer las definiciones básicas que la componen. Por ello, a continuación se presentan dos definiciones de servicios propuestas por reconocidos expertos en mercadotecnia.

#### **2.1- Orden de servicio**

Antes de establecer este concepto se debe definir que es un servicio, para lo cual Stanton, Etzel y Walker (2004) lo define "como actividades identificables e intangibles que son el objeto principal de una transacción ideada para brindar a los clientes satisfacción de deseos o necesidades", mientras que otros afirman que "un servicio es el resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente" (Lamb, Hair, & McDaniel, 2011).

Hoy en día, las compañías que ofrecen servicios operan, generalmente, en un ambiente muy competitivo y, por lo tanto, se encuentran constantemente en la búsqueda de nuevas formas de hacer que su negocio se eleve por encima de los demás. Esto forma parte del continuo crecimiento del mercado donde

surgen cada vez más clientes con nuevas necesidades de servicios. Una parte esencial de las comunicaciones entre una empresa que ofrece determinados servicios y sus clientes son las órdenes de servicio. En éstas se define el alcance del trabajo a ser realizado para lograr el objetivo de negocio: proveer un servicio al cliente. La orden de servicio debe registrar información valiosa como la descripción, ubicación, fecha, hora y cantidad de horas del servicio llevado a cabo. (Conjecture Corporation, 2014)

### **2.1.1- Definición**

De acuerdo con Conjecture (2014) una orden de servicio es un documento que provee detalles importantes sobre los servicios que un cliente quiere obtener de un fabricante, proveedor u otro departamento dentro de la misma compañía. Este documento, que podría ser un pedazo de papel impreso o un documento en un sistema informático o en Internet, incluye elementos que ayudan a saber exactamente al personal encargado de completar la orden de servicio lo que el cliente quiere que se haga. Se define el trabajo que se necesita hacer comprendiendo una o más tareas, las cuales son asignadas a un personal técnico. Por ejemplo, una orden de servicio se podría emitir cuando una pieza de un equipo necesite ser reparada, actualizada o reemplazada, se necesite hacer mantenimiento a algún equipo o en otra situación en la que el cliente tenga una necesidad de servicio.

### **2.1.2.- Elementos de una orden de servicio**

Los detalles propuestos por Vertex 42 (2010) que se pueden incluir en el formulario de una orden de servicio son:

- Número de orden de servicio
- Fecha de solicitud
- Nombre del cliente
- Dirección de facturación
- Dirección de entrega



### **2.1.3.- Ciclo de vida de una orden de servicio**

El proceso de solicitud de órdenes de servicio se puede realizar a través de varios mecanismos de comunicación, siendo los principales: vía telefónica o fax, vía email o a través de un sistema online. Algunas compañías requieren que el cliente envíe la petición de orden de servicio vía fax, email o Internet, mientras que otras les permiten llamar a un número telefónico para realizar la solicitud (Grace, 2012). Es decir, las solicitudes de servicio pueden ser planteadas por el cliente de forma oral o escrita. Una vez registradas de manera formal en el sistema, se convierten en órdenes de servicio.

Corrigo (2013) establece que el ciclo típico de una orden de servicio continúa luego de la siguiente manera:

- Creación

La petición de servicio original contiene una descripción básica del problema. El personal encargado de la empresa de servicio entra en contacto con el cliente en cuestión para verificar el problema planteado y formalizar con lujo de detalle la petición, generándose la correspondiente orden de servicio en su etapa inicial. En esta etapa el estatus de la orden aparece como Nueva.

- Planificación

La orden de servicio debe ser revisada en detalle en cuanto a parámetros como prioridad, requerimientos/disponibilidad de partes, tareas específicas que detallan el trabajo a realizar, estimación de fechas, tiempo requerido y disponibilidad de horas. A continuación, se procede a su aprobación y asignación al personal técnico encargado de ejecutar la orden de servicio. Al culminar esta etapa el estatus de la orden aparece como Asignada.

- Apertura

Una vez asignada la orden de servicio y se comenzado el trabajo, la misma puede cambiar entre varios estados:

*Abierta: En Progreso:* Se arranca la orden cuando se empieza a realizar la labor asignada.

*Abierta: En Pausa:* Se puede poner en pausa cualquier orden de servicio que esté en progreso y luego retomarla nuevamente.

*Abierta: En Espera:* Se podría poner una orden en espera por partes o aprobación del cliente.

Mientras el trabajo especificado en la orden de servicio esté en ejecución, se pueden recoger datos adicionales y añadirse a la orden. Se pueden añadir también nuevas partes o materiales utilizados para la realización del trabajo.

- Cierre

Cuando el trabajo especificado en la orden de servicio ha sido terminado se cambia el estatus de la misma a Completada.

Cuando se crea una orden de servicio, se puede estimar el tiempo requerido para el trabajo y los materiales que probablemente se usarán; cuando se cierra la orden de servicio, se registra el tiempo y los materiales que realmente se usaron.

Además, se podrían asignar códigos de cierre a cada orden de servicio. Por ejemplo, los códigos de cierre podrían estar basados en el tipo de problema presentado o en la acción ejecutada para resolver el problema (reparación, reemplazo, ajuste, entre otros).

- Facturación

La información de cada orden de servicio completada es revisada y aprobada, generándose su correspondiente factura. El estatus cambia de Completada a Facturada.

Es importante mencionar que las órdenes de servicio pueden ser canceladas a partir de cualquier estatus antes de ser facturadas. En este caso, adquieren el estatus Cancelada. (Véase

Tabla 1)

**Tabla 1. Estados del ciclo de vida de una orden de servicio**

<b>Acción:</b>	<b>Estado Resultante:</b>	<b>Notas:</b>
Crear una orden de servicio	<i>Nueva</i>	Estatus inicial para toda orden de servicio
Asignar una orden de servicio	<i>Asignada</i>	La orden ha sido revisada, aprobada y asignada al técnico de servicio
Arrancar una orden de servicio	<i>Abierta: En Progreso</i>	La orden ha sido comenzada por el técnico de servicio
Colocar una orden en pausa	<i>Abierta: En Pausa</i>	Se realiza una breve pausa sobre el tiempo de ejecución de la orden
Colocar una orden en espera	<i>Abierta: En Espera</i>	La orden queda en espera por una aprobación o algún otro evento
Completar una orden de servicio	<i>Completada</i>	Se han completado todas las tareas descritas en la orden
Facturar una orden de servicio	<i>Facturada</i>	Se genera la factura correspondiente a la orden de servicio
Cancelar una orden de servicio	<i>Cancelada</i>	La orden se cancela a partir de algún otro estado antes de facturar

**Fuente:** Elaboración por los autores

#### **2.1.4.- Sistema de gestión de órdenes de Servicio**

Un sistema de gestión de órdenes de servicio es un sistema que administra el ciclo de vida de la orden de servicio de principio a fin, permitiendo a los encargados crear, mantener y gestionar los datos importantes contenidos en dichas órdenes. Estos sistemas recopilan la data de forma centralizada para que pueda ser compartida y vista por personal de la organización e, incluso, los clientes.

Entre las capacidades más importantes que menciona Find Accounting Software (2014) que de los sistemas de orden servicio se tienen:

*Creación de órdenes y solicitudes de órdenes de servicio.* En su nivel más básico, el software para órdenes de servicio permite crear un documento rastreable identificando qué trabajo se debería hacer y cómo se debería cumplir. Plantillas y herramientas para el registro de órdenes guiadas puede mejorar considerablemente la habilidad de los empleados para la ejecución rápida y certera de las tareas asignadas.

*Planificación y asignación de recursos.* Determinar quién realiza el trabajo, qué se requiere para completarlo y cuando se debería hacer, es una tarea importante dentro la gestión de órdenes de servicio. Las soluciones de gestión de órdenes de servicio pueden mantener una base de datos de información que identifique qué equipo y materiales se requieren para completar cada tipo de trabajo. Similarmente, un programa de órdenes de servicio permite identificar qué empleados están calificados para qué tipo de trabajo. Adicionalmente, los programas de gestión de órdenes de servicio ofrecen una variedad de herramientas para gestionar la planificación y asignación de recursos.

*Cambio de órdenes.* Las órdenes de servicio no son siempre documentos estáticos. A menudo pueden ocurrir cambios en el alcance del servicio u otros datos después de que la orden de servicio ha sido creada.

*Actualizaciones de estatus y monitoreo de culminación.* Una parte importante del valor de una solución de gestión de órdenes de servicio es la visibilidad de la culminación de trabajos. Ser capaz de predecir de forma certera la culminación del servicio puede ser una parte crítica para asegurar la satisfacción del cliente.

*Generación de reportes.* Un sistema de órdenes de servicio provee una fuente abundante de data sobre la eficiencia operacional de la empresa. Las herramientas de reporte en el sistema de orden de servicio ayudan a convertir esta data en herramienta accionable para mejorar los procesos.

### **2.1.5.- Beneficios de los sistemas de órdenes de servicios**

La gestión de órdenes de servicio puede ayudar a lograr una variedad de metas organizacionales relacionadas con la mejora del servicio al cliente y el incremento de la eficiencia operacional, incluyendo:

*Tiempos promedios más rápidos.* Un sistema de orden de servicio robusto puede reducir los tiempos de ejecución, mejorando la satisfacción del cliente con la culminación de trabajo más rápido. En primer lugar, un sistema de orden de servicio que promueve la visibilidad dentro de los procesos puede ayudar a detectar oportunidades para completar trabajo más rápidamente. En segundo lugar, un sistema de orden de servicio intuitivo y fácil de usar reduce la sobrecarga administrativa relacionada con el procesamiento de órdenes de servicio.

*Tiempo de inactividad reducido.* Un pobre manejo de órdenes de servicio puede llevar al solapamiento de demandas de equipo y empleados en algunos momentos y recursos sin actividad en otros. La habilidad de manejar fácil y claramente la planificación de recursos permite la optimización de los recursos y la reducción del tiempo de inactividad.

*Actualizaciones de tiempo más precisas.* La predicción certera de los tiempos de los trabajos basada tanto en información histórica y la disponibilidad de recursos en tiempo real significa menos instancias de clientes decepcionados porque se sobrepasan los tiempos.

*Añaden eficiencia.* El software de gestión de orden de servicio basado en Internet hace el proceso completo de orden de servicio más eficiente y económico, comparado con los procesos manuales, debido a que los clientes pueden reportar sus incidencias a la compañía en línea. El proceso a menudo requiere que el cliente cree un ticket de problema, el cual es una simple solicitud de servicio. Los técnicos pueden revisar el ticket y llegar a una solución sin tener que visitar físicamente al cliente y por lo tanto evitar perder tiempo en tareas triviales. Los clientes además pueden usar los sistemas de orden de servicio para monitorear el estatus o tiempo utilizado para resolver sus problemas. (Find Accounting Software, 2014)

#### **2.1.6.- Tecnología móvil en órdenes de servicio**

Las órdenes de servicio son el elemento vital de toda organización de servicio. Desde la solicitud del cliente hasta la entrega de la factura al cliente, es crítico que la información de la orden de servicio sea certera y distribuida de manera oportuna a cada miembro del proceso de gestión de la orden.

Tomando ventaja del auge que vienen presentando las tecnologías móviles desde hace algún tiempo, se hace imprescindible la adopción del acceso móvil y remoto como parte integral de un sistema de órdenes de servicio. Si la compañía ejecuta trabajo de campo, la habilidad para acceder remotamente a la información de la orden de servicio es crítica. A menudo esto significa el acceso a la data vía dispositivos móviles como teléfonos inteligentes y tabletas. Para satisfacer esta necesidad, muchos proveedores han optimizado su solución para el acceso web móvil o han desarrollado aplicaciones móviles nativas para extender funcionalidades importantes a usuarios móviles.

### **2.1.7.- Casos de estudio de Sistemas de Gestión de órdenes de servicio.**

En la actualidad son muchas las opciones disponibles como software para la gestión de órdenes de servicio. El auge de las tecnologías móviles ha permitido la evolución de estos sistemas de gestión permitiendo aumentar el nivel de productividad y eficiencia de las organizaciones que decidan adquirir estos servicios. Además de contar con funcionalidades que brinden total soporte al ciclo de vida de la orden de servicio, estos paquetes de software incluyen aplicaciones web nativas para los dos sistemas operativos móviles más utilizados, Android e iOS, brindando un toque innovador y aumentando el alcance máximo de posibles clientes.

Algunos de estos sistemas son mencionados a continuación:

- Services por itsoft: software de pago el cual consta de un sistema web y una aplicación móvil para gestionar las órdenes de servicio. La aplicación móvil es compatible con el sistema operativo Android.
- Gestión servicio técnico por GB Systems: software de pago basado en tecnologías web que permite manejar cualquier tipo de empresa que brinde Servicio Técnico de cualquier clase. No posee aplicación móvil.
- Synchroteam: software de pago para la gestión de órdenes de servicio. Es una aplicación móvil compatible tanto con el sistema operativo Android como con iOS.

Son muchas las soluciones móviles en la actualidad pero todas poseen dos desventajas claras frente a la solución propuesta en el presente trabajo, son opciones de pago y las aplicaciones móviles que brindan no son multiplataforma. De allí la justificación para la realización de una solución móvil que brinde soporte a la gestión de órdenes de servicio que sea multiplataforma a través de una aplicación móvil híbrida, reduciendo los costos de desarrollo haciéndola más accesible para pequeñas empresas y a su vez tener un mayor alcance de posibles usuarios finales.

## **2.2.- Sistemas de información**

Según Luque (2009) un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de generar información que se pueda utilizar posteriormente. Un sistema de información realiza cuatro actividades básicas:

**Entrada de información:** Es el proceso mediante el cual el sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

**Almacenamiento de información:** Es la capacidad del sistema de información de guardar o recordar información que va a hacer utilizada posteriormente. A través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

**Procesamiento de Información:** Es la capacidad del sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección.

**Salida de Información:** La salida es la capacidad de un sistema de Información para exportar la información procesada o bien datos de entrada al exterior. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.

A continuación se listan los distintos tipos de sistemas de información mencionados por Luque (2009).

- Sistemas transaccionales.
- Sistema de apoyo a la toma de decisiones.
- Sistemas estratégicos.
- Sistemas de información formal.
- Sistemas de información informal.
- Sistemas de comunicación.
- Sistemas de información organizacional.
- Sistemas de información operativos.
- Sistemas de información gerencial.
- Sistema de procesamiento de datos.

Para efectos de esta investigación, se puede catalogar la solución propuesta como un sistema de información de comunicación, debido a que estos tienen como principal función, transmitir información entre los diferentes subsistemas del sistema padre. Lo cual se adapta perfectamente a nuestro caso de estudio, el cual intenta ofrecer la información publicada a través de una aplicación web administrativa en una aplicación móvil.

### **2.3.- Aplicaciones móviles.**

De acuerdo con Adolph (2009) las aplicaciones móviles son software complementario para dispositivos portátiles de mano, tales como teléfonos inteligentes y asistentes de datos personales (PDA). Entre las más populares están los juegos, redes sociales, mapas, noticias, negocio, clima e información de viaje. Todas ellas promueven al menos una de las características técnicas del dispositivo: interfaces de comunicación, procesadores de audio y video, cámara, sensores o módulos de posicionamiento (GPS).

#### **2.3.2.- Tipos de aplicaciones móviles**

Geospatial (2013) señala que en la mayor parte de las ocasiones se relacionan desarrollos móviles con aplicaciones nativas. Este tipo de desarrollos son

desarrollos hechos de forma específica para un determinado sistema operativo.

Por otro lado, se encuentran las aplicaciones web móviles, que no son más que aplicaciones que corren sobre un navegador web que, en este caso, es el navegador web del dispositivo móvil.

También existen lo que se conoce como aplicaciones híbridas. Este tipo de aplicaciones son una combinación de ambas: Nativas + Web.

### ***2.3.2.1.- Aplicaciones móviles nativas.***

Este tipo de aplicaciones están hechas para ejecutarse en un dispositivo y sistema operativo específico. Se crean utilizando distintos tipos de lenguajes, ejecutándose de forma más eficiente sobre los dispositivos móviles ya que sus componentes están diseñados de forma específica para el sistema operativo. Además, este tipo de aplicaciones pueden emplear todos los sensores y elementos del teléfono: cámara, sistema de posicionamiento global (GPS), acelerómetro, agenda, entre otros. Esta es una diferencia fundamental con respecto a las aplicaciones web.

El código fuente de estas aplicaciones se escribe en función del dispositivo para el que se desarrolle. Este código fuente se compila a un ejecutable. Es un proceso similar al de las aplicaciones tradicionales de escritorio. Todos aquellos recursos (imágenes, iconos, entre otros.) que la aplicación necesita para ejecutarse quedan en el archivo compilado. Este archivo está ya listo para ser distribuido y subido a las tiendas en línea de aplicaciones específicas del dispositivo para el que se trabaja. Una vez subido el ejecutable, las tiendas de aplicaciones tienen un proceso de auditoría de la aplicación para evaluar si se adecúa a los requerimientos del sistema.

- **Ventajas de las aplicaciones móviles nativas:**
  - Posibilidad de utilizar funcionalidades nativas: siempre se puede utilizar funcionalidades nativas, como la cámara, el acelerómetro, entre otros.
  - Mejor desempeño: típicamente llegan a desempeñarse mejor que las aplicaciones web.
  - Hay un mercado para estas aplicaciones: se pueden encontrar las aplicaciones puestas para la venta o incluso gratuitas utilizando las tiendas de aplicaciones.
  - Modo desconectado: la aplicación puede utilizarse sin necesidad de una conexión a internet; no obstante, esto está limitado por el comportamiento en particular de la aplicación.
- **Desventajas de las aplicaciones nativas:**
  - Generalmente son más costosas de implementar: se requiere una inversión en tiempo y recursos mucho mayor.
  - Mayores costos de mantenimiento: de igual forma, mantener varias versiones de la aplicación es costoso.
  - Actualizaciones y aprobaciones: publicar una aplicación en la tienda en línea es tedioso, puede llevar hasta más de una semana. De igual forma, si se tiene una nueva funcionalidad o se descubrió un error que debe ser arreglado, se debe esperar otra semana para se aprueben los cambios.

#### **2.3.2.2.- Aplicaciones web móviles**

Las aplicaciones web móviles, a diferencia de las aplicaciones nativas, se ejecutan dentro del navegador del teléfono. Por ejemplo, en la plataforma iOS, se ejecutan en el navegador Safari. Estas aplicaciones están desarrolladas con HTML, CSS y Javascript. (Ver Figura 3)

- **Ventajas de las aplicaciones web móviles:**
  - Compatibilidad: se reutiliza casi en un 100% el mismo código fuente para todos los sistemas operativos orientados a móviles, así se tiene que dar mantenimiento sólo a una versión.
  - Rango de usuarios: se puede llegar a más tipos de dispositivos.
  - Fácil uso: no hace falta descargar nada, sólo con acceder a una URL los usuarios podrán utilizar la aplicación.
  - Actualizado siempre: las actualizaciones llegan de inmediato, no se debe esperar una semana para que la tienda de aplicaciones apruebe una nueva actualización.
  - Más sencillo y rápido de llevar a cabo en muchas ocasiones: muchas veces, es mucho más fácil desarrollar y diseñar dentro de un browser utilizando HTML, CSS y JavaScript que hacer lo mismo en Java, Objective-C, por ejemplo. Posicionar elementos, estilizarlos y ciertas flexibilidades son posibles en minutos con tecnologías web.
- **Desventajas de las aplicaciones web móviles:**
  - No es posible utilizar la cámara y otros componentes nativos (al menos por ahora): Con aplicaciones web móviles, al ser accedidas desde el navegador, no se tiene disponibilidad de ciertos componentes y funcionalidades nativas del teléfono, entre ellos la cámara.
  - Posibilidad de usarla en modo desconectado (en algunos): No es posible utilizar el modo desconectado cuando se desarrolla con web.
  - Además, estas aplicaciones no se pueden vender en las tiendas en línea de aplicaciones.



Figura 3. Aplicaciones móviles: nativas vs web (Eternal Software, 2014)

### 2.3.2.3.- Aplicaciones móviles híbridas

Las aplicaciones híbridas reúnen lo mejor de los dos modelos anteriores. Este tipo de aplicaciones permite el uso de tecnologías multiplataforma como HTML, Javascript y CSS y, al mismo tiempo, permiten acceder a buena parte de los dispositivos y sensores del teléfono. Buena parte de la infraestructura es tipo web y la comunicación con los elementos del teléfono se hace mediante marcos de trabajo como PhoneGap.

Al igual que para las aplicaciones nativas, el código una vez creado se compila a un ejecutable. Además, al igual que en las aplicaciones Web, se genera código HTML, CSS y Javascript para ejecutar en un navegador. Las páginas HTML (totalmente o en parte) pueden provenir de cualquier servidor web o residir en el propio dispositivo. De esta forma disponemos de lo mejor de cada arquitectura. Por ejemplo, una aplicación híbrida puede presentar en pantalla una página HTML5 sobre la que se muestra la posición actual proporcionada por el GPS. El coste de desarrollo es similar al de las aplicaciones web y nos permite preparar las aplicaciones para trabajar sin necesidad de estar conectadas al servidor, es decir offline. Básicamente todos los proveedores de

entornos o frameworks para aplicaciones híbridas han seguido el siguiente modelo para diseñar el entorno de ejecución:

En la capa inferior se encuentra un Contenedor de Dispositivo (aplicación nativa). Éste proporciona a las capas superiores acceso a las funciones del dispositivo (cámara, acelerómetro, entre otros.) y una interfaz común para poder abstraer esta capa y hacer a las demás independientes del dispositivo. El creador del framework deberá construir este contenedor para cada sistema operativo al que desee dar soporte.

Sobre el Contenedor de Dispositivo se encuentra el Contenedor Web, que en esencia se trata de un Navegador Web con capacidades para interpretar HTML y ejecutar JavaScript. Aquí el acceso a las funciones nativas (cámara, acelerómetro, entre otros.) está disponible a través de funciones Javascript o lenguaje de marcas, dependiendo de las tecnologías ofrecidas por el framework. Estas dos primeras capas configuran la arquitectura mínima necesaria para ejecutar aplicaciones en los dispositivos móviles actuales, por tanto, éste debe ser el punto de partida al evaluar qué capacidades proporciona cada framework. (Ver Figura 4)

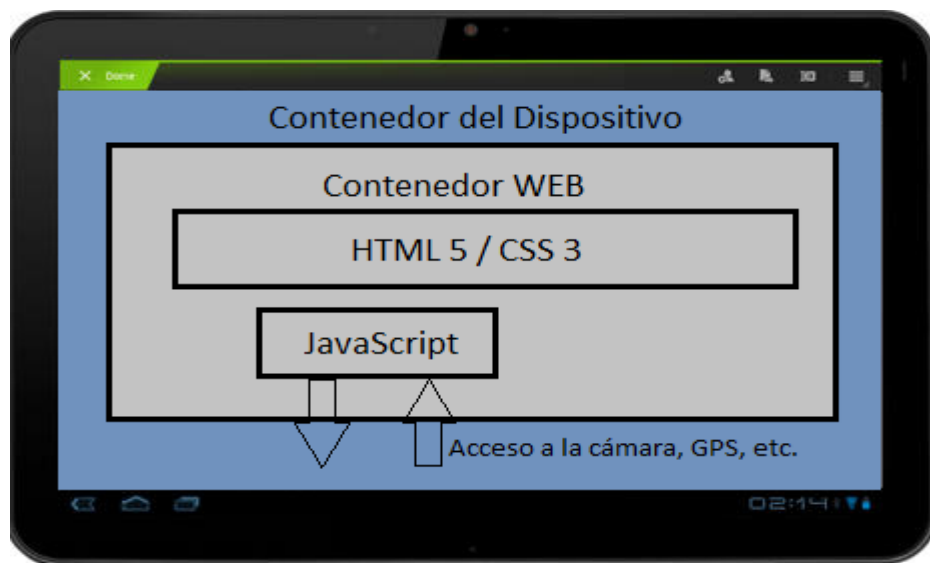


Figura 4. Arquitectura de una aplicación móvil híbrida (Garzo, 2012)

## **2.4.- Dispositivos móviles**

Un dispositivo móvil es un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales (Baz, Ferreira, Álvarez, & García). De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los sistemas de posicionamiento global, pasando por los teléfonos móviles. (Ver Figura 5)

### **2.4.1.- Clasificación de los dispositivos móviles**

Según Aranaz (2009) los dispositivos móviles pueden ser clasificados en los siguientes grupos:

- **Dispositivo de comunicación**

Un dispositivo de comunicación es aquel dispositivo móvil cuyo cometido principal es ofrecer una infraestructura de comunicación, principalmente telefónica. Estos dispositivos ofrecen además servicios como el envío de mensajes de texto y multimedia. En esta categoría se incluiría el tradicional teléfono móvil, precursor indiscutible dentro de los dispositivos móviles, y el teléfono inteligente, que amplía considerablemente las prestaciones del primero mediante pantalla táctil, conexión a Internet o la ejecución de aplicaciones.

- **Dispositivo de computación**

Los dispositivos de computación son aquellos dispositivos móviles que ofrecen mayores capacidades de procesamiento de datos y cuentan con una pantalla y teclado más cercanos a un ordenador de escritorio. Dentro de este grupo encontramos a los Asistentes Digitales Personales (PDA), muy populares a

finales de los años 90 y que permitían al usuario disponer de un organizador mucho más completo que los ofrecidos por los teléfonos móviles del momento, e incluso en ocasiones la visualización de documentos o acceso a Internet. Por otro lado, dispositivo de computación también es un ordenador portátil o laptop, que dentro de los dispositivos móviles son sin duda los que mayores prestaciones de hardware ofrecen (igualando o superando a los de escritorio) pero también los que tienen un mayor tamaño, peso y precio. Las calculadoras gráficas pueden ser igualmente incluidas en este grupo de dispositivos de computación.

- **Reproductor multimedia:**

Un reproductor multimedia es aquel dispositivo móvil que ha sido específicamente diseñado para proporcionar al usuario la reproducción de uno o varios formatos de datos digitales, ya sea audio, vídeo o imágenes. Dentro de estos dispositivos encontramos reproductores de MP3, los DVD portátiles y en los últimos años los reproductores multimedia de la popular familia iPod de Apple, que ofrecen tanto audio como vídeo. Estos dispositivos son con frecuencia los de más reducido tamaño y, junto a los teléfonos móviles tradicionales e inteligentes, los más extendidos.

- **Grabador multimedia**

Dentro de los dispositivos móviles, un grabador multimedia es aquel dispositivo que posibilita la grabación de datos en un determinado formato digital, principalmente de audio y vídeo. En esta categoría se hallan las cámaras fotográficas digitales o las cámaras de vídeo digital.

- **Consola portátil**

Una consola portátil es un dispositivo móvil cuya única función es la de proporcionar al usuario una plataforma de juego. Las consolas portátiles fueron, junto a los teléfonos, los primeros dispositivos móviles en convertirse

en un producto de masas. Hoy en día representan un importantísimo volumen de ventas dada su gran aceptación en la sociedad y son objeto de auténticas guerras comerciales entre las principales compañías del sector.

- **Teléfono inteligente**

Dentro de los dispositivos móviles, un teléfono inteligente es una evolución del teléfono móvil tradicional que cuenta con ciertas características y prestaciones que lo acercan más a un ordenador personal que a un teléfono tradicional.

Entre dichas características, se puede encontrar una mejora en la capacidad de proceso y almacenamiento de datos, conexión a Internet de forma inalámbrica, pantalla táctil, acelerómetro, sistema de posicionamiento global (GPS) y diversas aplicaciones de usuario como navegador web, cliente de correo, aplicaciones ofimáticas, reproductores de vídeo y audio, incluyendo la posibilidad de descargar e instalar otras nuevas.

A pesar de estas importantes mejoras con respecto a sus predecesores móviles, el reducido tamaño de los teléfonos inteligentes conlleva inexorablemente limitaciones de hardware que los mantienen claramente diferenciados de los ordenadores convencionales. Estas limitaciones se reflejan principalmente en pantallas más pequeñas, menor capacidad del procesador, restricciones de memoria RAM y memoria persistente, y necesidad de adaptar el consumo de energía a la capacidad de una pequeña batería.

Estas limitaciones obligan a tener muy en cuenta la capacidad real del dispositivo a la hora de desarrollar su software, ya sean aplicaciones de usuario o el propio sistema operativo.



**Figura 5. Ejemplos de dispositivos móviles: laptop, tableta y teléfono inteligente (Cruz, 2013)**

## **2.5.- Sistemas operativos móviles**

Un Sistema Operativo (SO) es el software básico de una computadora que provee una interfaz entre el resto de programas del computador, los dispositivos hardware y el usuario. Las funciones básicas del Sistema Operativo son administrar los recursos de la máquina, coordinar el hardware y organizar archivos y directorios en dispositivos de almacenamiento (Vásquez, 2011).

En su trabajo de investigación Pedrozo (2012) afirma que un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil, al igual que las computadoras más grandes utilizan Windows, Linux o Mac OS, entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

### **2.5.1.- Componentes**

De acuerdo con Pedrozo (2012) al igual que los sistemas operativos presentes en los componentes informáticos más grandes, los sistemas operativos móviles en general cuentan con unas capas específicas, sin embargo en algunos sistemas operativos esto es diferente de acuerdo a su funcionamiento.

En la Figura 6 se puede observar la forma en que está estructurado un sistema operativo móvil.



**Figura 6. Estructura de un sistema operativo móvil (Android vs iOS, 2013)**

Las capas o componentes se describen de la siguiente manera:

- **Kernel:** Una de las más importantes y esenciales piezas que componen cualquier sistema operativo, sea móvil o de ordenador, es el denominado núcleo o Kernel, el cual es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman el dispositivo móvil. También es el encargado de brindar diferentes servicios a las capas superiores como los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso y administración de la memoria del sistema. Los sistemas operativos para móviles pueden basarse en núcleos Linux, tal como lo hace Android, o inclusive iOS, el sistema operativo del iPhone que utiliza un kernel heredado de Unix.

- **Middleware:** Esta capa es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas. Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento. El Middleware brinda la posibilidad de ejecutar servicios

muy importantes para que otras aplicaciones, en capas superiores de la jerarquía, puedan ejecutarse. Estos servicios son vitales para el normal funcionamiento de la estructura de un sistema operativo móvil. Entre los servicios que presta esta capa están los motores de comunicaciones y mensajería, funciones de seguridad, intérpretes de páginas web, además de soporte para una gran variedad de servicios concernientes al apartado multimedia que es capaz de ejecutar el móvil.

- **Entorno de ejecución de aplicaciones:** Esta capa provee de todos los elementos necesarios para la creación y desarrollo de software a los programadores, es decir, contiene elementos que serán de gran ayuda al momento de escribir aplicaciones compatibles con determinado sistema operativo. Entre los servicios que los programadores pueden encontrar, se destacan un gestor de aplicaciones y una serie de interfaces de programación de aplicaciones (API).

- **Interfaz de usuario:** La interfaz de usuario es el elemento del teléfono que usualmente se utiliza para interactuar con el aparato. Sin esta capa no sería posible utilizar el dispositivo, ya que la misma presenta todos los elementos necesarios para facilitar cualquier tipo de tarea que se desee realizar en la terminal. Además incluye todos los elementos gráficos que hacen posible el uso cómodo y sencillo del móvil: botones, menús, pantallas y listas, entre otros.

Por otra parte, también coexisten en el teléfono una serie de aplicaciones que son nativas del móvil, y que se encargan de tareas tales como menús, marcador de números de teléfono y demás.

Una de las particularidades más importantes incorporadas a la interfaz gráfica de usuario en los últimos años ha sido, sin lugar a dudas, la posibilidad de utilizar todas las funciones del teléfono mediante el uso de los dedos de la mano, desplazando a los botones físicos a un segundo plano en cuanto al manejo de opciones y funciones en el móvil. También una gran ventaja son

los teclados del tipo virtual, que ya están reemplazando a los dispositivos de entrada tradicionales, permitiendo de esta manera un mejor y más ligero diseño del móvil.

### **2.5.2.- Principales sistemas operativos móviles**

El mercado de los dispositivos móviles es uno de los mercados más dinámicos dentro de la electrónica de consumo. El ciclo de cambio de producto es bastante más corto y la evolución de las tecnologías asociadas es mucho más acelerada. Hoy en día existen varios sistemas operativos móviles en el mercado, pero hablar de ellos es hablar en primera instancia de los grandes dominantes del mercado en la actualidad: iOS y Android (Celularis, 2013).

A continuación, Pedrozo (2012) realiza una descripción detallada de los sistemas operativos Android e iOS:

- **Android:**

Es un sistema operativo móvil basado en Linux, desarrollado por la Open Handset Alliance, la cual es liderada por Google. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El hecho de que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado dispositivo móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), capacidad del procesador, entre otros.

Google Play es la tienda en línea de distribución de aplicaciones desarrollada por Google para dispositivos Android. Ésta es una aplicación que se encuentra instalada en la mayoría de los dispositivos Android y permite a los usuarios navegar y descargar aplicaciones publicadas por los desarrolladores. Google retribuye a los desarrolladores el 70% del precio de las aplicaciones. Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

Comparado con el iOS, Android es un sistema más vulnerable, debido principalmente a que el proceso de certificación de aplicaciones es menos riguroso que el de Apple y el esquema de gestión de permisos para éstas, que pone muchas decisiones de seguridad importantes en manos del usuario, lo expone a mayores riesgos y le hace más atractivo a los ataques de ingeniería social.

En cuanto a la arquitectura del sistema operativo Android, la misma está formada por cuatro capas que facilitan al desarrollador la creación de aplicaciones. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores. (Ver Figura 7).

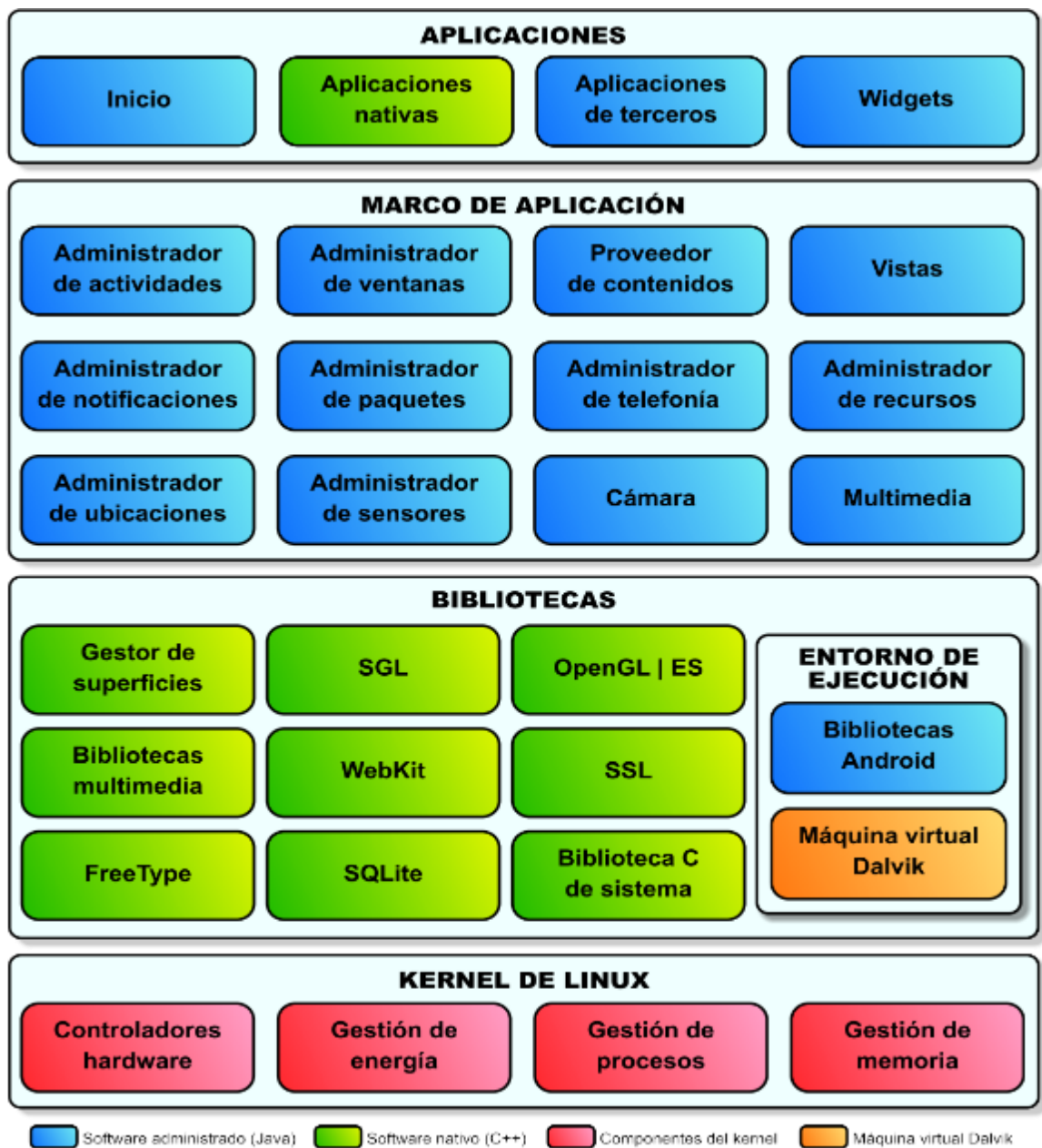


Figura 7. Arquitectura del sistema operativo Android (Android OS: Arquitectura, 2012)

Esta distribución permite acceder a las capas más bajas mediante el uso de librerías para que así el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware de los teléfonos. A continuación, se describe de manera más detallada la estructura del sistema operativo Android:

- **Aplicaciones**

Las aplicaciones base incluyen un cliente de correo electrónico, programa de mensajería de texto, calendario, mapas, navegador, contactos y otros.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java y utilizando el kit de desarrollo de software (SDK) de Android, pero están disponibles otras herramientas de desarrollo, incluyendo un kit de desarrollo nativo para aplicaciones o extensiones en C. El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software provisto por Google el cual se puede descargar gratuitamente.

- **Marco de trabajo de aplicación**

Los desarrolladores tienen acceso completo a las interfaces de programación de aplicaciones (API) del framework usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework).

- **Librerías o bibliotecas**

Android incluye un conjunto de librerías de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: librería C del sistema (implementación de librería C estándar), librerías de medios, librerías de gráficos 3D y librerías de SQLite, entre otras.

- **Entorno de ejecución**

Android incluye un set de librerías base que proporcionan la mayor parte de las funciones disponibles en las librerías base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La máquina virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta integrada "dx".

- **Núcleo Linux**

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

- **iOS**

Es el sistema operativo móvil del fabricante Apple Inc. Originalmente desarrollado para el iPhone, siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple Inc. no permite la instalación de iOS en hardware de terceros.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa. Los elementos de control consisten en deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslizamientos, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por

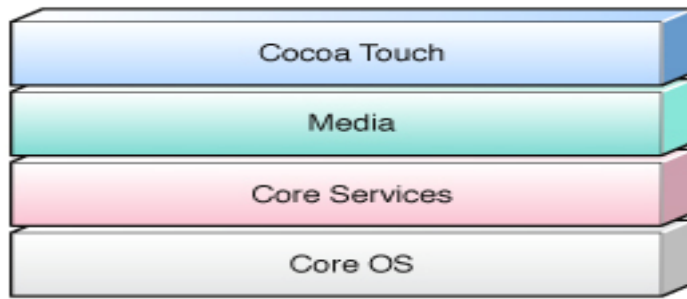
ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

La pantalla principal (llamada *SpringBoard*) es donde se ubican los íconos de las aplicaciones y el Dock en la parte inferior de la pantalla donde se pueden anclar aplicaciones de uso frecuente. La pantalla tiene una barra de estado en la parte superior para mostrar datos, tales como la hora, el nivel de batería, y la intensidad de la señal. El resto de la pantalla está dedicado a la aplicación actual. Desde iOS 3, se puede acceder a la búsqueda a la izquierda en la pantalla inicial, permitiendo buscar música, videos, aplicaciones, correos electrónicos, contactos y archivos similares.

Antes de iOS 4, la multitarea estaba reservada para aplicaciones por defecto del sistema. A Apple le preocupaba los problemas de batería y rendimiento si se permitiese correr varias aplicaciones de terceros al mismo tiempo. A partir de iOS 4, dispositivos de tercera generación y posteriores permiten el uso de 7 interfaces de programación de aplicaciones (API) para multitarea, específicamente:

- Audio en segundo plano
- Voz IP
- Localización en segundo plano
- Notificaciones *push*
- Notificaciones locales
- Completado de tareas
- Cambio rápido de aplicaciones

El iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix. Cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de Servicios Principales, la capa de Medios y la capa de Cocoa Touch. (Ver Figura 8)



**Figura 8. Arquitectura del sistema operativo iOS (Android vs iOS, 2013)**

El sistema operativo iOS no permite Adobe Flash ni Java. Steve Jobs escribió una carta abierta donde critica a Flash por ser inseguro, con errores, consumir mucha batería, ser incompatible con interfaces multitáctil e interferir con el servicio *App Store*. En cambio iOS usa HTML5 como una alternativa a Flash. Esta ha sido una característica muy criticada tanto en su momento como en la actualidad. Sin embargo, por métodos extraoficiales se le puede implementar aunque conllevaría la pérdida de la garantía.

## **2.6.- PhoneGap**

“PhoneGap es un framework de código abierto para construir aplicaciones nativas multiplataforma, utilizando tecnologías web estándar como HTML, CSS y Javascript. Este tipo de aplicación móvil es llamado aplicación híbrida” (Wargo, 2012). PhoneGap resulta ser entonces una especie de “envoltorio” utilizado para empaquetar de forma nativa aplicaciones web móviles. (Ver Figura 9)

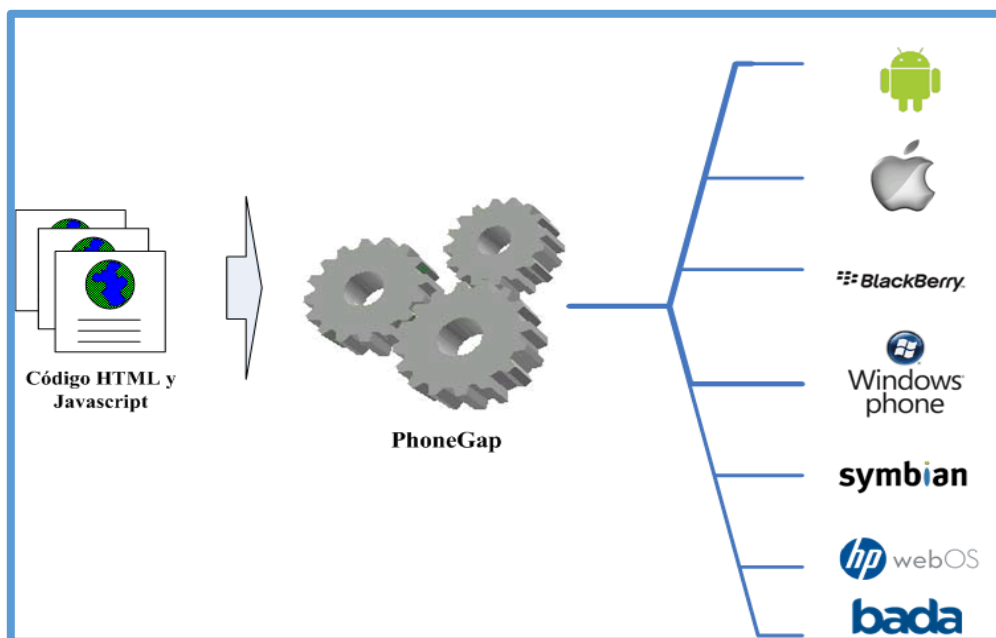


Figura 9. Funcionamiento de PhoneGap (Asociación javaHispano, 2012)

### 2.6.1.- Historia

El comienzo de PhoneGap se remonta al evento iPhoneDevCamp 2008 organizado por la empresa Nitobi. Se arrancó con la creación de la funcionalidad principal más el contenedor de aplicación nativo necesario para interpretar el contenido de la aplicación web en el iPhone. Después de la compilación inicial del framework, el proyecto se extendió para dar soporte a Android y luego a BlackBerry. Con el transcurrir del tiempo, PhoneGap ha ido añadiendo soporte para otras plataformas (Wargo, 2012).

En 2011, Adobe Corporation adquiere Nitobi. Luego, libera la tecnología PhoneGap como código abierto bajo el nombre de Apache Cordova (Wargo, 2012).

### 2.6.2.- Retos en el Desarrollo de Aplicaciones Multiplataforma

Dado el volumen del mercado móvil, se ha vuelto una experiencia difícil para los desarrolladores llegar a la mayor audiencia. En la Tabla 2 se puede apreciar que existen múltiples plataformas como Android, iPhone, BlackBerry,

Windows Phone, y cada una de ellas utiliza distintos lenguajes de programación y procesos de distribución. Por lo tanto, ello implica que el desarrollo de una aplicación requiera habilidades por separado para cada plataforma. (Bibhudatta, 2013)

**Tabla 2. Lenguajes, Herramientas y Formatos de Paquetes de las principales plataformas móviles.**

	<b>iOS</b>	<b>Android</b>	<b>BlackBerry</b>	<b>Windows</b>
<b>Lenguaje</b>	Objective C	Java	Java	C#, .NET.
<b>Herramienta</b>	Xcode	Android SDK	BlackBerry SDK	Windows Dev. Tools
<b>Formato de empaquetado</b>	.app	.apk	.cod	.xap
<b>App Store</b>	Apple App Store	Google Play	BlackBerry App World	Windows Phone Store

Fuente: (Bibhudatta, 2013)

Los retos de hoy en día para mantener una aplicación en distintas plataformas son descritos por Bibhudatta (2013) de la siguiente manera:

- Los desarrolladores requieren de distintas habilidades, dado que las herramientas y lenguajes de programación son diferentes para las distintas plataformas.
- Desarrollar una aplicación para distintas plataformas implica costos asociados.
- Debido a que el desarrollo es diferente para las distintas plataformas, se tendrá código fuente intrincado, dando como resultado un código base confuso y difícil de mantener.
- Se incrementan los tiempos hasta la publicación en el mercado de aplicaciones y se hace necesario equipos de desarrollo más grandes, dado que se necesitan desarrollos en paralelo.

### **2.6.3.- Cómo funciona PhoneGap**

De acuerdo con Wargo (2012) el desarrollador construye una aplicación web para el dispositivo móvil, y herramientas especiales de PhoneGap empaquetan la aplicación web en una aplicación móvil nativa para cada plataforma móvil soportada.

Dentro de la aplicación nativa, la interfaz de usuario de la aplicación consta principalmente de una sola pantalla que contiene un Web View que consume todo el espacio disponible en la pantalla del dispositivo. Un Web View es un componente de aplicación nativa que se usa para reproducir contenido web (típicamente páginas HTML) dentro de una pantalla o ventana de aplicación nativa. Es esencialmente un componente envoltorio accesible programáticamente alrededor del navegador incluido en el dispositivo móvil. A medida que el usuario interactúa con el contenido de la aplicación, el código Javascript puede cargar otro contenido desde los archivos de recursos empaquetados con esta aplicación o pueden llegar hasta la red y traer contenido desde un servidor web o de aplicación.

La aplicación web corriendo dentro del contenedor es como cualquier otra aplicación web que correría dentro de un navegador web móvil. Puede abrir otras páginas HTML; el Javascript embebido dentro de los archivos fuentes de la aplicación implementa la lógica de aplicación necesaria, ocultando o mostrando el contenido cuando sea necesario dentro de una página, reproduciendo archivos media, abriendo páginas nuevas, ejecutando cálculos y recuperando contenido desde o enviando contenido a un servidor. El estilo y apariencia de la aplicación es determinado por cualquier configuración de fuentes, líneas, espaciado, coloreado o atributos de sombra añadidos directamente a elementos HTML o implementados a través de CSS.

Un navegador web móvil no tiene acceso a componentes tales como la aplicación de Contactos del teléfono y hardware específico del dispositivo

(acelerómetro, cámara, micrófono, entre otros.). La aplicación móvil nativa, por otro lado, puede hacer uso frecuente de esos componentes. PhoneGap provee una suite de interfaces Javascript que un desarrollador puede usar para permitir a una aplicación web ejecutándose dentro del contenedor de aplicación PhoneGap acceder a los componentes nativos del dispositivo que están fuera del contexto web.

Cuando se implementa una funcionalidad en una aplicación que utiliza una de las interfaces de programación de aplicaciones (API) de PhoneGap, la aplicación llama al API utilizando Javascript, y luego una capa especial dentro de la aplicación traduce la llamada al API de PhoneGap al API nativo apropiado para esa funcionalidad en particular. Esto elimina la necesidad para el desarrollador de tener conocimiento detallado de las tecnologías subyacentes, permitiéndole enfocarse en su aplicación y no en cómo conseguir algo en múltiples dispositivos. (Ver Figura 10)

PhoneGap actualmente soporta los siguientes API:

- Acelerómetro
- Cámara
- Captura
- Brújula
- Conexión
- Contactos
- Dispositivo
- Eventos
- Archivo
- Geolocalización
- Media
- Notificación
- Almacenamiento

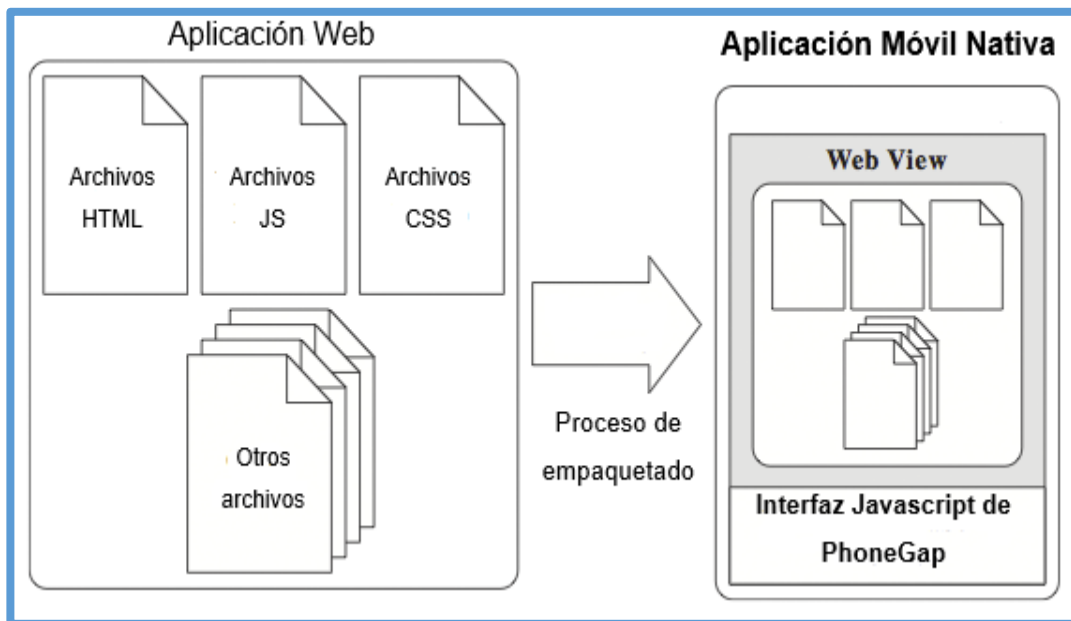


Figura 10. Arquitectura de aplicación PhoneGap (Wargo, 2012)

#### 2.6.4.- Limitaciones de PhoneGap

PhoneGap es un proyecto de código abierto, los voluntarios que trabajan en él son realmente quienes controlan de forma periódica la entrega de nuevas características y solución de errores.

Las entregas planteadas dependen de la suficiente cantidad de recursos, con habilidades notorias para cada plataforma móvil soportada, para hacer el trabajo. Lo que ocurre entonces es que las características y arreglos de errores para las plataformas más populares (como Android e iPhone) obtienen más atención mientras que las menos populares pueden descuidarse.

Como sucede con la mayoría de los proyectos de software de código abierto, existe una cantidad limitada de documentación para muchos tópicos. Incluso aunque la documentación del API es excelente y hay ejemplos de código fuente de la mayoría de las funciones del API, muchas cosas relativas a PhoneGap no están documentadas o no lo están en detalle. Por ejemplo, cuando se entra al sitio web de PhoneGap, rápidamente se puede llegar hasta la documentación del API, pero a excepción de algunas guías de rápido

comienzo, hay muy poca información sobre cómo realmente usar PhoneGap o realizar desarrollo multiplataforma utilizando PhoneGap (Wargo, 2012).

### **2.6.6.- Ventajas de PhoneGap**

Bibhudatta (2013) resume las ventajas de usar PhoneGap en:

- Un código base que se puede mantener para múltiples plataformas.
- Reutilizar habilidades de desarrollador web existentes.
- Desarrollo más rápido.
- Basado en estándares abiertos.

## CAPITULO 3

### MARCO METODOLÓGICO

En este capítulo, se presenta la metodología que se utilizó en la realización del trabajo especial de grado sobre la investigación asociado al desarrollo de una solución móvil que brinde soporte a los procesos inherentes a la gestión de órdenes de servicio en una organización.

Vasilis (2011) señala que una metodología de software es un proceso de software detallado y completo, y ésta puede basarse en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, espiral, entre otros). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto y guías para el uso de herramientas de apoyo.

Por otra parte Blanco (2008) dice que una metodología de desarrollo es una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. Es importante destacar que existen metodologías tradicionales y ágiles las cuales se describen a continuación.

Por su parte, Abrahamsson (2003) define a las metodologías no ágiles como aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes

indicadas pueden considerarse como metodologías tradicionales. Aunque en el caso particular de RUP, por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse), realizando una configuración adecuada, podría considerarse Ágil (Abrahamsson, 2003).

Sommerville (2002) relata que un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento).

Canós (2003), sugiere que cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se mencionan algunas metodologías ágiles. La mayoría de ellas ya estaban siendo utilizadas con éxito en proyectos reales pero les faltaba una mayor difusión y reconocimiento:

- Extreme Programming.
- SCRUM.
- RationalUnifiedProcess (RUP).
- CrystalMethodologies.
- Dynamic Systems Development Method (DSDM).
- Adaptive Software Development (ASD).
- Feature -Driven Development (FDD).
- Lean Development (LD).

Para el desarrollo de este trabajo especial de grado, se utilizó Scrum. Esta decisión fue tomada debido a que aunque SCRUM es muy similar a la metodología XP, la naturaleza de este proyecto encuentra mayor flexibilidad en SCRUM. Al solo incluir tres integrantes el uso de iteraciones más largas y

el no seguir un orden estricto en el desarrollo de los requerimientos convirtieron a SCRUM en una opción más atractiva. A continuación se presenta una descripción detallada de Scrum como metodología ágil, explicando en detalle los conceptos teóricos, características, importancia y las etapas del proceso.

### **3.1.- Scrum**

Scrum es un marco de trabajo para el desarrollo y el mantenimiento de productos complejos, dentro del cual las personas pueden hacer frente a complejos problemas adaptativos, y al mismo tiempo entregar productos del máximo valor posible de forma productiva y creativa (Sutherland & Schwaber, 2011).

En 1993, Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en Easel Corporation. En 1996 lo presentó junto con Ken Schwaber como proceso formal para la gestión del desarrollo de software. En 2001, serían dos de los promulgadores del Manifiesto Ágil. Scrum es considerado un modelo ágil por la Agile Alliance. (Palacio, 2006)

De acuerdo con Palacio (2006) Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Es orientada a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.
- Con base en los principios ágiles, permite:
  - Colaboración estrecha con el cliente
  - Predisposición y respuesta al cambio
  - Desarrollo incremental con entregas frecuentes de funcionalidad
  - Comunicación verbal directa
  - Simplicidad, solo los artefactos necesarios

- Motivación, compromiso y responsabilidad del equipo por la auto-gestión, auto-organización

Scrum no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. La metodología Scrum asume que el proceso de desarrollo de software es impredecible, y lo trata como a una “caja negra” controlada, en vez de manejarlo como un proceso completamente definido. Ésta es una de las principales diferencias entre Scrum y otras metodologías, como los modelos de espiral o de cascada, en los cuales el proceso de desarrollo se define por completo desde el inicio. Por tratar de planificar el proceso en forma completa desde el principio, las metodologías tradicionales fallan al toparse con algunos problemas habituales del desarrollo de software, como la falta de comprensión de los requerimientos al empezar el proceso, el cambio en los requerimientos durante el proceso, o la dificultad para prever los resultados del uso de nuevas herramientas y tecnologías. (El método Scrum, 2007)

En Scrum se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve (normalmente de 30 días).

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente. (Palacio, 2006)

En la Figura 11 se muestra una visión gráfica del proceso Scrum.

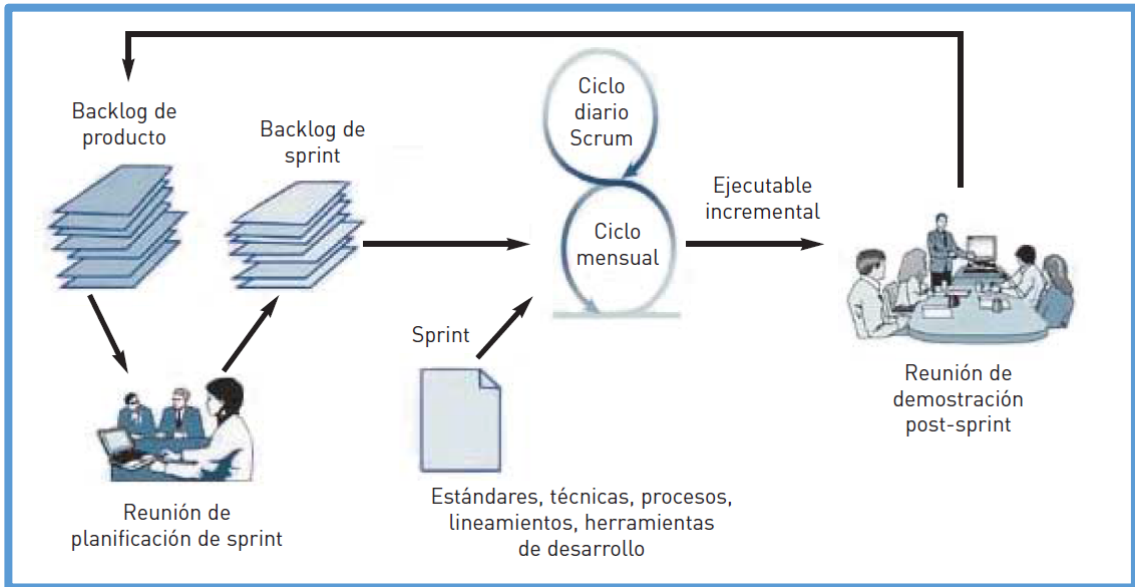


Figura 11. Núcleo del proceso Scrum. (El método Scrum, 2007)

### 3.1.1.- Teoría de Scrum

Sutherland y Schwaber (2011) afirman que la teoría empírica de control de procesos es la base fundamental de Scrum. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea una aproximación iterativa e incremental para optimizar la predictibilidad y controlar el riesgo.

Los pilares del control empírico de procesos son:

**Transparencia:** Los responsable del resultado deben tener a disposición los aspectos importantes del proceso. Dichos aspectos han de estar definidos por un estándar común, para que los observadores puedan tener una visión y entendimiento común.

Es muy importante que se comparta un lenguaje común para referirse al proceso y compartir una definición común de “Hecho”.

**Inspección:** Los usuarios de Scrum deben inspeccionar con frecuencia los artefactos de Scrum el progreso hacia un objetivo, para detectar variaciones no deseables.

**Adaptación:** Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe ser realizado cuanto antes para minimizar desviaciones mayores.

### **3.1.2.- Visión general del proceso**

Según Sutherland y Schwaber (2011) el desarrollo en Scrum consiste de elementos, roles y reuniones.

#### **3.1.2.1.- Elementos**

Los elementos o artefactos en Scrum representan trabajo o valor añadido que aportan transparencia y oportunidades para la revisión y adaptación. Los artefactos están diseñados específicamente para facilitar la transparencia de la información clave y unificar los criterios de comprensión de dicho artefacto.

- **Pila del producto o Product Backlog.** Lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
- **Pila del sprint o Sprint Backlog.** Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- **Incremento.** Producto operativo con una funcionalidad nueva o mejorada como resultado de cada sprint.

#### **3.1.2.2.- Roles**

Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: Propietario del Producto, Equipo de Desarrollo y Scrum Master. Los equipos Scrum son auto-organizados y multifuncionales.

Los equipos auto-organizados eligen la mejor forma de llevar a cabo su trabajo, en lugar de ser dirigidos por otros externos al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otros que no son parte del equipo.

- **Propietario del Producto.** Es el responsable del proyecto y de obtener el mayor valor de producto para los clientes, usuarios y resto de implicados. El Dueño de Producto es la única persona responsable de gestionar la Pila del Producto y sus decisiones se reflejan en el contenido y en la priorización de la misma.
- **Equipo de Desarrollo.** Es el grupo o grupos de trabajo que desarrollan o entregan un incremento de producto ejecutable al final de cada Sprint. Ellos organizan y gestionan su propio trabajo y son multifuncionales.
- **Scrum Master.** Es el responsable del funcionamiento de la metodología Scrum y de la productividad del equipo de desarrollo.

### **3.1.2.3.- Reuniones**

Las reuniones de Scrum se crean en forma de bloques de tiempo, con el fin de crear regularidad y reducir al máximo la necesidad de reuniones que no son parte de Scrum.

Cada uno de los eventos de Scrum resulta ser una oportunidad para la inspección y adaptación de algún aspecto.

El Sprint es el núcleo central que proporciona la base de desarrollo iterativo e incremental de Scrum. Consiste de un bloque de tiempo de un mes o menos de duración durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable.

Los Sprints contienen y consisten en la Reunión de Planificación del Sprint, los Scrums Diarios, el trabajo de desarrollo, la Revisión del Sprint y la Retrospectiva del Sprint.

- **Reunión de Planificación de Sprint.** Es una reunión previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración. En ella interviene de forma colaborativa todo el Equipo Scrum.
- **Scrum Diario.** Es una reunión de 15 minutos de duración, durante la cual se revisa el avance del trabajo realizado hasta la fecha desde el último Scrum Diario y lo que se prevé cumplir dentro de las próximas 24 horas.
- **Revisión de Sprint.** Es el análisis y revisión del incremento generado al final del Sprint. De acuerdo con ello podría adaptarse o no la Pila de Producto.
- **Retrospectiva de Sprint.** Se lleva a cabo después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. En ella el Equipo Scrum se inspecciona a sí mismo, creando un plan de mejoras para ser abordadas en el siguiente Sprint.

## **CAPÍTULO 4**

### **MARCO APLICATIVO**

En este capítulo se presenta la implementación de la propuesta de solución del Trabajo Especial de Grado; realizándose a una adaptación del proceso Scrum para la construcción de la solución móvil. En tal sentido, se describe el contexto de desarrollo y cada una de los elementos y actividades de Scrum utilizados para la automatización de los procesos relacionados con la gestión de órdenes de servicio en una organización. De esta manera, se expone la síntesis del proyecto y cada una de las etapas asociadas al enfoque aplicado.

#### **4.1.- Contexto del Desarrollo**

Este proyecto ha sido desarrollado en el contexto del TEG presentado ante la ilustre Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación, bajo la tutoría del profesor Franklin Sandoval, para optar al Título de Licenciado en Computación, por parte de los bachilleres Manuel Rojas y José Castro, integrantes del grupo desarrollador.

#### **4.2.- Proyecto**

Para iniciar con el desarrollo de la solución móvil fue necesaria la elección de una metodología adaptable a las características del proyecto de software a emprender, para lo cual fue importante realizar un análisis previo de las distintas metodologías existentes. Se detectó la necesidad de una metodología ágil que permitiera introducir cambios en el proyecto, debido a que las especificaciones no eran totalmente cerradas y en la medida que avanzara el proyecto podía ser interesante priorizar tareas o incluso redefinirlas. Finalmente se optó por Scrum como metodología de desarrollo.

### **4.3.- Aplicación de la metodología de desarrollo**

En Scrum el proyecto se ejecuta en iteraciones de un mes, aunque pueden ser de una o dos semanas, si así se necesita. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite, es decir, una versión del producto. Scrum parte de la misma base que las metodologías clásicas, la búsqueda de un proyecto y la construcción del documento de requerimientos.

La aplicación de Scrum en el presente proyecto tuvo que ser personalizada para el correcto desarrollo y aplicación de la misma, ya que los elementos del proyecto han sido diferentes a los de una situación ideal. También resultaron algunas variaciones debido a que el equipo es más reducido de lo normal.

#### **4.3.1.- Definición de Roles**

Como se mencionó en el capítulo 3, en Scrum existen tres actores responsables del proyecto: propietario del producto, equipo de desarrollo y Scrum Master.

En este punto es preciso aclarar que los roles definidos fueron distribuidos tomando en cuenta lo reducido del equipo de trabajo; por lo tanto los roles asignados quedaron como se muestra a continuación:

- Propietario del producto: Prof. Franklin Sandoval, quien proporcionó la información y funcionalidades del sistema, es decir, es el encargado de crear y mantener la Pila del Producto asegurándose que se realicen con anterioridad aquellos requerimientos con mayor prioridad.
- Equipo desarrollador: Br Manuel Rojas y Br. José Castro, autores de la presente tesis, quienes son los encargados de desarrollar las funcionalidades del software.

- Scrum Master: es también el autor de la presente tesis, Br. Manuel Rojas, responsable de la aplicación de la metodología y todos los aspectos relacionados.

Los actores en el presente proyecto tomando en cuenta la metodología utilizada se conformarían tal como aparece en la Tabla 3.

**Tabla 3. Actores de Scrum en el proyecto**

<b><i>Propietario del Producto</i></b>	<b><i>Prof. Franklin Sandoval</i></b>
<b><i>Scrum Manager</i></b>	<b><i>Br. Manuel Rojas</i></b>
<b><i>Equipo de Desarrollo</i></b>	<b><i>Br. Manuel Rojas - Br. José Castro</i></b>

#### **4.3.2.- Elementos**

*Sprint.* Se ha definido un sprint de 7 a 15 días, en los cuales se harán las revisiones respectivas de los elementos y entregables del proyecto.

*Pila del Sprint.* En este documento se registrará el requerimiento o actividad realizada en detalle. Será utilizada por cualquier miembro del equipo. (Ver Tabla 4)

**Tabla 4. Pila del Sprint**

<b>Descripción</b>	<b>Responsable</b>	<b>Prioridad</b>	<b>Estado</b>	<b>Est. (h)</b>

***Pila del Producto.*** Esta tabla será la herramienta de seguimiento del producto total. En ella se registrarán los cambios realizados en todos los Sprints. (Ver Tabla 5)

**Tabla 5. Pila del Producto**

Descripción	Prioridad	Estado	Observación

### 4.3.3.- Reuniones y aplicación en el proyecto

El desarrollo comienza con la creación del documento de Pila del Producto, que incluye todas las tareas, funcionalidades y requerimientos que el producto debe satisfacer. Éste es redactado y actualizado por el Propietario del Producto. Seguidamente, se describen los documentos correspondientes a cada una de las iteraciones o Sprints y las actividades llevadas a cabo para cumplir con los requerimientos. Para el presente proyecto se realizan 4 iteraciones o Sprints.

Al final de cada Sprint, se actualiza el estado de la pila del producto de acuerdo con los objetivos o requerimientos cumplidos en la iteración.

La pila del producto incluye todas las tareas, funcionalidades y requerimientos que el producto debe satisfacer. En la Tabla 6 se observan los requerimientos que tendrá el sistema; la misma puede variar en el transcurso de la iteración. Se establece el código de requisito R que identifica al requisito del sistema como único. En Prioridad se especifica en qué orden debe ser implementado.

Es importante destacar que existen aspectos definidos en diversas fuentes sobre Scrum que no se adaptaron fácilmente a las características de éste proyecto por lo que fueron modificados u omitidos.

**Tabla 6. Estado inicial de la pila del producto**

Descripción		Prioridad	Estado	Observación
R1	Diseño y construcción base de datos móvil y web	Alta	<b>Pendiente</b>	
R2	Desarrollo de casos de uso del sistema	Alta	<b>Pendiente</b>	
R3	Diseño de la interfaz de la página principal	Alta	<b>Pendiente</b>	
R4	Diseño de la interfaz para el login de usuario	Alta	<b>Pendiente</b>	
R5	Programación de conexiones base de datos móvil	Alta	<b>Pendiente</b>	
R6	Pruebas de versión	Alta	<b>Pendiente</b>	
R7	Diseño interfaz página de crear nuevas órdenes	Alta	<b>Pendiente</b>	
R8	Diseño interfaz de la página de lista de órdenes	Alta	<b>Pendiente</b>	
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Pendiente</b>	
R10	Diseño interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Pendiente</b>	
R11	Programación de servicio web	Alta	<b>Pendiente</b>	
R12	Pruebas de versión	Alta	<b>Pendiente</b>	
R13	Diseño interfaz página cancelación orden servicio	Alta	<b>Pendiente</b>	
R14	Diseño interfaz página calendario órdenes servicio	Media	<b>Pendiente</b>	
R15	Diseño interfaz de la página de reporte de servicio	Alta	<b>Pendiente</b>	
R16	Diseño interfaz página agregar partes y materiales	Alta	<b>Pendiente</b>	
R17	Programación de interfaces de conexión con servicios web	Alta	<b>Pendiente</b>	
R18	Implementación de la interfaz de la página principal	Alta	<b>Pendiente</b>	
R19	Implementación interfaz para el login de usuario	Alta	<b>Pendiente</b>	
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Pendiente</b>	
R21	Pruebas versión	Alta	<b>Pendiente</b>	
R22	Implementación de la interfaz de la página de crear nuevas órdenes	Alta	<b>Pendiente</b>	
R23	Implementación de la interfaz de la página de lista de órdenes	Alta	<b>Pendiente</b>	
R24	Implementación de la interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Pendiente</b>	
R25	Implementación interfaz página cancelación orden	Alta	<b>Pendiente</b>	
R26	Implementación interfaz página calendario orden	Alta	<b>Pendiente</b>	
R27	Implementación interfaz página reporte servicio	Alta	<b>Pendiente</b>	
R28	Implementación interfaz agregar parte y material	Alta	<b>Pendiente</b>	
R29	Implementación de la interfaz de sincronización de la aplicación móvil	Alta	<b>Pendiente</b>	
R30	Pruebas de versión final	Alta	<b>Pendiente</b>	

### ***Sprint 1***

Las actividades o requerimientos a ser cumplidos en el primer Sprint corresponden a las descritas en la Tabla 7

**Tabla 7. Pila del Sprint 1**

Descripción		Res.	Prioridad	Est.(h)
R1	Diseño y construcción de la base de datos	M. R.	Alta	12
R2	Desarrollo de casos de uso del sistema	M. R.	Alta	4
R3	Diseño de la interfaz de la página principal	M. R.	Alta	10
R4	Diseño y construcción de la página de login usuario	J. C.	Alta	8
R5	Programación de conexiones con la base de datos móvil	J. C.	Alta	4
R6	Pruebas de versión	J. C.	Alta	4

### **R1 - Diseño y construcción de la base de datos**

El diseño de la base de datos es la parte primordial para empezar con el desarrollo de cualquier aplicación, ya que un buen diseño de esta definirá el óptimo funcionamiento y éxito de cualquier software. (Ver Figura 12)

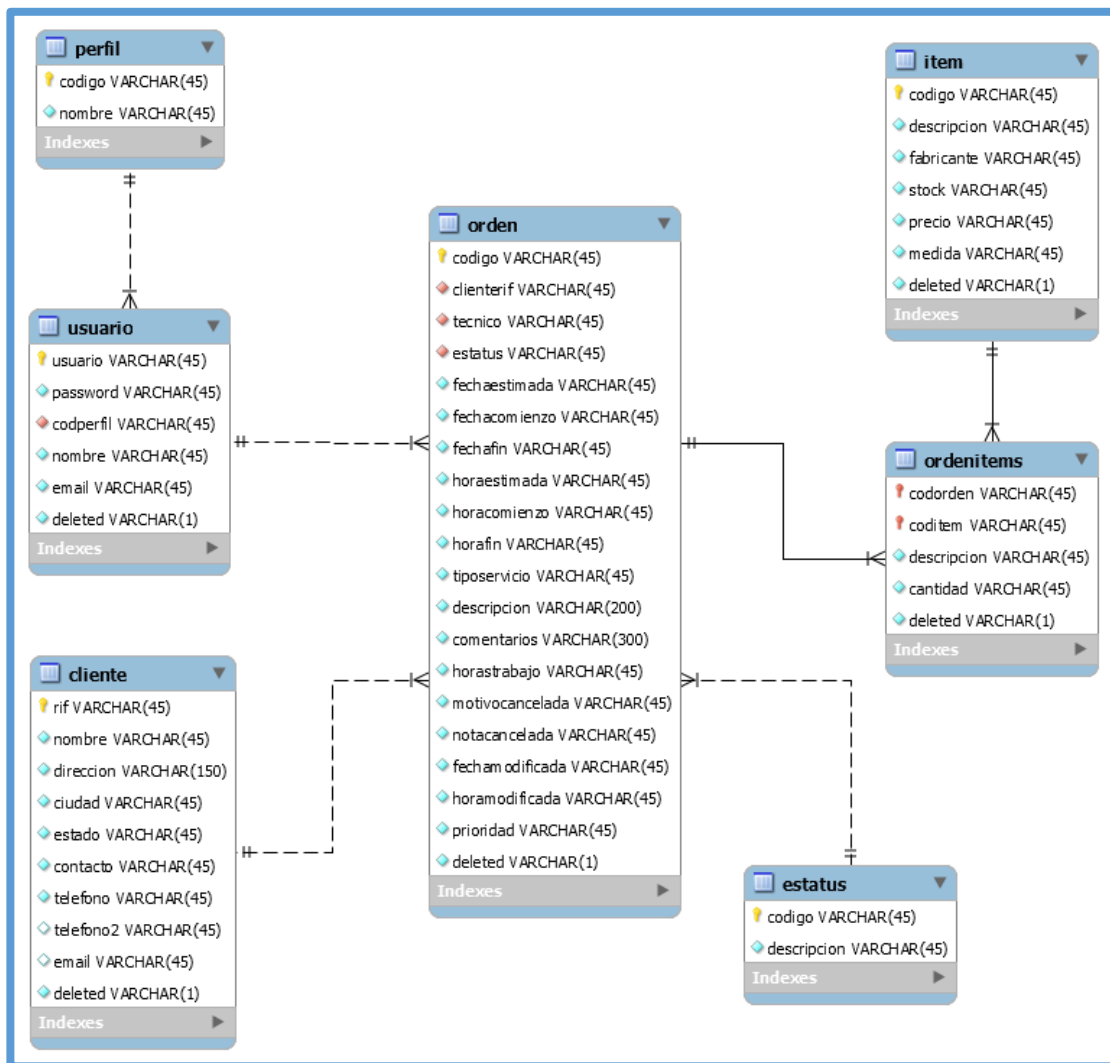


Figura 12. Modelo E/R de la base de datos

## R2 - Desarrollo de casos de uso del sistema

A continuación se describen a través de los diagramas de casos de uso cada una de las funcionalidades del sistema, tanto en la aplicación móvil como en la aplicación web.

Los actores involucrados en los casos de uso de la gestión de órdenes de servicio en una organización se describen de la siguiente manera:

*Despachador:* encargado de recibir las solicitudes verbales o escritas de órdenes de servicio y registrarlas en el sistema mediante la creación de una orden de servicio en estado “Nueva”.

*Gerente:* tiene los mismos privilegios que el Despachador pero, adicionalmente, también gestiona las órdenes de servicio a nivel de consulta y actualización, asignadas a todos los técnicos.

*Técnico:* Es el personal encargado de la ejecución en sitio de la orden de servicio. Sólo puede consultar y actualizar las órdenes de servicio que le han sido asignadas.

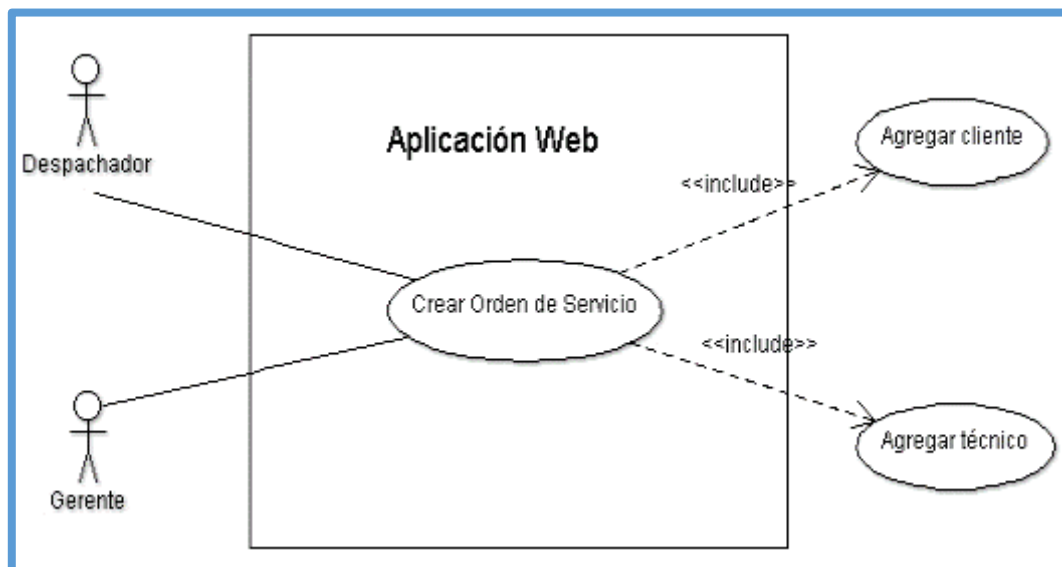


Figura 13. Diagrama de Casos de Uso Creación de Órdenes de Servicio en el Sistema Web

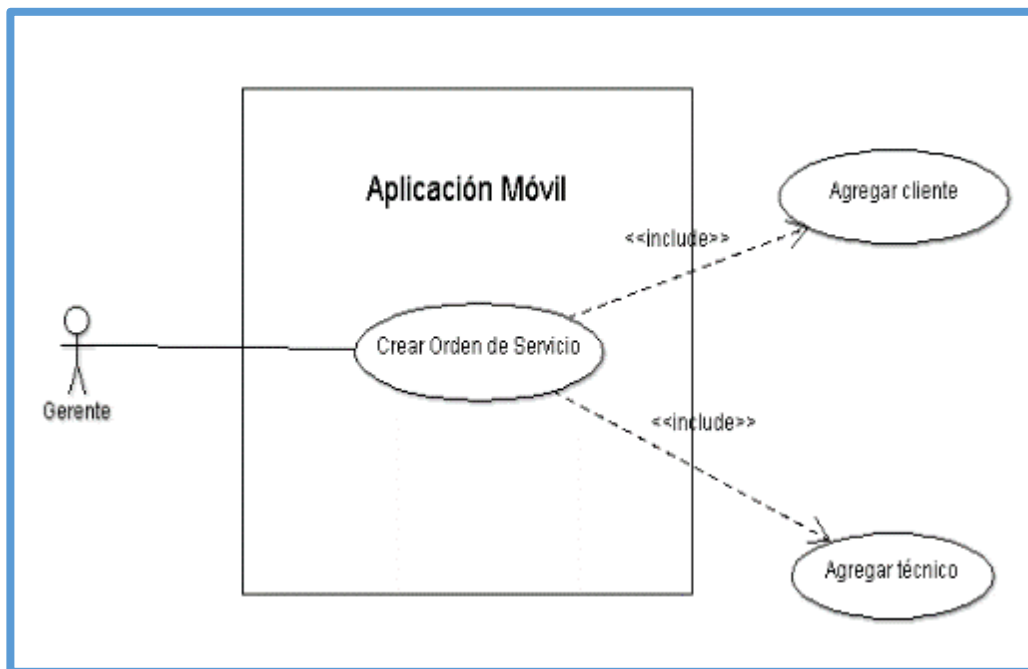


Figura 14. Diagrama de Casos de Uso para la Creación de Órdenes de Servicio en el Sistema Móvil

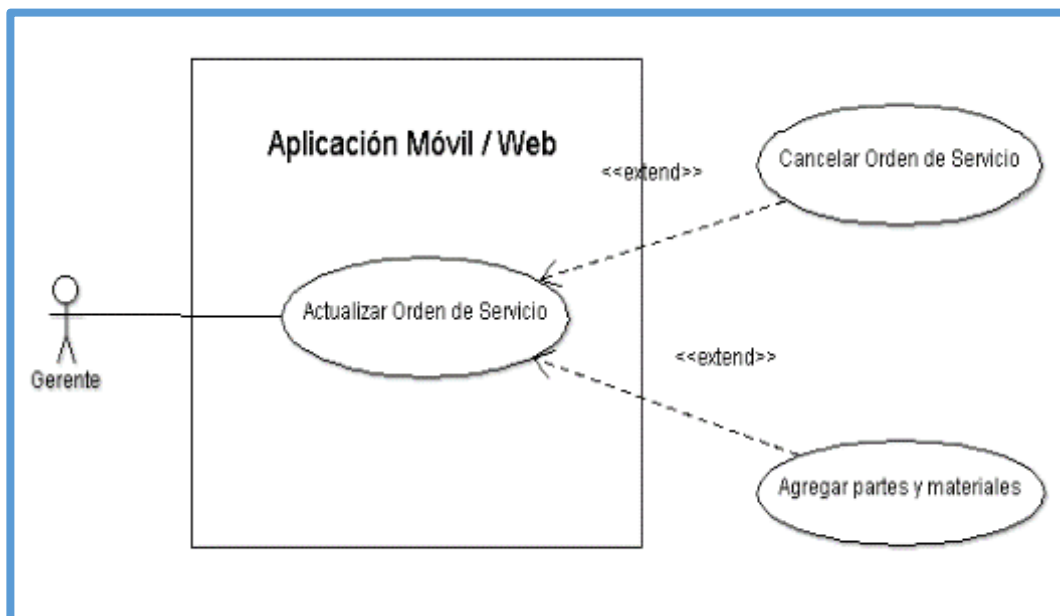


Figura 15. Diagrama de Casos de Uso para la Actualización de Órdenes de Servicio

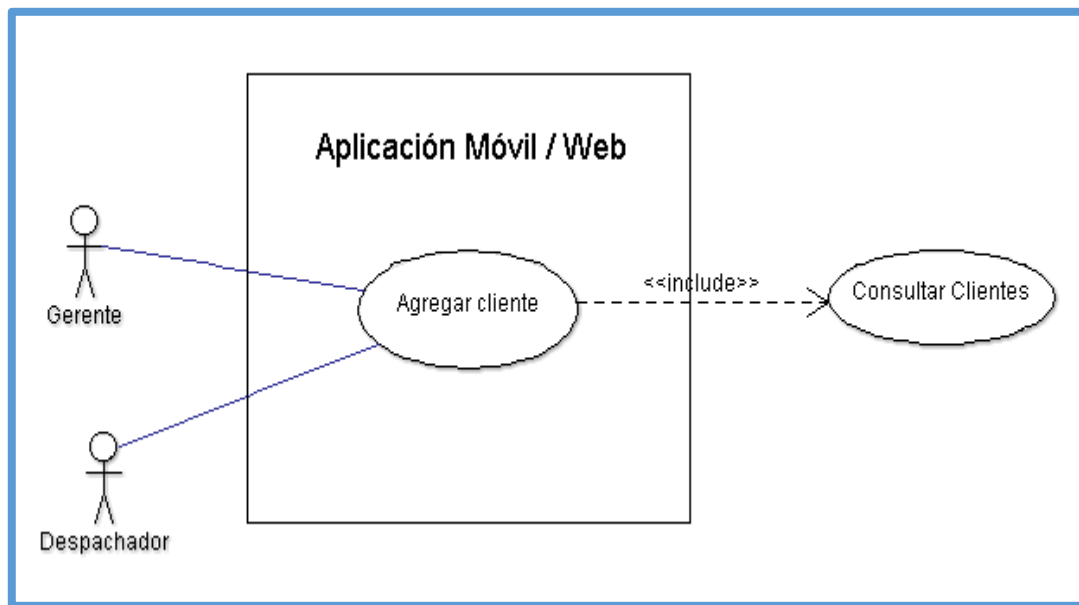


Figura 16. Diagrama de Casos de Uso para Agregar Cliente

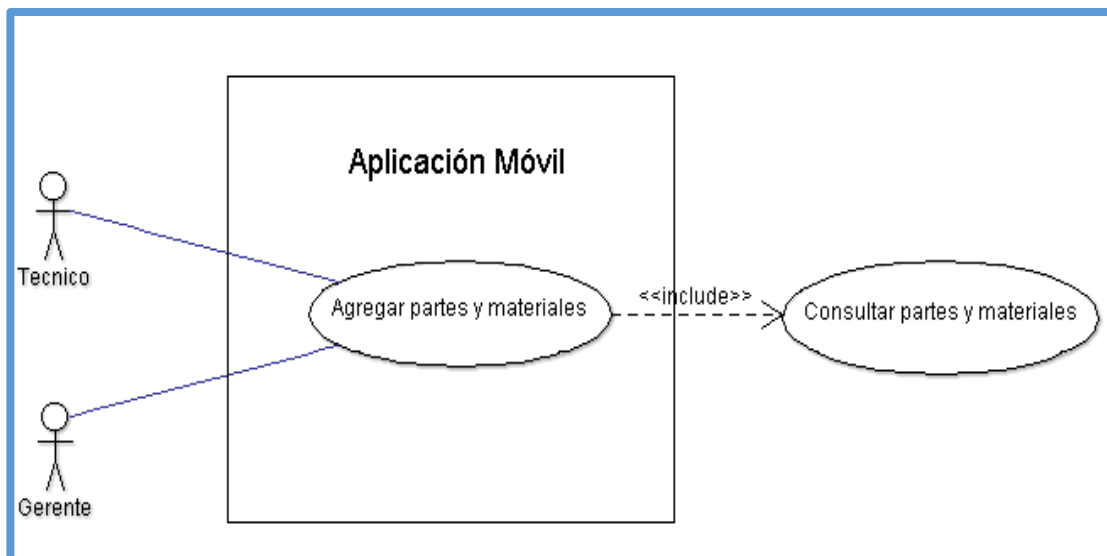


Figura 17. Diagrama de Casos de Uso para Agregar Partes y Materiales a una Orden de Servicio en el Sistema Móvil

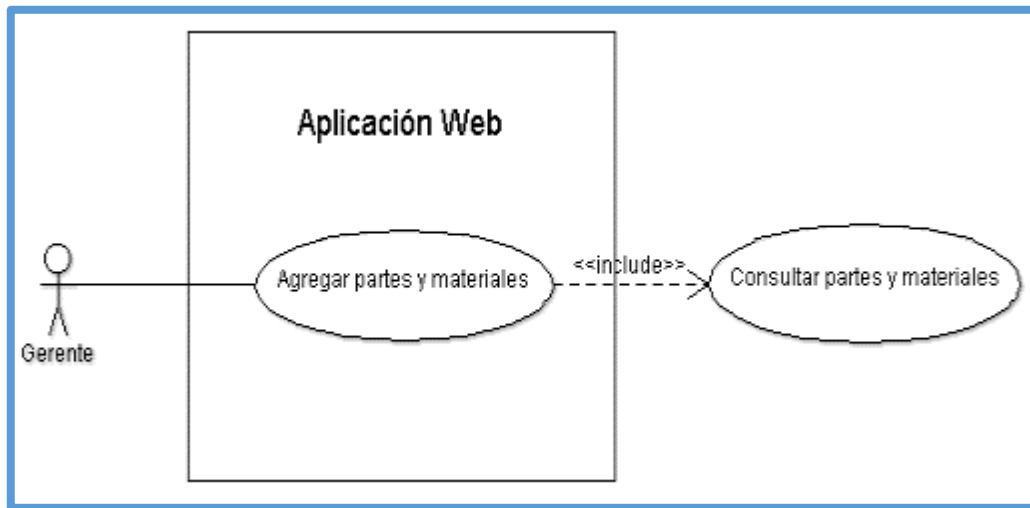


Figura 18. Diagrama de Casos de Uso para Agregar Partes y Materiales a una Orden de Servicio en el Sistema Web

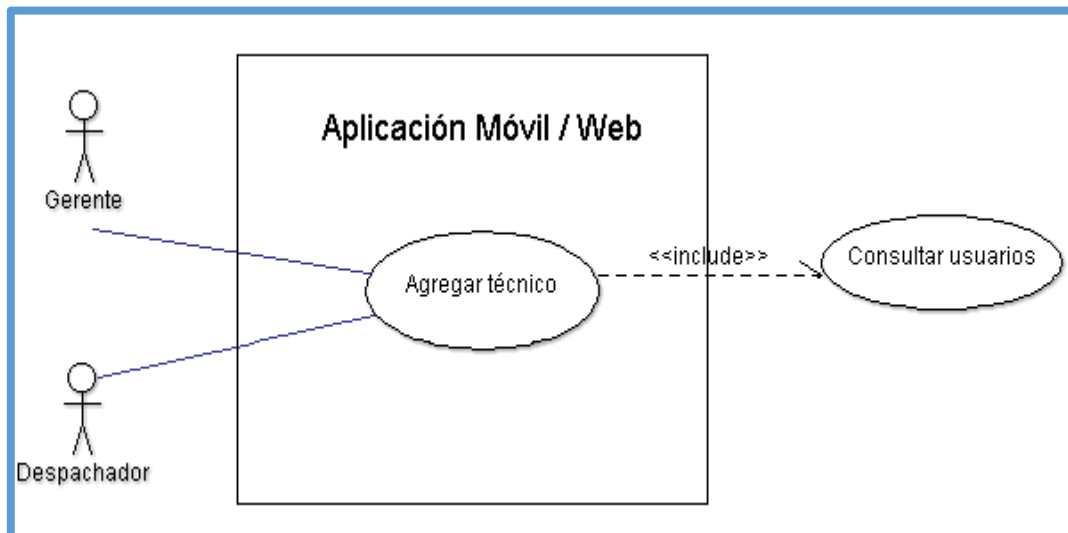


Figura 19. Diagrama de Casos de Uso para Agregar Técnico

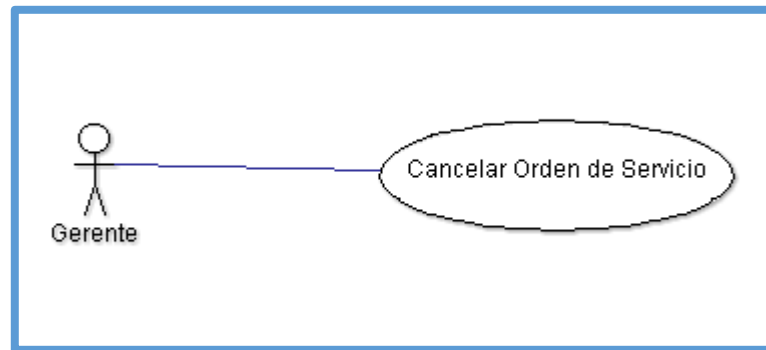


Figura 20. Diagrama de Casos de Uso para Cancelar una orden de servicio

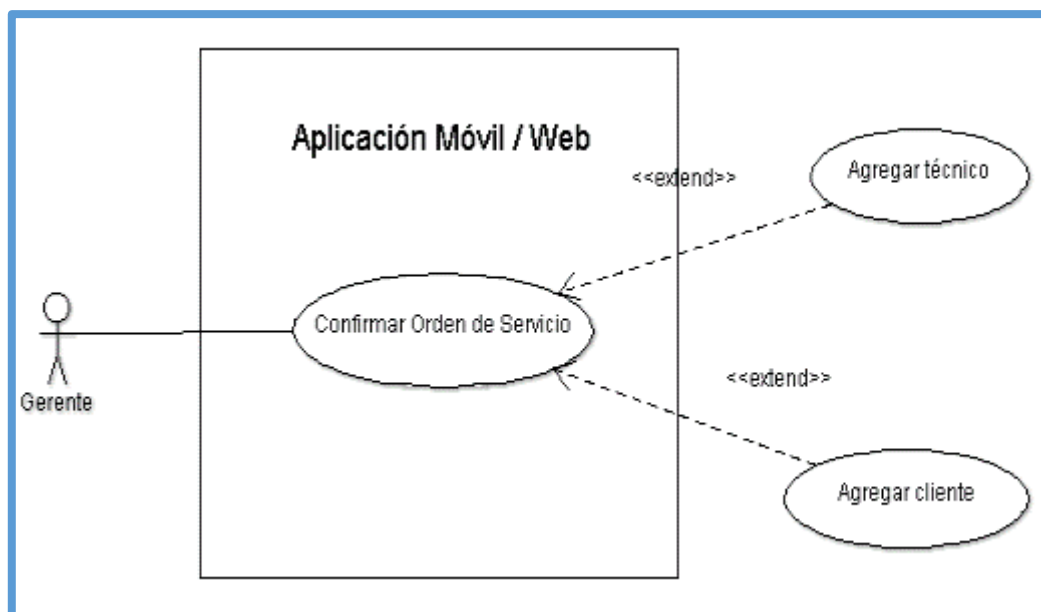


Figura 21. Diagrama de Casos de Uso para Confirmar una Orden de Servicio

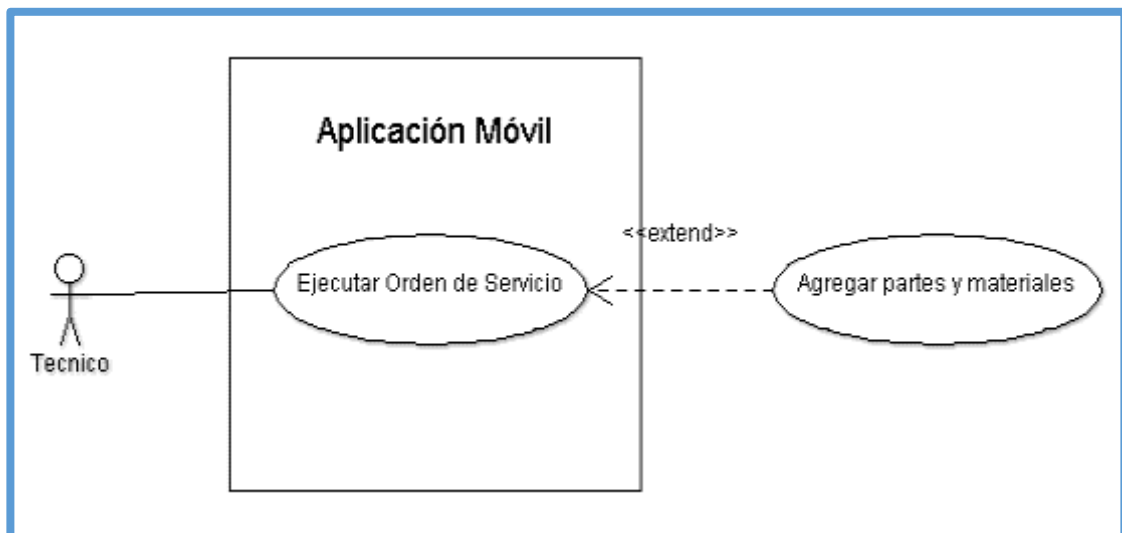


Figura 22. Diagrama de Casos de Uso para Ejecutar una Orden de Servicio

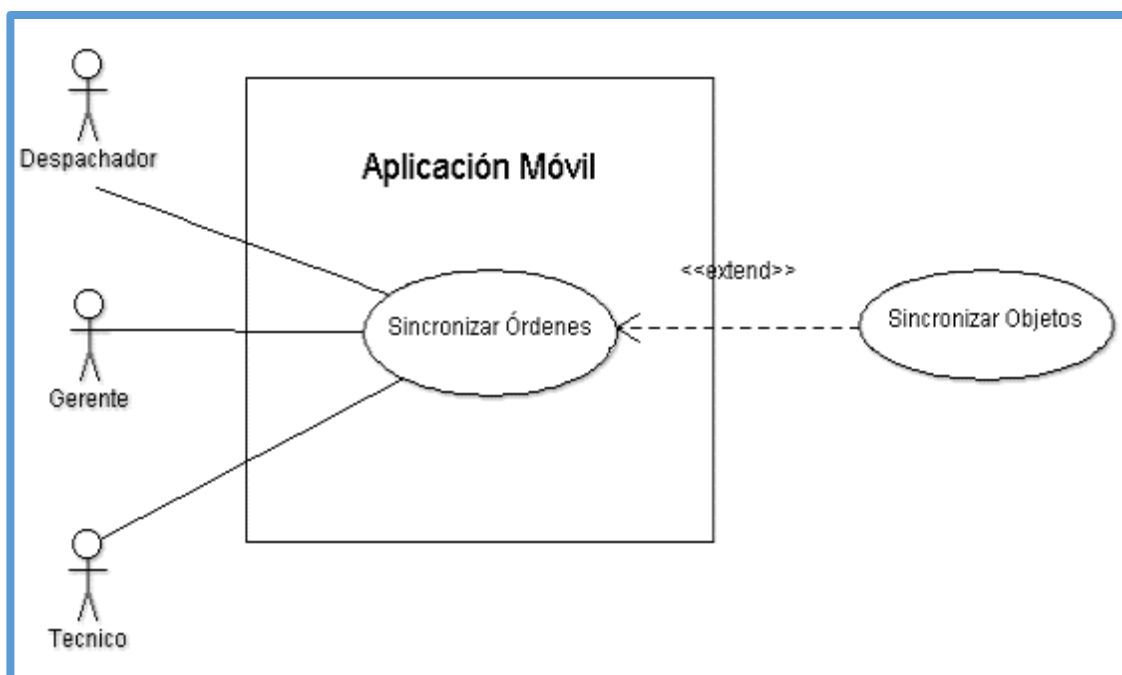


Figura 23. Diagrama de Casos de Uso para sincronizar Órdenes de Servicio

### **R3 - Diseño de la interfaz de la página principal**

Para la implementación de la solución móvil se requirió de un sub-sistema móvil y un sub-sistema web, trabajando en base a una arquitectura Cliente-Servidor. Es importante mencionar que para mantener la consistencia en la solución móvil como conjunto, se mantuvo en lo posible el mismo diseño tanto para la aplicación web como para la aplicación móvil.

En la Figura 24 se muestra el resultado del diseño de la interfaz de la página principal de la aplicación móvil. Esta página se encuentra dividida en tres secciones. La primera es la sección de órdenes de servicio, que contiene el ícono para acceder al formulario de crear órdenes de servicio y los íconos que permiten acceder a las listas de órdenes de servicio clasificadas por estatus. La sección de Búsqueda contiene los íconos de enlace que conducen a las distintas listas para la búsqueda de partes, clientes y técnicos. Finalmente, la sección Opciones presenta el ícono de enlace a las páginas de sincronización de datos entre la aplicación móvil y la aplicación web y el ícono que permite acceder a la configuración para conectar con el servidor. Es importante mencionar que la sección Opciones sólo aplica para la parte móvil.



**Figura 24. Página principal de la aplicación móvil**

En la aplicación web se consideró necesario agregar un botón de “cerrar sesión”, para permitirle al usuario actual finalizar su sesión de trabajo en la aplicación (Ver Figura 25)



Figura 25. Página principal de la aplicación web

#### R4 - Diseño de la interfaz para el login de usuario

Se diseña una página de login o autenticación estándar para permitir el ingreso del usuario a la aplicación móvil (Ver Figura 26). El mismo resultado se aprecia en la Figura 27 para la aplicación web.

Figura 26. Página de login de la aplicación móvil

The image shows a web application login interface. At the top, there is a blue header with the word 'Autenticacion' in white. Below the header, there is a white form area. On the left side of the form, there are two labels: 'Usuario' and 'Password'. The 'Usuario' label is above a text input field containing the word 'manuel'. The 'Password' label is above a password input field containing six asterisks. Below the input fields, there is a blue button with the word 'Login' in white text.

Figura 27. Página de login de la aplicación web

## R5 - Programación de conexiones con la base de datos móvil

Junto con HTML5 y PhoneGap existen varias tecnologías que permiten que las aplicaciones móviles almacenen datos en los dispositivos cliente. Según lo que decida el desarrollador, la información puede sincronizarse también con el servidor o permanecer siempre en el cliente. En el presente trabajo se opta por lo primero.

Hay varias razones por las que se consideró utilizar el almacenamiento en el cliente. En primer lugar, el almacenamiento en el cliente permite que una aplicación funcione cuando el usuario no está conectado, posiblemente sincronizando datos cuando vuelve a establecer conexión. En segundo lugar, aumenta el rendimiento, por lo que se puede mostrar una gran cantidad de datos en cuanto el usuario hace clic en el sitio en lugar de esperar a que vuelvan a descargarse. En tercer lugar, es un modelo de programación más sencillo que no requiere infraestructura de servidor.

La conexión con la base de datos móvil se realiza a través de interfaces de comunicación provistas por el API Web SQL Database. Esta es una especificación que trae el lenguaje de consulta SQL al lado del cliente, razón por la cual es muy conveniente cuando se tiene experiencia como

desarrollador de back-end. En la Figura 28 se puede apreciar un fragmento del código utilizado para la conexión a la base de datos móvil.

```
receivedEvent: function(id) {
    $("#deviceready").hide();
    db = window.openDatabase("ordenservicio_db", "1.0", "Ordenes de Servicio", 200000);
    db.transaction(populateDB, errorCallback, successCB);
}

function populateDB(tx) {
    console.log('creating database...');

    tx.executeSql('CREATE TABLE IF NOT EXISTS Cliente (rif TEXT PRIMARY KEY, ' +
        'nombre TEXT NOT NULL, '+
        'direccion TEXT NOT NULL, '+
        'ciudad TEXT NOT NULL, '+
        'estado TEXT NOT NULL, '+
        'contacto TEXT NOT NULL, '+
        'telefono TEXT NOT NULL, '+
        'telefono2 TEXT, '+
        'email TEXT)');
}
```

Figura 28. Conexión a la base de datos móvil

## R6 – Pruebas de funcionalidad de versión

Sánchez (2015) afirma que “este tipo de pruebas se basa en las funcionalidades de un sistema que se describen en la especificación de requisitos, es decir, lo que hace el sistema”. Por lo tanto, las pruebas de funcionalidad están destinadas a verificar que se cumpla con los requerimientos funcionales descritos en de los casos de uso.

Una vez desarrollados en esta primera iteración los requisitos de la Pila del Sprint, se procede a realizar las pruebas funcionalidad. Se utilizan datos tipo para observar el funcionamiento y los controles implementados en el sistema e identificar las posibles fallas.

Para garantizar que la versión obtenida sea consistente, se realizaron las siguientes pruebas de funcionalidad.

Tabla 8. Prueba de funcionalidad de control de acceso a la aplicación

<b>Caso de Prueba:</b> Control de acceso a la aplicación
<b>Requisito 4:</b> Diseño y construcción de la página de login de usuario
<b>Descripción:</b> Se ingresa el nombre de usuario y password. Luego se presiona sobre el botón Login para obtener acceso a la página principal de la aplicación.
<b>Condiciones de Ejecución:</b> Ninguno
<b>Pasos de ejecución:</b> 1. Se inicia la aplicación. 2. Se ingresan el usuario y password 3. Se presiona el botón de Login
<b>Resultado Esperado:</b> El usuario es autenticado y se visualiza la página principal del sistema
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 9. Prueba de funcionalidad de acceso a la página principal de la aplicación

<b>Caso de Prueba:</b> Acceso a la página principal de la aplicación
<b>Requisito 3:</b> Diseño de la interfaz de la página principal
<b>Descripción:</b> Se abre la página principal de la aplicación.
<b>Condiciones de Ejecución:</b> El usuario debe estar autenticado en el sistema
<b>Pasos de ejecución:</b> 1. Se inicia la aplicación 2. El usuario se autentica y se abre la página principal de la aplicación
<b>Resultado Esperado:</b> se visualiza la página principal del sistema y todos sus componentes
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

El estado actualizado de la pila del producto al final del Sprint 1 se describe en la

Tabla 10

**Tabla 10. Estado de la pila del producto luego del Sprint 1**

Descripción		Prioridad	Estado	Observación
R1	Diseño y construcción base de datos móvil y web	Alta	<b>Terminada</b>	
R2	Desarrollo de casos de uso del sistema	Alta	<b>Terminada</b>	
R3	Diseño de la interfaz de la página principal	Alta	<b>Terminada</b>	
R4	Diseño de la interfaz para el login de usuario	Alta	<b>Terminada</b>	
R5	Programación de conexiones base de datos móvil	Alta	<b>Terminada</b>	
R6	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R7	Diseño interfaz página de crear nuevas órdenes	Alta	<b>Pendiente</b>	
R8	Diseño interfaz de la página de lista de órdenes	Alta	<b>Pendiente</b>	
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Pendiente</b>	
R10	Diseño interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Pendiente</b>	
R11	Programación de servicio web	Alta	<b>Pendiente</b>	
R12	Pruebas de funcionalidad de versión	Alta	<b>Pendiente</b>	
R13	Diseño interfaz página cancelación orden servicio	Alta	<b>Pendiente</b>	
R14	Diseño interfaz página calendario órdenes servicio	Media	<b>Pendiente</b>	
R15	Diseño interfaz de la página de reporte de servicio	Alta	<b>Pendiente</b>	
R16	Diseño interfaz página agregar partes y materiales	Alta	<b>Pendiente</b>	
R17	Programación de interfaces de conexión con servicios web	Alta	<b>Pendiente</b>	
R18	Implementación de la interfaz de la página principal	Alta	<b>Pendiente</b>	
R19	Implementación interfaz para el login de usuario	Alta	<b>Pendiente</b>	
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Pendiente</b>	
R21	Pruebas de funcionalidad de versión	Alta	<b>Pendiente</b>	
R22	Implementación de la interfaz de la página de crear nuevas órdenes	Alta	<b>Pendiente</b>	
R23	Implementación de la interfaz de la página de lista de órdenes	Alta	<b>Pendiente</b>	
R24	Implementación de la interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Pendiente</b>	
R25	Implementación interfaz página cancelación orden	Alta	<b>Pendiente</b>	

R26	Implementación interfaz página calendario orden	Alta	<b>Pendiente</b>	
R27	Implementación interfaz página reporte servicio	Alta	<b>Pendiente</b>	
R28	Implementación interfaz agregar parte y material	Alta	<b>Pendiente</b>	
R29	Implementación de la interfaz de sincronización de la aplicación móvil	Alta	<b>Pendiente</b>	
R30	Pruebas de funcionalidad de versión	Alta	<b>Pendiente</b>	

### ***Sprint 2***

Las actividades o requerimientos a ser cumplidos en el Sprint 2 corresponden a las descritas en la Tabla 11

**Tabla 11. Pila del Sprint 2**

<b>Descripción</b>		<b>Res.</b>	<b>Prioridad</b>	<b>Est.(h)</b>
R7	Diseño de la interfaz de la página de crear nuevas órdenes	M. R.	Alta	4
R8	Diseño de la interfaz de la página de lista de órdenes	J.C	Alta	4
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	J.C.	Alta	6
R10	Diseño de la interfaz de la página de confirmación de nuevas órdenes	J.C.	Alta	4
R11	Programación de servicio web	M.R.	Alta	20
R12	Pruebas de funcionalidad de versión	M.R.	Alta	4

### **R7 - Diseño de la interfaz de la página de crear nuevas órdenes**

El formulario de creación de nuevas órdenes de servicio permite, como su nombre lo indica, registrar nuevas órdenes de servicio en el sistema. Los roles asignados para este propósito son el de gerente y despachador. Los datos solicitados para crear una orden de servicio son: nombre del cliente, fecha estimada, hora estimada, tipo de servicio, técnico asignado, descripción. (Ver Figura 29 y Figura 30)

The image shows a mobile application interface for creating a new order. The title bar is blue with a back arrow and the text "Nueva Orden". Below the title bar, there are several input fields: "Nombre del Cliente:" with a dropdown menu showing "Seleccione un cliente"; "Fecha Estimada" with a date picker icon; "Hora Estimada" with a time picker icon; "Tipo de Servicio" with a dropdown menu showing "Inspeccion"; "Tecnico" with a text input field; and "Descripcion" with a text input field.

Figura 29. Página de crear orden de la aplicación móvil

The image shows a web application interface for creating a new order. The title bar is blue with the text "Nueva Orden". Below the title bar, there are several input fields: "Nombre del Cliente:" with a dropdown menu showing "Seleccione un cliente"; "Fecha / Hora Estimada." with a combined date and time input field; "Tipo de Servicio" with a dropdown menu showing "Inspeccion"; "Descripcion" with a text input field; and "Tecnico" with a text input field. At the bottom of the form, there is a blue button labeled "Aceptar".

Figura 30. Página de crear orden de la aplicación web

## R8 - Diseño de la interfaz de la página de lista de órdenes

La lista de órdenes de servicio despliega por cada una un resumen de la información de la orden de servicio. Cada fila de la lista muestra el nombre del cliente, el tipo de servicio contratado, la fecha y hora estimada de ejecución del servicio y el estatus de la orden. (Ver Figura 31)

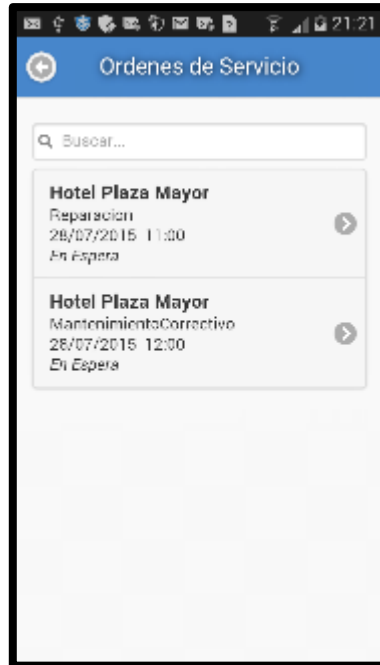


Figura 31. Página de lista de órdenes de servicio

## R9 - Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos

La estructura de la lista que se despliega para clientes, partes y técnicos es básicamente la misma de la lista de órdenes de servicio. En la pantalla de lista de clientes se muestra la información de nombre y dirección resumida de cliente por cada fila. (Ver Figura 32).

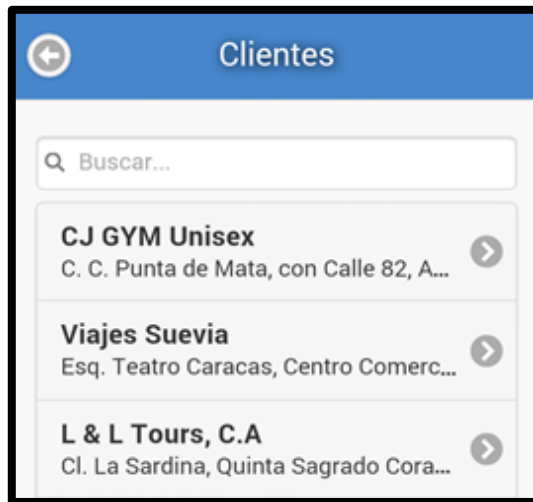


Figura 32. Página de Lista de Clientes

En la Figura 33 se muestra una sección de la pantalla de lista de técnicos con información de nombre y código de parte o material por cada fila.

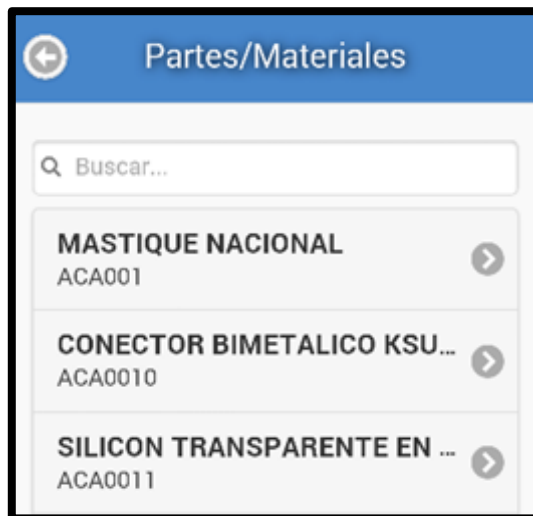


Figura 33. Página de Lista de Partes/Materiales

La pantalla de lista de técnicos muestra por cada fila el nombre y correo electrónico correspondiente a cada técnico. (Ver Figura 34)

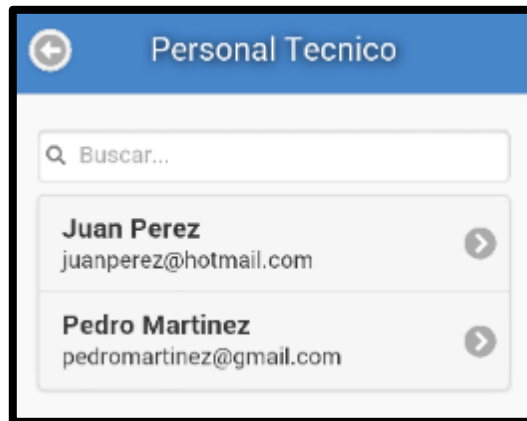


Figura 34. Página de Lista de Técnicos

## R9 - Diseño de la interfaz de las páginas de detalle de clientes, partes y técnicos

La pantalla Detalle de Partes/Materiales contiene la información completa correspondiente a la parte o material seleccionado de la lista de partes. (Ver Figura 35)



Figura 35. Página de Detalle de Partes

La pantalla Detalle de Cliente contiene la información completa correspondiente al cliente seleccionado de la lista de clientes. Al mismo tiempo, se tiene un botón de enlace "Ver órdenes" que permite acceder a la

lista de órdenes de servicio correspondiente al cliente en cuestión. (Ver Figura 36)



Figura 36. Página de Detalle de Cliente

La pantalla Detalle de Técnico contiene la información completa correspondiente al técnico seleccionado de la lista de técnicos. Se tiene un botón de enlace “Ver órdenes” que permite acceder a la lista de órdenes de servicio que han sido asignadas al técnico en cuestión. El botón “Ver Calendario” permite visualizar en un calendario las fechas que tienen órdenes programadas para el técnico seleccionado. (Ver Figura 37)



Figura 37. Página de Detalle de Técnico

### R10 - Diseño de la interfaz de la página de confirmación de nuevas órdenes

La página de confirmación de orden de servicio muestra los mismos campos del formulario de creación de orden de servicio. Pero en este caso los campos vienen por defecto con el valor que les ha sido asignado cuando se creó la orden.

El botón confirmar sirve para validar las órdenes nuevas, pudiendo modificarse la información de la orden de servicio en caso de ser necesario. El botón rechazar redirige a la página de cancelación de la orden de servicio. (Ver Figura 38)

**Confirmar Orden**

Nombre del Cliente:

RIF:

Direccion:

Nombre de Contacto:

Telefono de Contacto:

Email:

Fecha / Hora Estimada:

Tipo de Servicio:

Descripcion:

Tecnico:  
  
[pedromartinez@gmail.com](mailto:pedromartinez@gmail.com)

Figura 38. Página de confirmar orden de servicio de la aplicación web

## R11 - Programación de servicio web

La solución móvil implementada provee un acceso a la base de datos con soporte en una arquitectura orientada a servicios. El lenguaje de programación utilizado para el desarrollo de los servicios web es PHP.

Con el fin de reducir el tiempo de desarrollo de los servicios se utilizó Slim, un framework de código abierto bastante ligero y potente, orientado

especialmente a servicios bajo especificación REST. En la Figura 39 se puede apreciar un fragmento de código de la implementación de los servicios web.

```
<?php
require 'db.php';
require 'Slim/Slim.php';
\Slim\Slim::registerAutoloader();

$app = new \Slim\Slim();

$app->get('/login/:usuario/:password', 'loginUsuario');
$app->get('/ordenes', 'getOrdenes');
$app->get('/ordenescliente/:clienrif', 'getOrdenesPorCliente');
$app->get('/ordenestecnico/:tecnico', 'getOrdenesPorTecnico');
$app->get('/ordenesestatus/:estatus/:tecnico', 'getOrdenesPorEstatus');
$app->get('/clientes', 'getClientes');
$app->get('/usuarios', 'getUsuarios');
$app->get('/perfiles', 'getPerfiles');
$app->get('/estatus', 'getEstatus');
$app->get('/usuarios/:codperfil', 'getUsuariosPorPerfil');
$app->get('/items', 'getItems');
$app->get('/ordenitems', 'getOrdenItems');
$app->get('/ordenitems/:codigo', 'getOrdenItemsPorOrden');
$app->post('/addorden', 'addOrden');
$app->post('/addordenitems', 'addOrdenItems');
$app->put('/updateorden', 'updateOrden');
$app->delete('/deleteordenitems/:codigo', 'deleteOrdenItems');

$app->response()->header("Content-Type", "application/json; Charset=UTF-8");

$app->run();
```

Figura 39. Fragmento de código del servicio web

## R12 - Pruebas de funcionalidad de versión

Finalmente, se realizaron pruebas de versión sobre el incremento del Sprint 2.

Tabla 12. Prueba de funcionalidad de acceso a la página de crear nuevas órdenes

<b>Caso de Prueba:</b> Acceso a la página de crear nuevas órdenes
<b>Requisito:</b> R7
<b>Descripción:</b> Se abre la página para crear órdenes de servicio en la aplicación.
<b>Condiciones de Ejecución:</b> El usuario debe estar en la página principal
<b>Pasos de ejecución:</b> 1. Se presiona sobre el ícono Crear orden
<b>Resultado Esperado:</b> se visualiza la página para crear nuevas órdenes de servicio
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 13. Prueba de funcionalidad de acceso a la página de lista de órdenes

<b>Caso de Prueba:</b> Acceso a la página de lista de órdenes
<b>Requisito:</b> R8
<b>Descripción:</b> Se abre la página mostrando la lista órdenes de servicio correspondiente al estatus seleccionado
<b>Condiciones de Ejecución:</b> El usuario debe estar en la página principal
<b>Pasos de ejecución:</b> 1. Se presiona sobre el ícono Nuevas, Confirmadas, En Espera, Completadas ó Canceladas
<b>Resultado Esperado:</b> se visualiza la página de la lista de órdenes de servicio correspondientes al estatus seleccionado
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 14. Prueba de funcionalidad de acceso a la página de lista de clientes

<b>Caso de Prueba:</b> Acceso a la página de lista de clientes
<b>Requisito:</b> R9
<b>Descripción:</b> Se abre la página con la lista de clientes
<b>Condiciones de Ejecución:</b> El usuario debe estar en la página principal
<b>Pasos de ejecución:</b> 1. Se presiona sobre el ícono Clientes
<b>Resultado Esperado:</b> se visualiza la página de la lista de clientes
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 15. Prueba de funcionalidad de acceso a la página de confirmación de nuevas órdenes

<b>Caso de Prueba:</b> Acceso a la página de confirmación de nuevas órdenes
<b>Requisito:</b> R10
<b>Descripción:</b> Se abre la página con la lista de clientes
<b>Condiciones de Ejecución:</b> El usuario debe estar en la página principal
<b>Pasos de ejecución:</b> 1. Se presiona sobre el ícono Nuevas 2. Se selecciona una orden de la lista
<b>Resultado Esperado:</b> se visualiza la página de confirmación de la orden de servicio con los botones para confirmar o rechazar la orden.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

El estado actualizado de la pila del producto al final del Sprint 2 se describe en la Tabla 16

**Tabla 16. Estado de la pila del producto luego del Sprint 2**

	<b>Descripción</b>	<b>Prioridad</b>	<b>Estado</b>	<b>Observación</b>
R1	Diseño y construcción base de datos móvil y web	Alta	<b>Terminada</b>	
R2	Desarrollo de casos de uso del sistema	Alta	<b>Terminada</b>	
R3	Diseño de la interfaz de la página principal	Alta	<b>Terminada</b>	
R4	Diseño de la interfaz para el login de usuario	Alta	<b>Terminada</b>	
R5	Programación de conexiones base de datos móvil	Alta	<b>Terminada</b>	
R6	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R7	Diseño interfaz página de crear nuevas órdenes	Alta	<b>Terminada</b>	
R8	Diseño interfaz de la página de lista de órdenes	Alta	<b>Terminada</b>	
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Terminada</b>	
R10	Diseño interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Terminada</b>	
R11	Programación de servicio web	Alta	<b>Terminada</b>	
R12	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R13	Diseño interfaz página cancelación orden servicio	Alta	<b>Pendiente</b>	
R14	Diseño interfaz página calendario órdenes servicio	Media	<b>Pendiente</b>	
R15	Diseño interfaz de la página de reporte de servicio	Alta	<b>Pendiente</b>	
R16	Diseño interfaz página agregar partes y materiales	Alta	<b>Pendiente</b>	
R17	Programación de interfaces de conexión con servicios web	Alta	<b>Pendiente</b>	
R18	Implementación de la interfaz de la página principal	Alta	<b>Pendiente</b>	
R19	Implementación interfaz para el login de usuario	Alta	<b>Pendiente</b>	
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Pendiente</b>	
R21	Pruebas de funcionalidad de versión	Alta	<b>Pendiente</b>	
R22	Implementación de la interfaz de la página de crear nuevas órdenes	Alta	<b>Pendiente</b>	
R23	Implementación de la interfaz de la página de lista de órdenes	Alta	<b>Pendiente</b>	
R24	Implementación de la interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Pendiente</b>	
R25	Implementación interfaz página cancelación orden	Alta	<b>Pendiente</b>	
R26	Implementación interfaz página calendario orden	Alta	<b>Pendiente</b>	
R27	Implementación interfaz página reporte servicio	Alta	<b>Pendiente</b>	
R28	Implementación interfaz agregar parte y material	Alta	<b>Pendiente</b>	
R29	Implementación de la interfaz de sincronización de la aplicación móvil	Alta	<b>Pendiente</b>	
R30	Pruebas de funcionalidad de versión	Alta	<b>Pendiente</b>	

### **Sprint 3**

Las actividades o requerimientos a ser cumplidos en el Sprint 3 corresponden a los descritos en la Tabla 17

**Tabla 17. Pila del Sprint 3**

	<b>Descripción</b>	<b>Res.</b>	<b>Prioridad</b>	<b>Est.(h)</b>
R13	Diseño de la interfaz de la página de cancelación de orden de servicio	J. C.	Alta	2
R14	Diseño de la interfaz de la página de calendario de órdenes de servicio	J. C.	Media	2
R15	Diseño de la interfaz de la página de reporte de servicio	J. C.	Alta	4
R16	Diseño de la interfaz de la página de agregar partes y materiales	J. C.	Alta	2
R17	Programación de interfaces de conexión con servicios web	M. D	Alta	12
R18	Implementación de la interfaz de la página principal	M. D	Alta	8
R19	Implementación de la interfaz para el login de usuario	M. D	Alta	4
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y	J. C.	Alta	12
R21	Pruebas de funcionalidad de versión	M. D	Alta	4

#### **R13 - Diseño de la interfaz de la página de cancelación de orden de servicio**

La página de cancelar orden de servicio es una opción disponible para cancelar órdenes que se encuentren en cualquier otro estado distinto a "Cancelada". El formulario muestra los campos motivo y nota de cancelación. (Ver Figura 40)



Figura 40. Página de cancelar orden de servicio

#### R14 - Diseño de la interfaz de la página de calendario de órdenes de servicio

En la Figura 41 se aprecia el diseño del calendario de órdenes de servicio. El mismo resalta la fecha actual y las fechas con órdenes asignadas. Esta información se limita sólo a las órdenes en estado Confirmada o En Espera.



Figura 41. Página de calendario de órdenes de servicio asignadas a un técnico

## **R15 - Diseño de la interfaz de la página de reporte de servicio**

La página de reporte de servicio se compone realmente de dos páginas: en la primera se muestra en detalle la información del cliente que solicita el servicio y en la segunda se presenta el formulario que contiene los campos que serán llenados con la información obtenida luego de la ejecución del servicio. En este formulario es posible introducir los datos de fecha y hora de comienzo del servicio, partes y materiales utilizados, horas de trabajo real, fecha y hora de finalización, estatus de la orden, comentarios adicionales. (Ver Figura 42 y Figura 43)

Si la orden en cuestión es una del listado de órdenes canceladas, el formulario mostrará todos los campos inhabilitados, como indicativo de que no es posible ejecutar ninguna otra acción de modificación sobre una orden cancelada.

En caso de que al momento de guardar la orden de servicio (con estatus distinto a Cancelada) el estatus seleccionado sea "Cancelada", la aplicación redirige al usuario a la página de cancelación de orden de servicio.



Figura 42. Página de reporte de orden de servicio en la aplicación móvil

**Reporte de Servicio**

*Descripción del Servicio*

Fecha / Hora Comienzo:

Partes/Materiales:

Horas de trabajo efectivo:

Fecha / Hora Fin:

Estatus:

Confirmada

Observaciones:

**Guardar**

**Figura 43. Página de reporte de orden de servicio en la aplicación web**

### **R16 - Diseño de la interfaz de la página de agregar partes y materiales**

La página de agregar partes y materiales permite al usuario, desde la página de reporte de servicio, agregar o eliminar partes y materiales en la orden de servicio. El formulario presentado posee un campo para el nombre del ítem a agregar y otro para indicar la cantidad requerida de esa parte o material. (Ver Figura 44)

← Agregar Partes

Parte/Material

Seleccione parte/material

Cantidad

Aceptar

Figura 44. Página de agregar partes y materiales a una orden de servicio

### R17 - Programación de interfaces de conexión con servicios web

Para el desarrollo de las interfaces de conexión con el servicio web, se consideró muy importante la idea de utilizar el framework JQuery para agilizar dicha tarea. Dicha importancia radica en la simpleza y capacidad del framework de operar de forma muy transparente tanto en el ambiente web como en el ambiente móvil. Ver Figura 45



```

function mostrarDisenoPerfiles() {
  if ( codPerfil === 'tecnico' ) {
    var ptr = $("#tableOrdenes").find("tr:first");
    ptr.find("td:first").remove();
    ptr.find("td:first").remove();
    ptr = $("#tableOrdenes").find("tr:first");
    ptr.append('<td id="tdOrdenesEnEspera">' + $("#tdOrdenesEnEspera").html() + '</td>');
    ptr.append('<td id="tdOrdenesCompletadas">' + $("#tdOrdenesCompletadas").html() + '</td>');
    ptr = $("#tableOrdenes").find("tr:last");
    ptr.find("td:first").remove();
    ptr.find("td:first").remove();
    ptr = $("#tableBusqueda").find("tr:first");
    ptr.find("td:last").html('&nbsp;');
  }
  if ( codPerfil === 'despachador' ) {
    $('#tdCrearOrden').show();
    $('#tdCrearOrdenTitulo').show();
    $('#tdOrdenesNuevas').show();
    $('#tdCountNuevas').show();
    $('#tdOrdenesActivas').hide();
    $('#tdCountActivas').hide();
    $('#divConfirmarTecnico').hide();
    $('#divSubmitConfirmarOrden').hide();
    $("#formConfirmarOrden input[name=fechaestimada]").attr('readonly',true);
    $("#formConfirmarOrden input[name=horaestimada]").attr('readonly',true);
    $("#formConfirmarOrden select").attr('readonly',true);
    $("#formConfirmarOrden textarea").attr('readonly',true);
    $("#pageConfirmarOrden div h1").text('Orden Nueva');
  }
  if ( codPerfil === 'gerente' ) {
    $('#divConfirmarTecnico').show();
    $('#divSubmitConfirmarOrden').show();
  }
}

```

Figura 46. Función para la gestión de contenido de la página principal por perfil de usuario

## R19 - Implementación de la interfaz para el login de usuario

En la Figura 47 se aprecia un fragmento de código con la función para implementar el login en la aplicación móvil

```

$('#formLogin').validate({
  messages: {
    usuario: 'Usuario es requerido',
    password: 'Password es requerido'
  },
  errorPlacement: function (error, element) {
    error.appendTo(element.parent().prev());
  },
  submitHandler: function (form) {
    db.transaction(function (tx) {
      tx.executeSql("SELECT password,codperfil,nombre,email,deleted FROM Usuario
        if ( results.rows.length === 0 ) {
          reset();
          alertify.error('Usuario o password incorrecto');
        } else {
          var row = results.rows.item(0);
          codPerfil = row.codperfil;
          usuario = row.usuario;
          $('#usuariomain').html(row.nombreusuario);
          mostrarDisenoPerfiles();
          mostrarCantidadOrdenes();
          $('#btnConfigurar').show();
          $.mobile.changePage("#main");
        }
      });
    });
    return false;
  }
});

```

Figura 47. Función que implementa la autenticación de usuarios en la aplicación móvil

## R20 - Implementación de la interfaz de la página de lista de clientes

En la Figura 48 se aprecia un fragmento de código con la función para implementar la lista de clientes en la aplicación móvil

```

function mostrarListaClientes(action) {
$.mobile.changePage ($("#pageListaClientes"), { transition : "slide", reverse : false });
$("#listaClientes").empty();
db.transaction(function(transaction) {
transaction.executeSql('SELECT * FROM Cliente;', [],
function(transaction, result) {
if (result !== null && result.rows !== null) {
for (var i = 0; i < result.rows.length; i++) {
var row = result.rows.item(i);
$("#listaClientes").append('<li id="' + action + '_' + row.rif + '_' + row.nombre + '</span></br>' +
'<span>' + row.direccion + '</span></br>' +
'</a></li>');
$("#listaClientes").listview('refresh');
}
}
});
});
if (action==='details'){
$("#radioGroupVerSeleccionarC").hide();
$("#pageListaClientes div[data-role='header'] a").attr("href", "#main");
} else {
$("#radioGroupVerSeleccionarC").show();
$("#pageListaClientes div[data-role='header'] a").attr("href", "#pageCrearOrden");
}
}
}

```

Figura 48. Función que implementa la lista de clientes en la aplicación móvil

## R21 - Pruebas de funcionalidad de versión

Finalmente, se realizaron las pruebas de funcionalidad de versión sobre el incremento del Sprint 3.

Tabla 18. Prueba de acceso a la página de cancelación de órdenes de servicio

<b>Caso de Prueba:</b> Acceso a la página de cancelación de órdenes de servicio
<b>Requisito:</b> R13
<b>Descripción:</b> Se abre la página de cancelación de órdenes de servicio
<b>Condiciones de Ejecución:</b> El usuario debe estar en la página principal. Deben existir órdenes nuevas, confirmadas, en espera o completadas.
<p><b>Pasos de ejecución (Caso 1):</b></p> <ol style="list-style-type: none"> <li>1. Presionar el ícono Nuevas</li> <li>2. Seleccionar una orden de la lista</li> <li>3. Se presiona el botón rechazar</li> </ol> <p><b>Pasos de ejecución (Caso 2):</b></p> <ol style="list-style-type: none"> <li>1. Presionar el ícono Confirmadas, En Espera o Completadas.</li> <li>2. Seleccionar una orden de la lista</li> <li>3. Presionar el botón siguiente</li> <li>4. Seleccionar la opción Cancelar del campo estatus</li> <li>5. Presionar el botón guardar</li> </ol>
<b>Resultado Esperado:</b> se visualiza la página de cancelación de la orden de servicio.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 19. Prueba de acceso a la página de calendario de órdenes de servicio

<b>Caso de Prueba:</b> Acceso a la página de calendario de órdenes de servicio
<b>Requisito:</b> R14
<b>Descripción:</b> Se abre la página de calendario de órdenes de servicio
<b>Condiciones de Ejecución:</b> Deben existir órdenes nuevas, confirmadas, en espera o completadas.
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Presionar el ícono Técnicos</li> <li>2. Seleccionar un técnico de la lista</li> <li>3. Presionar el botón Ver calendario</li> </ol>
<b>Resultado Esperado:</b> se visualiza la página de calendario de órdenes de servicio.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 20. Prueba de acceso a la página de reporte de órdenes de servicio

<b>Caso de Prueba:</b> Acceso a la página de reporte de órdenes de servicio
<b>Requisito:</b> R15
<b>Descripción:</b> Se abre la página de reporte de órdenes de servicio
<b>Condiciones de Ejecución:</b> Deben existir órdenes nuevas, confirmadas, en espera o completadas.
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Presionar el ícono Confirmadas, En Espera o Completadas</li> <li>2. Seleccionar una orden de la lista</li> <li>3. Navegar adelante o atrás entre el detalle del cliente y el detalle del reporte</li> </ol>
<b>Resultado Esperado:</b> se visualiza la página de reporte de órdenes de servicio.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 21. Prueba de acceso a la página de agregar partes y materiales

<b>Caso de Prueba:</b> Acceso a la página de agregar partes y materiales
<b>Requisito:</b> R16
<b>Descripción:</b> Se abre la página de agregar partes y materiales
<b>Condiciones de Ejecución:</b> Deben existir órdenes confirmadas, en espera o completadas.
<b>Pasos de ejecución:</b>  <ol style="list-style-type: none"><li>1. Presionar el ícono Confirmadas, En Espera o Completadas</li><li>2. Seleccionar una orden de la lista</li><li>3. Navegar adelante hacia la sección de descripción del servicio</li><li>4. Presionar el ícono de agregar partes y materiales</li></ol>
<b>Resultado Esperado:</b> se visualiza la página de agregar partes y materiales.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

El resultado de la pila del producto al final del Sprint se describe en la Tabla 22.

Tabla 22. Estado de la pila del producto luego del Sprint 3

Descripción		Prioridad	Estado	Observación
R1	Diseño y construcción base de datos móvil y web	Alta	Terminada	
R2	Desarrollo de casos de uso del sistema	Alta	Terminada	
R3	Diseño de la interfaz de la página principal	Alta	Terminada	
R4	Diseño de la interfaz para el login de usuario	Alta	Terminada	
R5	Programación de conexiones base de datos móvil	Alta	Terminada	
R6	Pruebas de funcionalidad de versión	Alta	Terminada	
R7	Diseño interfaz página de crear nuevas órdenes	Alta	Terminada	
R8	Diseño interfaz de la página de lista de órdenes	Alta	Terminada	
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	Terminada	
R10	Diseño interfaz de la página de confirmación de nuevas órdenes	Alta	Terminada	
R11	Programación de servicio web	Alta	Terminada	
R12	Pruebas de funcionalidad de versión	Alta	Terminada	
R13	Diseño interfaz página cancelación orden servicio	Alta	Terminada	
R14	Diseño interfaz página calendario órdenes servicio	Media	Terminada	
R15	Diseño interfaz de la página de reporte de servicio	Alta	Terminada	
R16	Diseño interfaz página agregar partes y materiales	Alta	Terminada	
R17	Programación de interfaces de conexión con servicios web	Alta	Terminada	
R18	Implementación de la interfaz de la página principal	Alta	Terminada	
R19	Implementación interfaz para el login de usuario	Alta	Terminada	
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	Terminada	
R21	Pruebas de funcionalidad de versión	Alta	Terminada	
R22	Implementación de la interfaz de la página de crear nuevas órdenes	Alta	Pendiente	
R23	Implementación de la interfaz de la página de lista de órdenes	Alta	Pendiente	
R24	Implementación de la interfaz de la página de confirmación de nuevas órdenes	Alta	Pendiente	
R25	Implementación interfaz página cancelación orden	Alta	Pendiente	
R26	Implementación interfaz página calendario orden	Alta	Pendiente	
R27	Implementación interfaz página reporte servicio	Alta	Pendiente	
R28	Implementación interfaz agregar parte y material	Alta	Pendiente	
R29	Implementación de la interfaz de sincronización de la aplicación móvil	Alta	Pendiente	
R30	Pruebas de funcionalidad de versión	Alta	Pendiente	

#### **Sprint 4**

Las actividades o requerimientos a ser cumplidos en el Sprint 4 corresponden a los descritos en la Tabla 23.

**Tabla 23. Pila del Sprint 4**

<b>Descripción</b>		<b>Res.</b>	<b>Prioridad</b>	<b>Est.(h)</b>
R22	Implementación de la interfaz de crear órdenes	M. R.	Alta	2
R23	Implementación de la interfaz de la lista de órdenes	M. R.	Alta	2
R24	Implementación de la interfaz de confirmación de nuevas órdenes	M. R.	Alta	4
R25	Implementación de la interfaz de cancelación de órdenes	M. R.	Alta	2
R26	Implementación de la interfaz de calendario de órdenes	J. C.	Alta	12
R27	Implementación de la interfaz de reporte de servicio	J. C.	Alta	8
R28	Implementación de la interfaz de agregar partes y materiales	J. C.	Alta	4
R29	Implementación de la interfaz de sincronización de la aplicación móvil	M. R.	Alta	4
R30	Pruebas de funcionalidad de versión	M.R. J.C.	Alta	4

## **R22 - Implementación de la interfaz de crear nuevas órdenes**

En la Figura 49 se aprecia un fragmento de código con la función para implementar la creación de órdenes de servicio

```
var validatorFormCrearOrden = $('#formCrearOrden').validate({
  messages: {
    cliente: "Cliente es obligatorio.",
    fechahoracrear: "Fecha/Hora Estimada es obligatorio.",
    tiposervicio: "Tipo de servicio es obligatorio.",
    descripcion: "Descripcion es obligatorio.",
    tecnicocrear: "Tecnico es obligatorio."
  },
  errorPlacement: function (error, element) {
    if ($(element).is('select')) {
      error.appendTo(element.parent().parent().prev()); // special
    } else if ($(element).is('textarea')) {
      error.appendTo(element.prev()); // default placement for eve
    } else {
      error.appendTo(element.parent().prev()); // default placemen
    }
  },
  submitHandler: function (form) {
    $.ajax({
      type: 'GET',
      url: rootURL + "idnuevaorden",
      cache: false,
      timeout: 5000,
      success: function(data) {
        if ( data.id!==null && data.id.length!==0 ) {
          var codigo = data.id[0].codigo;
          var tempCodigo = parseInt(codigo.substring(2)) + 1;
          crearOrden("SO"+tempCodigo);
        }
      },
      error: function (jqXHR, textStatus, errorThrown) {
```

Figura 49. Fragmento de código de función para crear orden de servicio

## R23 - Implementación de la interfaz de lista de órdenes

En la Figura 50 se aprecia un fragmento de código con la función para implementar la lista de órdenes de servicio

```
function queryListaOrdenes(executeQuery) {
    $('#listaOrdenes').empty();
    $.ajax({
        type: 'GET',
        url: rootURL + executeQuery,
        dataType: "json",
        success: function(data) {
            if ( data.ordenes!=null && data.ordenes.length!=0 ) {
                arrayOrdenes = data.ordenes;
                $.each( data.ordenes, function( index, orden ) {
                    $('#listaOrdenes').append('<li id="' + orden.codigo + "&" + orden.cliente.rif + "&" +
                        '<span class="listRow1">'+orden.cliente.nombre+'</span></br>'+
                        '<span>'+orden.tiposervicio+'</span></br>'+
                        '<span>'+orden.fechaestimada+'&nbsp;&nbsp;&nbsp;'+orden.horaestimada+'</span></br>'+
                        '<span class="listRow3">'+orden.estatus+'</span></br>'+
                        '</a></li>');
                    $('#listaOrdenes').listview('refresh');
                });
            }
        }
    });
    $.mobile.changePage ($("#pageListaOrdenes"), { transition: "slide", reverse: false });
    if ((prevPage=="pageDetalleCliente") || (prevPage=="pageDetalleTecnico")) {
        $("#pageListaOrdenes div[data-role='header'] a").attr("href", "#"+prevPage);
    }
}
```

Figura 50. Fragmento de código de función para listar las órdenes de servicio

## R24 - Implementación de la interfaz de confirmación de nuevas órdenes

En la Figura 51 se aprecia un fragmento de código con la función para implementar la confirmación de nuevas órdenes de servicio

```
// Confirmar Orden - Editar orden nueva y asignar tecnico
$('#formConfirmarOrden').validate({
  messages: {
    cliente2: "Cliente es obligatorio.",
    fechahoraconfirmar: "Fecha/Hora estimada es obligatorio.",
    tiposervicio2: "Tipo de servicio es obligatorio.",
    descripcion2: "Descripcion es obligatorio.",
    tecnico: "Tecnico es obligatorio."
  },
  errorPlacement: function (error, element) {
    if ($(element).is('select')) {
      error.appendTo(element.parent().parent().prev()); // special placemen
    } else if ($(element).is('textarea')) {
      error.appendTo(element.prev()); // default placement for everything
    } else {
      error.appendTo(element.parent().prev()); // default placement for ev
    }
  },
  submitHandler: function (form) {
    var fechahoraestimada = $("#fechahoraconfirmar").val().split(" ");
    var fechaestimada = fechahoraestimada[0];
    var horaestimada = fechahoraestimada[1] + " " + fechahoraestimada[2];
    var formData = "codigo="+idOrden +
      "&clnterif=" + $('#clnterif2').val() +
      "&tecnico=" + $('#hiddenTecnico').val() +
      "&estatus=" + estatusConfirmar +
      "&fechaestimada=" + fechaestimada +
      "&horaestimada=" + horaestimada +
      "&descripcion=" + $('#descripcion2').val() +
      "&tiposervicio=" + $('#tiposervicio2').val();
    actualizarOrden(formData);
  }
});
```

Figura 51. Fragmento de código de función para confirmar una nueva orden de servicio

## R25 – Implementación de la interfaz de cancelación de órdenes

En la Figura 52 se aprecia un fragmento de código con la función para implementar la cancelación de órdenes de servicio

```
var validatorFormCancelarOrdenNueva = $('#formCancelarOrdenNueva').validate({
  messages: {
    motivoCancelarOrden: "Campo Motivo es requerido.",
    notaCancelarOrden: "Campo Nota es requerido."
  },
  errorPlacement: function (error, element) {
    if ($(element).is('select')) {
      error.appendTo(element.parent().parent().prev()); // special place
    } else {
      error.appendTo(element.prev()); // default placement for everythi
    }
  },
  submitHandler: function (form) {
    var formData = "codigo="+idOrden +
      "clienrif="+ clienteOrdenActual +
      "tecnico=" + tecnicoOrdenActual +
      "estatus=" + estatusOrdenActual +
      "&motivocancelada=" + $('#motivoCancelarOrden').val() +
      "&notacancelada=" + $('#notaCancelarOrden').val();
    actualizarOrden(formData);
  }
});
```

Figura 52. Fragmento de código de función para cancelación de una orden de servicio

## R26 - Implementación de la interfaz de calendario de órdenes

En la Figura 53 se aprecia un fragmento de código con la función para implementar el calendario de órdenes de servicio

```
$("#linkCalendarioOrdenes").on('click', function (event){
    $("#divNombreTecnicoCalendario").text(nombreTecnico);
    $.ajax({
        type: 'GET',
        url: rootURL + 'ordenestecnico/' + usuarioTecnico,
        dataType: "json",
        success: function(data){
            if ( data.ordenes!=null && data.ordenes.length!=0 ) {
                arrayOrdenes = data.ordenes;
                eventsArray.length = 0;
                $('#calendar').trigger('refresh');
                for( var i=0; i<arrayOrdenes.length; i++){
                    // Dynamically add events to the array (this can be done later on as well)
                    if ( (arrayOrdenes[i].estatus === 'Confirmada') || (arrayOrdenes[i].estatus === 'En Espera') ) {
                        var nota = arrayOrdenes[i].horaestimada + " - " + arrayOrdenes[i].estatus + " - " + array
                        var arrayFecha = arrayOrdenes[i].fechaestimada.split('/');
                        var dd = arrayFecha[0];
                        var mm = arrayFecha[1];
                        var yyyy = arrayFecha[2];
                        eventsArray.push({ "summary" : nota, "begin" : new Date(yyyy,mm-1, dd ), "end" : new Date(
                    }
                }
                // Trigger refresh
                $('#calendar').trigger('refresh');
            }
        }
    });
});
```

Figura 53. Fragmento de código de función para el calendario de órdenes de servicio

## R27 - Implementación de la interfaz de reporte de servicio

En la Figura 54 se aprecia un fragmento de código con la función para implementar el reporte ejecución de órdenes de servicio

```
function submitReporteOrden() {
    var result = $.grep(arrayOrdenes, function(e){ return e.codigo === idOrden; });
    var ordenActual = result[0];
    var fechahoracomienzo = $("#horacomienzo").val().split(" ");
    var fechacomienzo = fechahoracomienzo[0];
    var horacomienzo = fechahoracomienzo[1] + " " + fechahoracomienzo[2];
    var fechahorafin = $("#horafin").val().split(" ");
    var fechafin = fechahorafin[0];
    var horafin = fechahorafin[1] + " " + fechahorafin[2];
    //Agregar cambio de estatus a comentarios
    var comentarios = $('#comentarios').val();
    $('#comentarios').val(getFechaActual() + " - " + getHoraActual() + "\nEstatus: " + estatus + "\n" + comentarios);
    var formData = "codigo=" + idOrden +
        "&clnterif=" + clienteOrdenActual +
        "&tecnico=" + tecnicoOrdenActual +
        "&estatus=" + estatus +
        "&fechaestimada=" + ordenActual.fechaestimada +
        "&fechacomienzo=" + fechacomienzo +
        "&fechafin=" + fechafin +
        "&horaestimada=" + ordenActual.horaestimada +
        "&shoracomienzo=" + horacomienzo +
        "&shorafin=" + horafin +
        "&tiposervicio=" + ordenActual.tiposervicio +
        "&descripcion=" + ordenActual.descripcion +
        "&comentarios=" + $('#comentarios').val() +
        "&horastrabajo=" + $('#horasTrabajo').val() +
        "&imgcamara=" + encodeURIComponent($('#attachedImage').attr('src')) +
        "&imgfirma=" + encodeURIComponent($('#saveSignature').attr('src')) +
        "&motivocancelada=" +
        "&notacancelada=" +
```

Figura 54. Fragmento de código de función para el reporte de ejecución órdenes de servicio

## R28 - Implementación de la interfaz de agregar partes y materiales

En la Figura 55 se aprecia un fragmento de código con la función para agregar partes y/o materiales a una orden de servicio

```
$('#formAddItems').validate({
  messages: {
    descItem: "Item es obligatorio.",
    cantidadItem: "Cantidad es obligatorio."
  },
  errorPlacement: function (error, element) {
    if ($(element).is('select')) {
      error.appendTo(element.parent().parent().prev()); // special placement for select elements
    } else {
      error.appendTo(element.parent().prev()); // default placement for everything else
    }
  },
  submitHandler: function (form) {
    //Revisar si el item ya está en lista
    var idItemPrevious;
    $('#listaAddedItems li').each(function() {
      var actin_id = this.id.split('&');
      idItemPrevious = actin_id[0];
      if (idItemPrevious === codigoItem) {
        $(this).remove();
      }
    });
    //Agregar el item a la lista
    var addstr = '<li id="' + codigoItem + '&' + descItem + '&' + $('#cantidadItem').val() + '&' + precio +
      '<span class="ui-li-count">'+$('#cantidadItem').val()+'</span></a><a href="#" id="delbtn-'+codi
    $('#listaAddedItems').append(addstr);
    $('#delbtn-'+codigoItem).button('refresh');
    $('#listaAddedItems').listview('refresh');
  });
});
```

Figura 55. Fragmento de código de función para agregar partes y/o materiales a una orden de servicio

## R29 - Implementación de la interfaz de sincronización de la aplicación móvil

En la Figura 56 se aprecia un fragmento de código con la función para sincronizar órdenes de servicio desde el servidor hacia la aplicación móvil

```

function sincronizarOrdenesDesdeServidor() {
    $.ajax({
        type: 'GET',
        url: rootURL + "ordenes",
        async: false,
        cache: false,
        timeout: 5000,
        beforeSend: function () { loading('show'); },
        success: function(data){
            if ( data.ordenes!=null && data.ordenes.length!=0 ) {
                var executeQuery = queryBase;
                db.transaction(function(tx) {
                    tx.executeSql(executeQuery, [],
                        function(transaction, result) {
                            if (result !== null && result.rows !== null) {
                                arrayOrdenes = [];
                                for (var i = 0; i < result.rows.length; i++) {
                                    insertarArrayOrdenes(result.rows.item(i));
                                }
                                $.each( data.ordenes, function( index, orden ){
                                    var result = $.grep(arrayOrdenes, function(e){ return e
                                    var modif = 0;
                                    if ( result.length===0 ) {
                                        modif = 1; // Orden en servidor pero no en cliente
                                    } else {
                                        modif = dateTimeCompare(orden.fechamodificada, resu.
                                    }
                                    // Si la orden ha sido modificada en el servidor o no
                                    if ( modif === 1 ) {
                                        db.transaction(function(tx) {

```

Figura 56. Fragmento de código de función para sincronizar desde el servidor hacia la aplicación móvil

### R30 - Pruebas de funcionalidad de versión

Finalmente, se realizaron pruebas de versión sobre el incremento del Sprint 4.

Tabla 24. Prueba de creación de órdenes de servicio

<b>Caso de Prueba:</b> Creación de órdenes de servicio
<b>Requisito:</b> R22
<b>Descripción:</b> Se crea una orden de servicio y se guarda en el sistema
<b>Condiciones de Ejecución:</b> El usuario debe tener perfil de despachador o gerente
<b>Pasos de ejecución:</b> 1. Presionar el ícono Crear orden 2. Ingresar los datos para completar el formulario de Nueva Orden. 3. Presionar el botón Aceptar para guardar la orden 4. Verificar la orden creada en la página principal con estatus Nueva
<b>Resultado Esperado:</b> se crea una nueva orden de servicio y se guarda en el sistema.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 25. Prueba de mostrar lista de órdenes de servicio

<b>Caso de Prueba:</b> Mostrar lista de órdenes de servicio
<b>Requisito:</b> R23
<b>Descripción:</b> muestra una lista con las órdenes de servicio registradas en el sistema
<b>Condiciones de Ejecución:</b> Deben haber órdenes de servicio registradas para el tipo de orden seleccionado
<b>Pasos de ejecución:</b> 1. Presionar alguno de los íconos Nuevas, Confirmadas, En Espera, Completadas, Canceladas 2. Seleccionar una de las órdenes de lista 3. Navegar entre la información del cliente y la descripción del servicio
<b>Resultado Esperado:</b> despliegue de la lista de órdenes de servicio con acceso a su información detallada
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 26. Prueba de confirmación de órdenes de servicio

<b>Caso de Prueba:</b> Confirmación de órdenes de servicio
<b>Requisito:</b> R24
<b>Descripción:</b> Se confirma o rechaza una orden de servicio nueva
<b>Condiciones de Ejecución:</b> El usuario debe tener perfil de gerente. Debe haber órdenes de servicio Nuevas.
<b>Pasos de ejecución:</b> 1. Presionar el ícono Nuevas 2. Seleccionar una orden de la lista 3. Modificar los datos en los campos de la orden de ser necesario 4. Seleccionar Confirmar 5. Verificar que la orden se ha agregado a la lista de Confirmadas
<b>Resultado Esperado:</b> la orden de servicio cambia de estatus a Confirmada y se agrega a la lista de Confirmadas.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 27. Prueba de cancelación de órdenes de servicio

<b>Caso de Prueba:</b> Cancelación de órdenes de servicio
<b>Requisito:</b> R25
<b>Descripción:</b> Se cancela una orden de servicio nueva
<b>Condiciones de Ejecución:</b> El usuario debe tener perfil de gerente. Debe haber órdenes de servicio Nuevas.
<b>Pasos de ejecución:</b> 1. Presionar el ícono Nuevas 2. Seleccionar una orden de la lista 3. Modificar los datos en los campos de la orden de ser necesario 4. Seleccionar Rechazar 5. Verificar que la orden se ha agregado a la lista de Canceladas
<b>Resultado Esperado:</b> la orden de servicio cambia de estatus a Confirmada y se agrega a la lista de Confirmadas.
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 28. Prueba de mostrar calendario de órdenes de servicio

<b>Caso de Prueba:</b> Mostrar calendario de órdenes de servicio
<b>Requisito:</b> R26
<b>Descripción:</b> Se muestra el calendario de órdenes de servicio asociadas a un técnico
<b>Condiciones de Ejecución:</b> El usuario debe tener perfil de gerente.
<b>Pasos de ejecución:</b> 1. Presionar el ícono Técnicos 2. Seleccionar un técnico de la lista 3. Presionar el ícono Ver Calendario 4. Seleccionar Rechazar 5. Seleccionar las fechas del calendario resaltadas para ver órdenes del día
<b>Resultado Esperado:</b> se muestra el calendario de órdenes de servicio asociadas a un técnico, con detalle de las órdenes registradas por día
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 29. Prueba de reporte de ejecución de órdenes de servicio

<b>Caso de Prueba:</b> Reporte de ejecución de órdenes de servicio
<b>Requisito:</b> R27
<b>Descripción:</b> Se muestra el reporte de ejecución de la orden de servicio y se guarda en el sistema
<b>Condiciones de Ejecución:</b> El usuario debe tener perfil de técnico o gerente. Debe haber órdenes registradas para el tipo de orden seleccionado
<b>Pasos de ejecución:</b> 1. Presionar alguno de los íconos Confirmadas, En Espera o Completadas. 2. Seleccionar una orden de la lista 3. Ingresar los datos correspondientes en cada uno de los campos 4. Presionar el botón Guardar
<b>Resultado Esperado:</b> se actualiza la orden de servicio en el sistema
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 30. Prueba de agregar partes y/o materiales a una orden de servicio

<b>Caso de Prueba:</b> Agregar partes y/o materiales a una orden de servicio
<b>Requisito:</b> R28
<b>Descripción:</b> Se agregan partes y/o materiales a una orden de servicio
<b>Condiciones de Ejecución:</b> Debe haber órdenes registradas para el tipo de orden seleccionado
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Presionar alguno de los íconos Confirmadas, En Espera o Completadas.</li> <li>2. Seleccionar una orden de la lista</li> <li>3. Presionar el botón para agregar partes y/o materiales</li> <li>4. En el campo Parte/Material, seleccionar un ítem de la lista de Partes/Materiales</li> <li>5. Ingresar la cantidad por unidad de medida de la parte/material seleccionado</li> <li>6. Seleccionar el botón Aceptar</li> </ol>
<b>Resultado Esperado:</b> se agrega un nuevo ítem a la lista de Partes/Materiales utilizados en el reporte de la orden servicio
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

Tabla 31. Prueba de sincronización de órdenes de servicio

<b>Caso de Prueba:</b> Sincronización de órdenes de servicio
<b>Requisito:</b> R29
<b>Descripción:</b> Se sincronizan las órdenes de servicio en el sistema
<b>Condiciones de Ejecución:</b> Debe haber órdenes registradas en el sistema
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Desde la página principal, presionar el ícono Sincronizar</li> </ol>
<b>Resultado Esperado:</b> se sincronizan las órdenes de servicio entre la aplicación móvil y el servidor
<b>Evaluación de la prueba:</b> Muy Buena. Se cumplió con el resultado esperado

El resultado de la pila del producto al final del Sprint es como se describe en la Tabla 32.

**Tabla 32. Estado de la pila del producto luego del Sprint 4**

	<b>Descripción</b>	<b>Prioridad</b>	<b>Estado</b>	<b>Observación</b>
R1	Diseño y construcción base de datos móvil y web	Alta	<b>Terminada</b>	
R2	Desarrollo de casos de uso del sistema	Alta	<b>Terminada</b>	
R3	Diseño de la interfaz de la página principal	Alta	<b>Terminada</b>	
R4	Diseño de la interfaz para el login de usuario	Alta	<b>Terminada</b>	
R5	Programación de conexiones base de datos móvil	Alta	<b>Terminada</b>	
R6	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R7	Diseño interfaz página de crear nuevas órdenes	Alta	<b>Terminada</b>	
R8	Diseño interfaz de la página de lista de órdenes	Alta	<b>Terminada</b>	
R9	Diseño de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Terminada</b>	
R10	Diseño interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Terminada</b>	
R11	Programación de servicio web	Alta	<b>Terminada</b>	
R12	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R13	Diseño interfaz página cancelación orden servicio	Alta	<b>Terminada</b>	
R14	Diseño interfaz página calendario órdenes servicio	Media	<b>Terminada</b>	
R15	Diseño interfaz de la página de reporte de servicio	Alta	<b>Terminada</b>	
R16	Diseño interfaz página agregar partes y materiales	Alta	<b>Terminada</b>	
R17	Programación de interfaces de conexión con servicios web	Alta	<b>Terminada</b>	
R18	Implementación de la interfaz de la página principal	Alta	<b>Terminada</b>	
R19	Implementación interfaz para el login de usuario	Alta	<b>Terminada</b>	
R20	Implementación de la interfaz de las páginas de lista de clientes, partes y técnicos	Alta	<b>Terminada</b>	
R21	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	
R22	Implementación de la interfaz de la página de crear nuevas órdenes	Alta	<b>Terminada</b>	
R23	Implementación de la interfaz de la página de lista de órdenes	Alta	<b>Terminada</b>	
R24	Implementación de la interfaz de la página de confirmación de nuevas órdenes	Alta	<b>Terminada</b>	
R25	Implementación interfaz página cancelación orden	Alta	<b>Terminada</b>	
R26	Implementación interfaz página calendario orden	Alta	<b>Terminada</b>	
R27	Implementación interfaz página reporte servicio	Alta	<b>Terminada</b>	

R28	Implementación interfaz agregar parte y material	Alta	<b>Terminada</b>	
R29	Implementación de la interfaz de sincronización de la aplicación móvil	Alta	<b>Terminada</b>	
R30	Pruebas de funcionalidad de versión	Alta	<b>Terminada</b>	

### **Pruebas de aceptación:**

Una vez verificado a través de las pruebas de funcionalidad que la aplicación funciona de una forma correcta y esperada, se hace necesario realizar pruebas con el usuario final.

Sánchez (2015) afirma que en las pruebas de aceptación, el cliente o usuario final prueba el software y verifica que cumpla con sus expectativas. Estas pruebas también se conocen como de “caja negra” porque se evalúan sólo las entradas que se realizan sobre el sistema y las salidas que se producen, sin importar a implementación interna.

Al final de estas pruebas, se termina de definir la calidad del producto final de software y el equipo de desarrollo puede conocer en qué grado supo interpretar los requerimientos del cliente.

Para la solución móvil desarrollada, se realizó una encuesta a usuarios finales para conocer sus impresiones acerca del sistema. La encuesta se aplicó a un grupo de diez (10) profesionales del área de servicio técnico de computación.

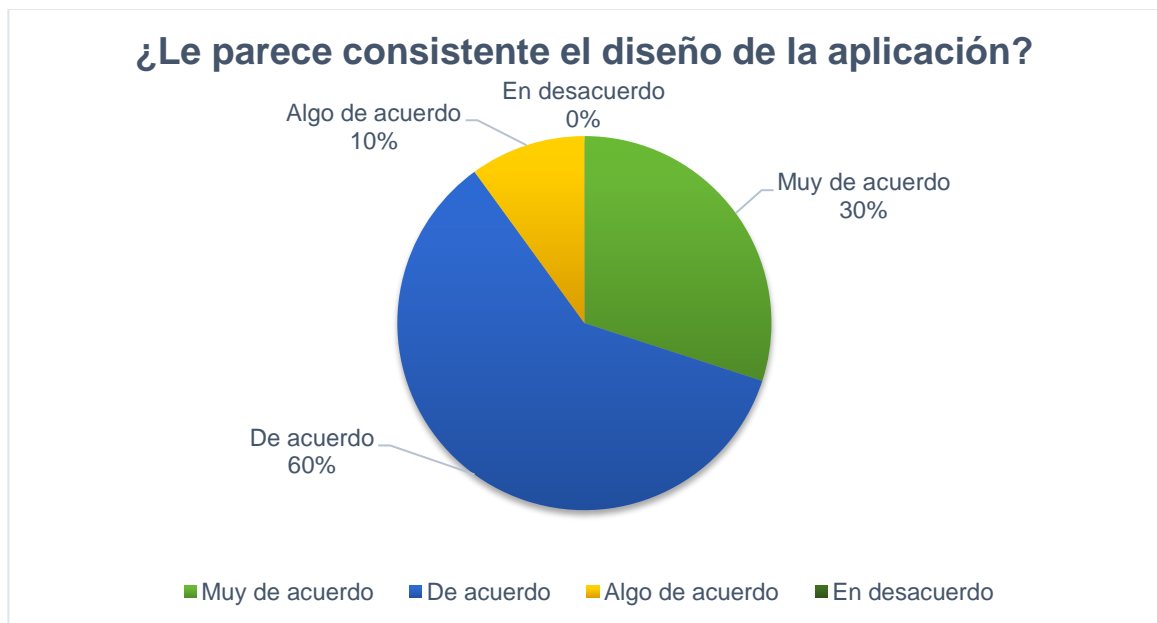
Las preguntas que componen la encuesta son las siguientes:

1. ¿Le parece consistente el diseño de la aplicación?
2. ¿La interfaz de usuario le parece apropiada?
3. ¿Le resulta fácil navegar a través de las distintas pantallas?
4. ¿Le resulta eficiente el proceso para reportar una orden de servicio?
5. ¿Le parece rápido el proceso de sincronización de órdenes de servicio?
6. ¿Considera sencilla la configuración de la aplicación?
7. ¿En términos generales, le resulta atractivo el sistema?

Las opciones disponibles como respuesta a estas preguntas son:

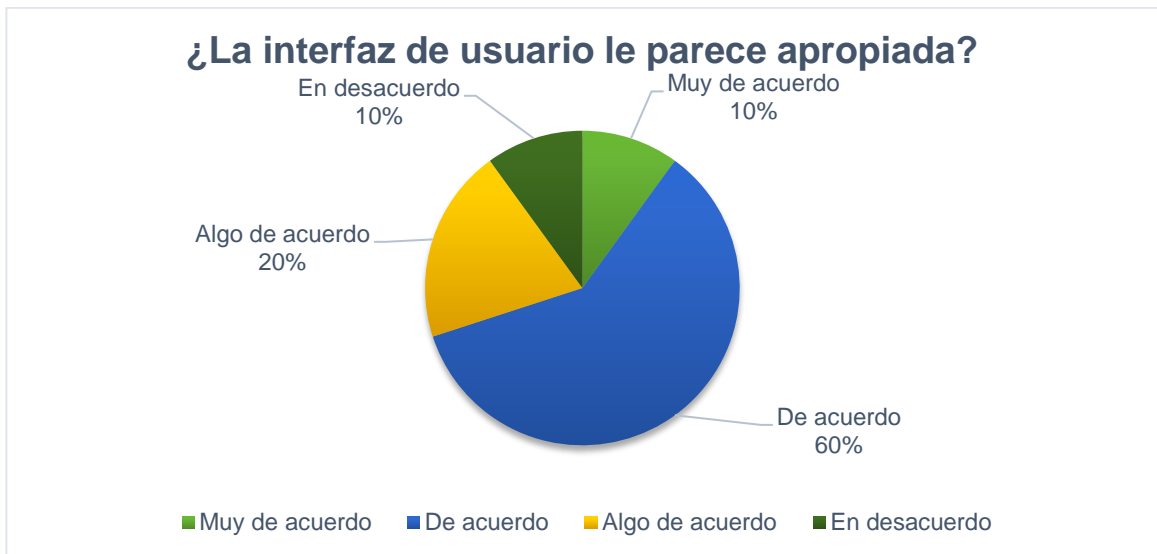
- Muy de acuerdo
- De acuerdo
- Algo de acuerdo
- En desacuerdo

A continuación se exponen los resultados obtenidos a partir de la realización de la encuesta de aceptación. Para la primera pregunta se obtuvieron los resultados mostrados en la figura 57, el sesenta por ciento (60%) manifestó estar “De acuerdo” con esta pregunta, un diez por ciento (10%) “Algo de acuerdo” y un treinta por ciento (30%) estuvo “Muy de acuerdo”. Los resultados en el grafico siguiente. (Ver figura 57)



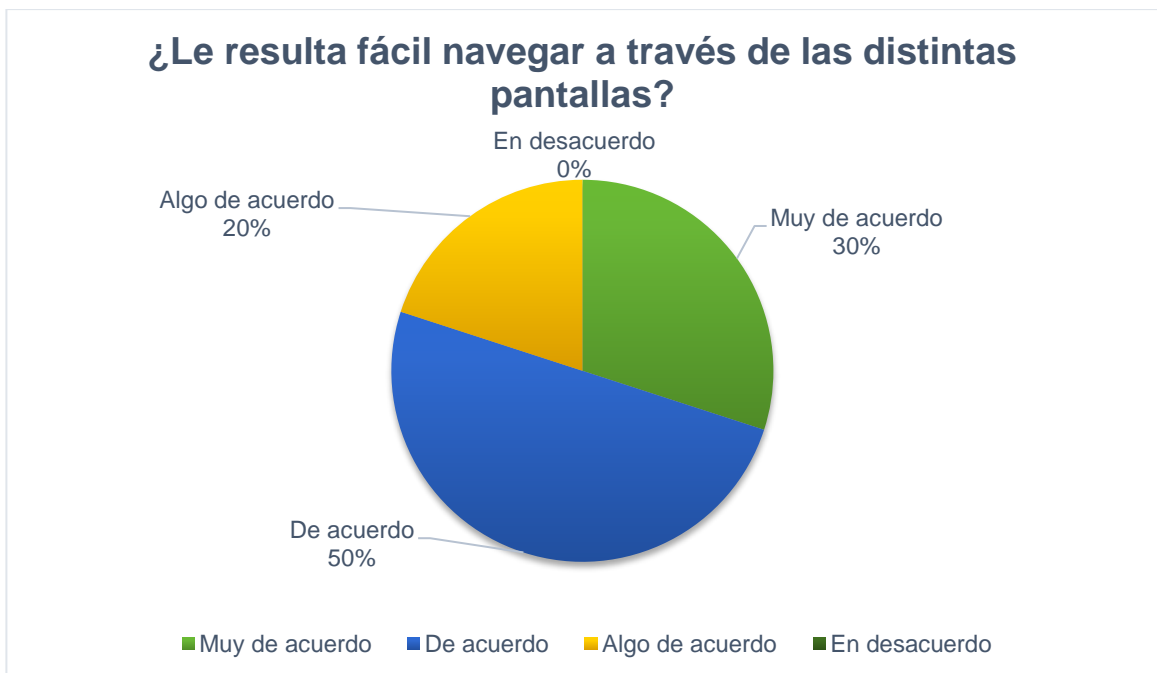
**Figura 57: Encuesta– Pregunta 1.**

Para la segunda pregunta se obtuvieron los resultados mostrados en la figura 58, un sesenta por ciento (60%) estuvo “De acuerdo” que la interfaz le parece apropiada, un diez por ciento (10%) estuvo “Muy de acuerdo”, un veinte por ciento (20%) “Algo de acuerdo” y un diez por ciento (10%) estuvo en desacuerdo. (Ver Figura 58)



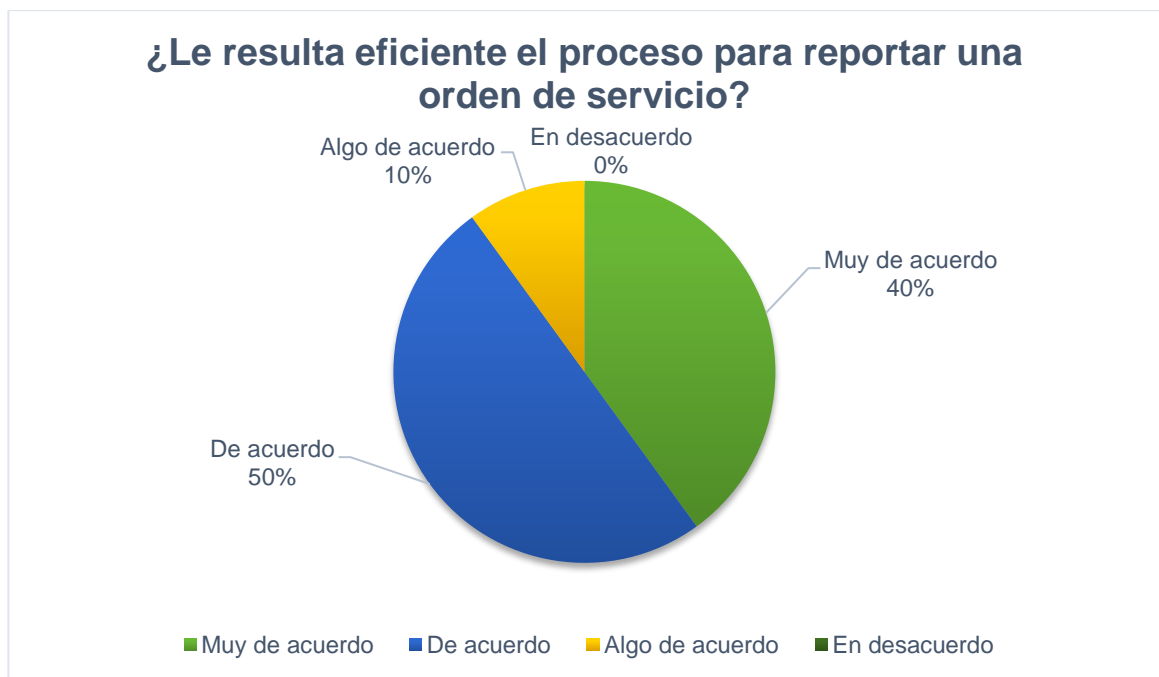
**Figura 58: Encuesta– Pregunta 2.**

Al analizar los resultados de la tercera pregunta se observa, un cincuenta por ciento (50%) estuvo “De acuerdo” que la navegación a través de las distintas pantallas le resultó fácil, un veinte por ciento (20%) “Algo de acuerdo” y un treinta por ciento (30%) estuvo “Muy de acuerdo”. (Ver Figura 59)



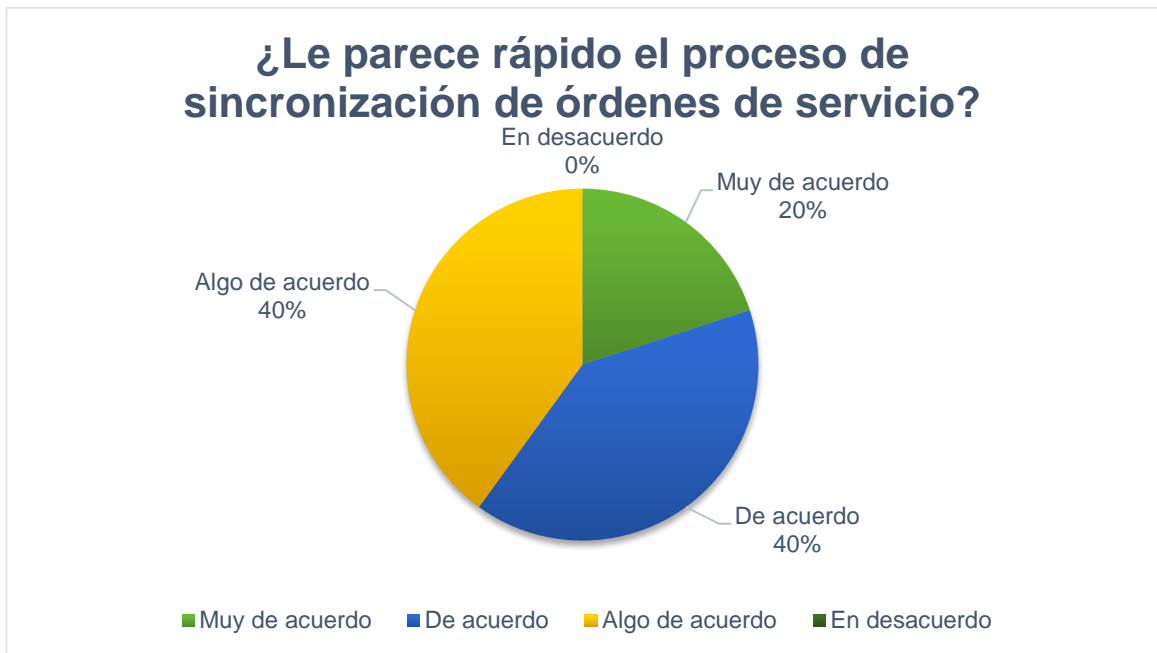
**Figura 59: Encuesta– Pregunta 3.**

En la cuarta pregunta se observa una tendencia igual a la pregunta anterior, un cincuenta por ciento (50%) estuvo “De acuerdo” que la navegación a través de las distinta pantallas le resultado fácil, un diez por ciento (10%) “Algo de acuerdo “y un cuarenta por ciento (40%) estuvo “Muy de acuerdo”. Los resultados se reflejan en el grafico 60. (Ver Figura 60)



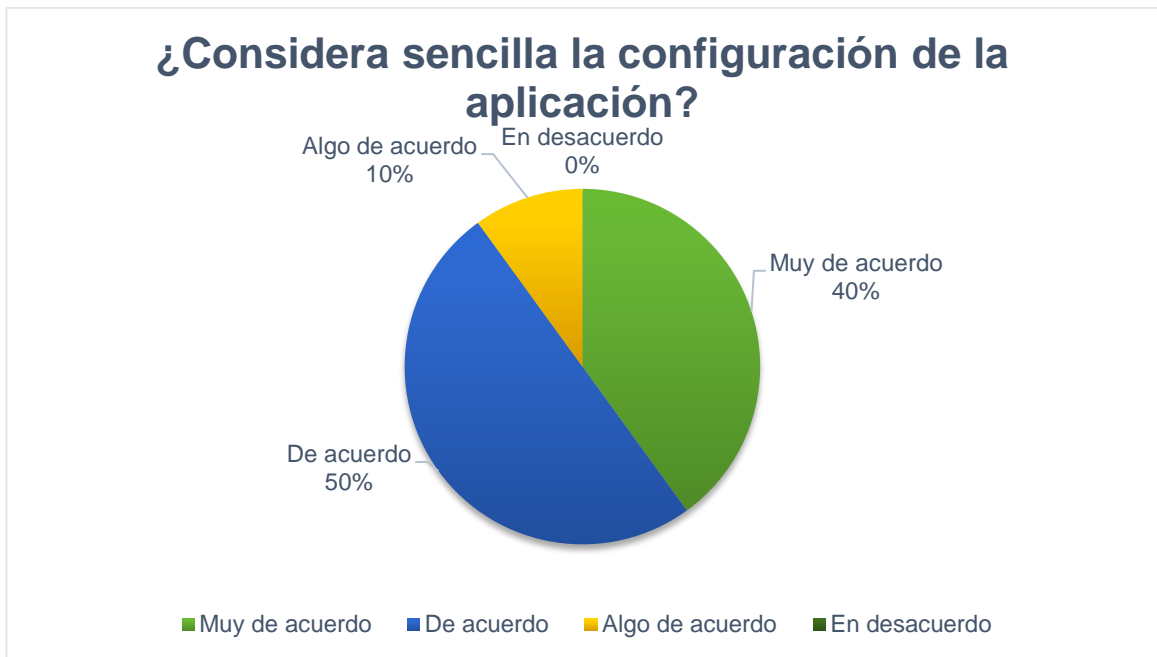
**Figura 60: Encuesta– Pregunta 4.**

Al analizar los resultados de la quinta pregunta se observa, un cuarenta por ciento (40%) estuvo “De acuerdo” con la rapidez del proceso de sincronización, un cuarenta por ciento (40%) igual estuvo “Algo de acuerdo“ y un veinte por ciento (20%) estuvo “Muy de acuerdo”. (Ver Figura 61)



**Figura 61: Encuesta– Pregunta 5.**

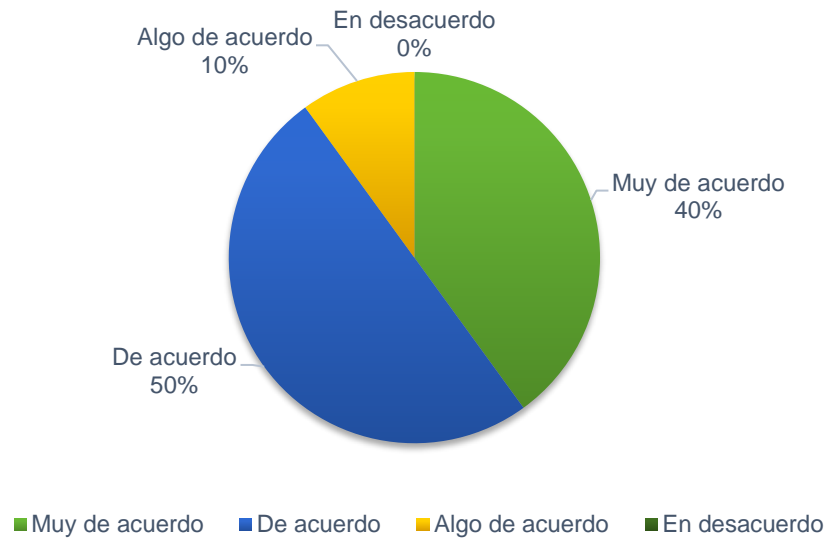
Al analizar los resultados de la sexta pregunta se observa, un cincuenta por ciento (50%) estuvo “De acuerdo” con la configuración de la aplicación, un cuarenta por ciento (40%) igual estuvo “Muy de acuerdo” y un diez por ciento (10%) estuvo “Algo de acuerdo”. (Ver Figura 62)



**Figura 62: Encuesta– Pregunta 6.**

Al analizar los resultados de la séptima pregunta se observa, un cincuenta por ciento (50%) estuvo “De acuerdo” con que la aplicación les parece atractivas visualmente, un cuarenta por ciento (40%) igual estuvo “Muy de acuerdo” y un diez por ciento (10%) estuvo “Algo de acuerdo”. (Ver Figura 63)

## ¿En términos generales, le resulta atractivo el sistema?



**Figura 63: Encuesta– Pregunta 7.**

## CONCLUSIONES Y RECOMENDACIONES

Hoy en día, las órdenes de servicio constituyen una parte vital del negocio en muchas empresas. En consecuencia, una eficaz y eficiente gestión de dichas órdenes puede tener gran impacto en la productividad y éxito de la organización, brindándole ventaja y haciendo que destaque por encima del resto en el mercado.

Para cumplir con el proceso de gestión de órdenes de servicio se necesita tomar las solicitudes de los clientes rápidamente, planificar las tareas a realizar, asignar el trabajo al personal correspondiente, realizar seguimiento al estatus del trabajo y facturar el servicio cuando se haya completado.

El presente proyecto de investigación tuvo como producto final una solución móvil multiplataforma, desarrollada utilizando estándares web, destinada a organizaciones que incluyan la gestión de órdenes de servicio como proceso de negocio. El desarrollo de la solución comprende una app para dispositivos móviles y una aplicación web de apoyo para la administración de la data. Así mismo, se desarrollaron una serie de servicios web que exponen la información de la base de datos a través de un API RESTful, desarrollado utilizando el framework Slim para aplicaciones programadas en lenguaje PHP. Aquí es importante mencionar también que los servicios web resultaron pieza clave en el desarrollo, pues son la parte central del proceso de sincronización de datos entre la aplicación móvil y la aplicación web.

La aplicación móvil está destinada a operar sobre las principales plataformas existentes en el mercado, lo cual puede tener un gran impacto en la masificación y alcance de una mayor cuota de usuarios finales.

La adopción del marco de trabajo para metodología ágil Scrum resultó de especial importancia en la consecución de los objetivos planteados. La planificación del proyecto a través de iteraciones que generan un incremento funcional al final de cada Sprint, permitió llevar control del avance en el desarrollo de las funcionalidades hasta la entrega del producto final. Así mismo, cabe destacar que las pruebas realizadas al final de cada iteración jugaron un papel preponderante en la calidad del producto final obtenido. Éstas, a través de las distintas reuniones del proceso Scrum, permitieron analizar y evaluar continuamente el trabajo realizado hasta el momento, haciendo posible detectar y corregir errores y realizar mejoras a tiempo antes de la entrega final.

El empleo de tecnologías web como Javascript, HTML y CSS, permitió agilizar el proceso de desarrollo de la solución. Su conocimiento y uso es bastante extendido, lo cual resultó una ventaja bastante importante para el equipo de desarrollo, que pudo optimizar el tiempo para realizar las entregas parciales con la regularidad requerida.

Durante el desarrollo se determinó que este proyecto aplica principalmente a la gestión de órdenes de servicio técnico.

Finalmente, se determinó que es posible a través de una solución de software móvil multiplataforma hacer uso de las tecnologías móviles y el internet para optimizar los procesos de gestión de órdenes de servicio en una organización. Llevar a cabo estos procesos haciendo uso de dispositivos móviles que operan sobre distintas plataformas, suponía un problema para muchas organizaciones, sobre todo en lo concerniente a costo de recursos y tiempo. Pero una aplicación móvil que pueda desarrollarse una vez y ejecutarse en plataformas diferentes se puede traducir en una solución efectiva que genere mayor beneficio de rentabilidad para la empresa.

## Recomendaciones

Las pruebas de usabilidad realizadas al final del ciclo de desarrollo del proyecto resultaron ser un instrumento muy importante para medir el nivel de satisfacción del usuario final en cuanto a la calidad del producto de software. Así pues, el resultado de estas pruebas contribuyó mucho a la experiencia adquirida durante el desarrollo del proyecto de investigación. Se pudo observar que existen posibles mejoras a realizar, pero las mismas no se llevaron a cabo por estar fuera del alcance de este trabajo.

Para trabajos futuros se realizan las siguientes recomendaciones:

- Establecer prioridades en las órdenes de servicio. De esta manera se puede tener un mayor control en la asignación de técnicos de servicio, estimación de fecha y hora de ejecución, etc.
- Implementar geolocalización haciendo uso del GPS del dispositivo móvil. Esto podría mejorar el proceso de negocio optimizando los tiempos de entrega del servicio a través de, por ejemplo, cálculo de rutas, ubicación de personal disponible más cercano, entre otros.
- Permitir que el cliente tenga acceso vía web o app móvil, a las órdenes de servicio que ha contratado.
- Desarrollar un generador de reportes para llevar un control histórico y estadístico de las órdenes de servicio. Dicho módulo incluso podría implementarse de forma que otorgue también acceso a los clientes y sus órdenes correspondientes.
- Enviar notificaciones vía correo electrónico o mensajería de texto con la actualización del estado de las órdenes de servicio. Igualmente, se puede utilizar a manera de recordatorio cuando esté próximo el día de ejecución de la orden.

- Utilizar notificaciones *push* para informar de manera automática a través de la app móvil acerca de cambios en el estatus o creación de nuevas órdenes de servicio.
- Utilizar un calendario *drag-and-drop* para hacer más amigable y eficiente la creación de órdenes de servicio, asignación de técnicos, fechas y horas estimadas.
- Con respecto al módulo de reporte de servicio, en vez de una descripción general se pueden definir las tareas a realizar de forma independiente (por ejemplo, a través de una lista de tareas, cada una con un *checkbox* para indicar si fue completada). De esta manera, se podría llevar un mejor control acerca del trabajo realizado durante la ejecución de la orden de servicio.
- La carga y manipulación de datos de clientes, usuarios, técnicos y partes en el sistema se realiza vía administración de base datos. Por esto, se recomienda la creación de un módulo que permita la gestión de esta tarea desde la aplicación.
- A pesar de lo reducido del grupo de trabajo, se logró implementar Scrum como marco de trabajo de metodología ágil de manera efectiva. Se recomienda ampliamente no centrarse de forma rígida en el proceso que plantea, sino más bien adaptarla a lo que se tiene de la mejor manera, tomando en cuenta las mejores prácticas y experiencias previas de otros en proyectos similares.
- Se recomienda implementar la solución desarrollada en un caso de estudio real con el fin de realizar una evaluación completa para mejorar la calidad del software.

## REFERENCIAS BIBLIOGRAFICAS

- Adolph, M. (julio de 2009). Mobile Applications. *ITU-T TechWatch Alert*, 1. Recuperado el 25 de septiembre de 2014, de [http://www.itu.int/dms\\_pub/itu-t/oth/23/01/T230100000C0004PDFE.pdf](http://www.itu.int/dms_pub/itu-t/oth/23/01/T230100000C0004PDFE.pdf)
- Android OS: Arquitectura*. (2012). Recuperado el 17 de octubre de 2014, de <http://androidos.readthedocs.org/en/latest/data/caracteristicas/>
- Android vs iOS*. (26 de mayo de 2013). Recuperado el 17 de octubre de 2014, de <http://jcampohucn.blogspot.com/2013/05/arquitectura.html>
- Aranaz, J. (2009). *Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma android de google*. Proyecto fin de carrera, Universidad Carlos III de Madrid, Ingeniería en informática. Recuperado el 8 de julio de 2013, de [http://e-archivo.uc3m.es/bitstream/10016/6506/1/PFC\\_Jaime\\_Aranaz\\_Tudela\\_2010116132629.pdf](http://e-archivo.uc3m.es/bitstream/10016/6506/1/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf)
- Asociación javaHispano. (9 de febrero de 2012). *Diseñando aplicaciones multiplataforma en HTML con PhoneGap*. Recuperado el 17 de octubre de 2014, de <http://www.javahispano.org/android/2012/2/9/diseando-aplicaciones-multi-plataforma-en-html-con-phonegap.html>
- Avilé, M. J., & Gerardino, D. G. (2012). *El Teletrabajo como alternativa de desempeño profesional para el ejercicio de los Contadores Públicos Independientes en el estado Sucre*. Trabajo de Curso Especial de Grado, Universidad de Oriente, Escuela de Administración, Cumaná. Recuperado el 1 de julio de 2014

- Baz, A., Ferreira, I., Álvarez, M., & García, R. (2013). *Dispositivos Móviles*. Universidad de Oviedo, Ingeniería de Telecomunicación. Recuperado el 8 de julio de 2013, de <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>
- Bibhudatta. (25 de febrero de 2013). *What Is Phonegap Cordova*. Recuperado el 12 de julio de 2013, de Javatechig. Resources for developers: <http://javatechig.com/cross-platform/phonegap/what-is-phonegap>
- Boiarov, S. A., & Poliszuk, N. (2013). *Centro de Teletrabajo y Teleformación de la Universidad de Buenos Aires*. Recuperado el 1 de julio de 2014, de [http://www.caminandoutopias.org.ar/institucional/que\\_es\\_el\\_teletrabajo.php](http://www.caminandoutopias.org.ar/institucional/que_es_el_teletrabajo.php)
- Celularis*. (18 de febrero de 2013). Recuperado el 16 de octubre de 2014, de <http://www.celularis.com/mercado/sistemas-operativos-moviles-2013/>
- Conjecture Corporation. (2014). *What Is a Work Order?* Recuperado el 25 de septiembre de 2014, de wiseGEEK: <http://www.wisegeek.com/what-is-a-work-order.htm>
- Corrigo Incorporated. (2013). Work Order Lifecycle. *Work order network*, 1-2. Retrieved septiembre 25, 2014, from <http://media.corrigo.com/workordernetwork/WON%20Documents/Collateral/Work%20Orders/Work%20Order%20LifeCycle%20-%20Overview.pdf>
- Cruz, C. (2013). *Dispositivos Móviles*. Recuperado el 17 de octubre de 2014, de <http://arielrodrigoreyes.wordpress.com/que-es-un-dispositivo-movil/>

El método Scrum. (2007). 54-58. Recuperado el 16 de octubre de 2014, de [http://www.mastersoft.com.ar/MsWeb/otros\\_archivos/NotaScrumPCUsers.pdf](http://www.mastersoft.com.ar/MsWeb/otros_archivos/NotaScrumPCUsers.pdf)

Eternal Software. (2014). *Desarrollos Mobile en WEB o aplicaciones nativas*. Recuperado el 17 de octubre de 2014, de [http://www.eternalsoftware.com.ar/noticia\\_mobileweb\\_vs\\_mobileapp.php](http://www.eternalsoftware.com.ar/noticia_mobileweb_vs_mobileapp.php)

Find Accounting Software. (2014). *Work Order Management Software: Solutions for Performance Optimization*. Recuperado el 30 de septiembre de 2014, de <http://findaccountingsoftware.com/focus-on-functionality/work-order-management-software-solutions-for-performance-optimization/>

Garzo, M. (28 de diciembre de 2012). *Desarrollo de aplicaciones móviles híbridas con ADF Mobile*. Recuperado el 17 de octubre de 2014, de <http://blog.avanttic.com/2012/12/28/desarrollo-de-aplicaciones-moviles-hibridas-con-adf-mobile/> (100% Oracle):

Geospatial Training. (2013). *Tipos de aplicaciones móviles*. Recuperado el 30 de septiembre de 2014, de <http://geospatialtraininges.com/recursos-gratuitos/tipos-de-aplicaciones-moviles/>

Grace, N. (2012). *Work Order vs. Invoice*. Recuperado el 25 de septiembre de 2014, de Small Business: <http://smallbusiness.chron.com/work-order-vs-invoice-48573.html>

Lamb, C., Hair, J., & McDaniel, C. (2011). *Marketing* (11va ed.). Cengage Learning.

Naranjo. (2011). Oracle Corporation.

- Naranjo, F. (2011). El teletrabajo, una alternativa para mejorar la calidad de vida. *Éxito Empresarial*(139), 1-3. Recuperado el 1 de Julio de 2013, de [http://www.cegesti.org/exitoempresarial/publicaciones/publicacion\\_139\\_280211\\_es.pdf](http://www.cegesti.org/exitoempresarial/publicaciones/publicacion_139_280211_es.pdf)
- Palacio, J. (2006). *El modelo Scrum*. Navegapolis. Recuperado el 22 de julio de 2013, de [http://www.navegapolis.net/files/s/NST-010\\_01.pdf](http://www.navegapolis.net/files/s/NST-010_01.pdf)
- Pedrozo, G. (2012). *Sistemas Operativos en Dispositivos Móviles*. Monografía, Universidad Nacional del Nordeste, Licenciatura en Sistemas de Información. Recuperado el 8 de julio de 2013, de [http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Sistemas\\_Operativos\\_en\\_Dispositivos\\_Moviles.pdf](http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Sistemas_Operativos_en_Dispositivos_Moviles.pdf)
- Sanchez, J. M. (2015). *Pruebas de Software. Fundamentos y Técnicas*. Madrid: Universidad Politécnica de Madrid.
- Stanton, W., Etzel, M., & Walker, B. (2004). *Fundamentos de Marketing* (13va edición ed.). Mc Graw Hill.
- Sutherland, J., & Schwaber, K. (2011). *La Guía Definitiva de Scrum: Las Reglas del Juego*. Scrum.org. Recuperado el 22 de julio de 2013, de [http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum\\_Guide%202011%20-%20ES.pdf](http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide%202011%20-%20ES.pdf)
- Vásquez, S. (25 de enero de 2011). *Tecnología e informática*. Recuperado el 8 de julio de 2013, de <http://solvasquez.wordpress.com/2011/01/24/definicion-de-sistema-operativo/>
- Vertex42 LLC. (2010). *Free Work Order Form Template*. Recuperado el 25 de septiembre de 2014, de vertex42: <http://www.vertex42.com/ExcelTemplates/work-order-form.html>

Wargo, J. (2012). *PhoneGap essentials: building cross-platform apps*.  
Crawfordsville: Addison-Wesley.