

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

**SISTEMA INTEGRAL PARA EL REPORTE Y SEGUIMIENTO DE DEFICIENCIAS
DEL SERVICIO PÚBLICO DEL MUNICIPIO DE OTHÓN P. BLANCO, QUINTANA
ROO (072 REPORTES)**

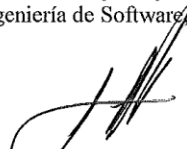
POR:

ANGEL GUILLERMO TREJO ROJAS

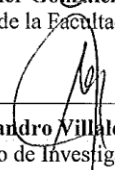
**TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA DE SOFTWARE**



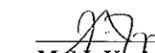
Sistema integral para el reporte y seguimiento de deficiencias del servicio público del municipio de Othón P. Blanco, Quintana Roo (072 Reportes). Tesis presentada por Ángel Guillermo Trejo Rojas como requisito parcial para obtener el grado de Maestro en Ingeniería de Software, ha sido aprobado y aceptado por:



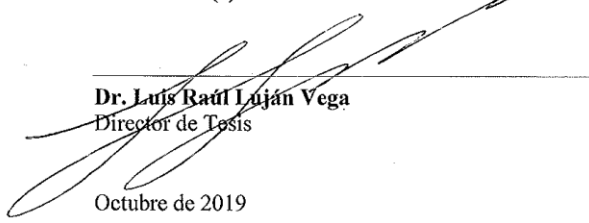
M.I. Javier González Cantú
Director de la Facultad de Ingeniería



Dr. Alejandro Villalobos Aragón
Secretario de Investigación y Posgrado



M.S.I. Karina Rocío Requena Yáñez
Coordinador(a) Académico



Dr. Luis Raúl Luján Vega
Director de Tesis

Octubre de 2019

Comité:

Dr. Luis Raúl Luján Vega
Dr. Rafael Camilo Lozoya Gámez
M.S.I. Karina Rocío Requena Yáñez
M.I. David Maloof Flores

© Derechos Reservados

Ángel Guillermo Trejo Rojas
Circuito No. 1, Campus
Universitario 2, Chihuahua, Chih.
México C.P. 31125

Octubre 2019



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

7 de octubre de 2019

L.I. ÁNGEL GUILLERMO TREJO ROJAS
Presente

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ingeniería de Software, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Luis Raúl Luján Vega** para que lo desarrolle como tesis, con el título: **SISTEMA INTEGRAL PARA EL REPORTE Y SEGUIMIENTO DE DEFICIENCIAS DEL SERVICIO PÚBLICO DEL MUNICIPIO DE OTHÓN P. BLANCO, QUINTANA ROO (072 REPORTES).**

Índice

Dedicatoria

Agradecimientos

Resumen

Índice de Contenido

Índice de tablas

Índice de figuras

Capítulo 1. Introducción

Capítulo 2. Antecedentes

- 2.1 Los servicios públicos
- 2.2 El gobierno y las tic's
- 2.3 El sistema anterior 072 ayuntatel

Capítulo 3. Justificación

- 3.1 Delimitación del problema

Capítulo 4: Fundamentación teórica

- 4.1 Conceptos generales
- 4.2 Conceptos técnicos

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

Capítulo 5. Objetivos

5.1 Objetivo general

5.2 Objetivos específicos

Capítulo 6. Hipótesis

Capítulo 7. Metodología

Capítulo 8. Resultados

8.1 Resultados

Capítulo 9. Conclusiones

Capítulo 10. Trabajo futuro

Referencias

Apéndice 1

Apéndice 2

Curriculum Vitae

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

ATENTAMENTE
"Naturam subiecit aliis"

EL DIRECTOR

FACULTAD DE
INGENIERÍA
UACH.

EL SECRETARIO DE INVESTIGACIÓN
Y POSGRADO

M.I. JAVIER GONZÁLEZ CANTÚ



DIRECCIÓN

DR. ALEJANDRO VILLALOBOS ARAGÓN

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

Dedicatoria

Este trabajo está dedicado a mis padres Ade y Miguel por la confianza que han depositado en mí y por su invaluable apoyo incondicional, ellos son mi ejemplo de vida, trabajo, esfuerzo y voluntad.

Agradezco especialmente a cada uno de los maestros sinodales que se han tomado el tiempo y dedicación en la revisión de este trabajo, sus observaciones y apoyo han sido de gran valor para mí.

Agradecimientos

A Dios, por su bondad, sus bendiciones y pruebas.

Resumen

El municipio de Othón P. Blanco se ubica en el estado de Quintana Roo, México, cuya cabecera es la Ciudad de Chetumal también capital del estado.

Como parte de su labor de atención a las demandas ciudadanas respecto a los servicios públicos, el municipio de Othón P. Blanco implementó en el año 2007 el sistema 072 Ayuntatel para la recepción de reportes ciudadanos. Este sistema automatiza el proceso de registro y seguimiento de dichos reportes. Con el transcurso del tiempo las modificaciones al sistema ya no fueron suficientes para satisfacer las necesidades de los usuarios, esto aunado con la evolución del software usado para su desarrollo han dado la pauta para la creación de un nuevo sistema informático para cubrir dicho proceso.

Por lo antes descrito el proyecto 072 Reportes se crea para sustituir al anterior sistema 072 Ayuntatel y de la misma manera ampliar su cobertura de servicio al involucrar por primera vez a los ciudadanos directamente mediante el desarrollo y uso de una aplicación para *Smartphone*. De esta manera, el Sistema Integral para el Reporte y Seguimiento de Deficiencias del Servicio Público del Municipio de Othón P. Blanco, Quintana Roo (072 Reportes) se compone de dos productos: un sistema web visible en una intranet y de uso exclusivo para el personal del ayuntamiento, y una aplicación móvil para sistemas operativos Android.

Índice de Contenido

Dedicatoria.....	1
Agradecimientos.....	2
Resumen	3
Índice de Contenido.....	4
Índice de tablas	6
Índice de figuras	7
Capítulo 1: Introducción.....	11
Capítulo 2: Antecedentes.....	12
2.1 LOS SERVICIOS PÚBLICOS	12
2.2 EL GOBIERNO Y LAS TIC'S	13
2.3 EL SISTEMA ANTERIOR 072 AYUNTATEL	15
Capítulo 3: Justificación.....	18
3.1 DELIMITACIÓN DEL PROBLEMA.....	19
Capítulo 4: Fundamentación teórica.....	21
4.1 CONCEPTOS GENERALES.....	21
4.2 CONCEPTOS TÉCNICOS	33
Capítulo 5: Objetivos.....	46
5.1 OBJETIVO GENERAL	46
5.2 OBJETIVOS ESPECÍFICOS	46
Capítulo 6: Hipótesis	47
Capítulo 7: Metodología.....	48
Capítulo 8: Resultados.....	50
8.1 RESULTADOS	50



Capítulo 9: Conclusiones.....	80
Capítulo 10: Trabajo futuro	82
Referencias	83
Apéndice 1	86
Apéndice 2	91
Curriculum Vitae	96

Índice de tablas

Tabla 1: Principales protocolos de internet. 30

Tabla 2: Requerimientos del proyecto 072 Reportes. 59

Índice de figuras

Figura 1: Funcionamiento del sistema actual	16
Figura 2: Funcionamiento del proyecto 072 Reportes	19
Figura 3: Partes de un sistema	22
Figura 4: Clasificación del software	24
Figura 5: Metodología en cascada	26
Figura 6: Metodología incremental	27
Figura 7: Metodología orientada a objetos	28
Figura 8: La internet	29
Figura 9: Medios físicos de conexión	31
Figura 10: El internet de las cosas	32
Figura 11: El funcionamiento de la metodología modelo – vista - controlador	35
Figura 12: Principales plataformas de las aplicaciones móviles	39
Figura 13: Modelo Cliente - Servidor	41
Figura 14: Modelo Maestro - Esclavo	42
Figura 15: Modelo Productor - Consumidor	43
Figura 16: Modelo Publicador - Suscriptor	45
Figura 17: Metodología del desarrollo de tesis	49
Figura 18: Funcionamiento del sistema actual	50
Figura 19: Diagrama 0 del nuevo sistema 072 Reportes	51
Figura 20: Diagrama del proceso 1 del nuevo sistema 072 Reportes	52
Figura 21: Diagrama del proceso 2 del nuevo sistema 072 Reportes	53
Figura 22: Diagrama del proceso 3 del nuevo sistema 072 Reportes	54
Figura 23: Diagrama del proceso 4 del nuevo sistema 072 Reportes	55
Figura 24: Diagrama del proceso 5 del nuevo sistema 072 Reportes	56

Figura 25: Diagrama ER de la base de datos en el dispositivo móvil	57
Figura 26: Diagrama ER de la base de datos del sistema web	58
Figura 27: Diagrama de navegación del sistema web	62
Figura 28: Ventana de <i>logueo</i>	63
Figura 29: Formulario de captura para un nuevo reporte	64
Figura 30: Detalle del reporte	65
Figura 31: Designación del área responsable de atención	65
Figura 32: Pantalla de seguimiento	66
Figura 33: Pantalla de finalización de reporte	66
Figura 34: Mapa de reportes registrados	67
Figura 35: Gráfica “por estado del reporte”	68
Figura 36: Gráfica “por unidad administrativa”	68
Figura 37: Gráfica “por año de registro”	68
Figura 38: Pantalla para crear un nuevo aviso	69
Figura 39: Edición de un aviso	69
Figura 40: Envío de aviso	70
Figura 41: Diagrama de navegación de la <i>app</i>	71
Figura 42: Pantalla principal de la aplicación	72
Figura 43: Formulario de captura	72
Figura 44: Mapa de reportes realizados	73
Figura 45: Detalle del reporte	73
Figura 46: Encuesta de calidad del servicio	74
Figura 47: Detalle del aviso	74
Figura 48: Evaluación del sistema web	75
Figura 49: Evaluación del sistema web	76

Figura 50: Evaluación de la aplicación móvil	77
Figura 51: Evaluación de la aplicación móvil	77
Figura 52: Presentación del proyecto a directores de área.....	78
Figura 53: Presentación del proyecto a directores de área.....	78
Figura 54: El sistema web en producción	79
Figura 55: La aplicación móvil publicada	79
Figura 56: Código del modelo de la tabla reportes	86
Figura 57: Función para guardar un nuevo reporte	87
Figura 58: Función para asignar un área responsable de atención.....	87
Figura 59: Función para eliminar un reporte	87
Figura 60: Parte de la función para guardar un seguimiento	88
Figura 61: Parte de la función para enviar una notificación de finalización de reporte	88
Figura 62: Parte de la función para crear un nuevo aviso	89
Figura 63: Función para eliminar un aviso	89
Figura 64: Función para registrar los resultados de la encuesta de calidad del servicio	89
Figura 65: Parte del código de la vista para mostrar el mapa de reportes realizados	90
Figura 66: Parte del código de la vista para enviar un aviso a los dispositivos móviles	90
Figura 67: Parte del código para enviar los datos del reporte al servidor	91
Figura 68: Parte del código para generar el <i>layout</i> para la captura del reporte	92
Figura 69: Parte del código para recuperar las coordenadas del ciudadano..	92
Figura 70: Parte del código para recuperar los reportes y mostrarlos en un mapa	93
Figura 71: Código para mostrar el <i>layout</i> del mapa de reportes	93
Figura 72: Parte del código para mostrar el detalle del reporte.....	94

Figura 73: Parte del código para mostrar el <i>layout</i> del detalle del reporte	94
Figura 74: Definición de la estructura de la base de datos en el dispositivo móvil	95
Figura 75: Parte del archivo AndroidManifest.xml	95

Capítulo 1: Introducción

El presente trabajo presenta la tesis, esta consta del desarrollo del proyecto “072 Reportes”, cuyo objetivo es optimizar el proceso de reporte, seguimiento y resolución de deficiencias en los servicios públicos que proporciona el municipio de Othón P. Blanco, Quintana Roo. Para esto se ha realizado la construcción de un sistema web para uso exclusivo del personal del ayuntamiento en donde se recibirán, se dará seguimiento y solución a los reportes, y una aplicación nativa para dispositivos móviles que los ciudadanos utilizarán para el levantamiento de reportes.

Se ha revisado la literatura para conocer el estado actual de proyectos de este tipo en México, así como la funcionalidad y aporte de las TIC's (Tecnologías de Información y Comunicación) en el funcionamiento de los procesos gubernamentales (e-Gobierno) y con base en este análisis construir una solución informática moderna capaz de registrar, monitorear y dar seguimiento al proceso de atención a la ciudadanía.

Se explican también los conceptos tecnológicos claves para una comprensión plena del proyecto aquí planteado, mismos que van desde conceptos generales hasta conceptos específicos relacionados con el desarrollo de software. De igual manera, se describen los pasos en orden cronológico a seguir para el desarrollo del proyecto, y con base en estos, se plantea un plan de trabajo acotado a cinco meses.

Para finalizar se explica el aporte y beneficios que este proyecto traerá a la sociedad y gobierno, los resultados obtenidos después de su evaluación y pruebas y el trabajo a futuro planteado para este proyecto.

Capítulo 2: Antecedentes

Este trabajo presenta el proyecto Sistema Integral para el Reporte y Seguimiento de Deficiencias del Servicio Público del Municipio de Othón P. Blanco, Quintana Roo (072 Reportes), en adelante “072 Reportes” el cual es un sistema integral, modular y escalable para reportar y realizar el seguimiento en la resolución de deficiencias en los servicios públicos en el municipio de Othón P. Blanco, estado de Quintana Roo. Este sistema busca volver más eficiente y ágil el proceso de reporte de deficiencias en los servicios públicos municipales (fugas de agua, baches, lotes baldíos en mal estado, alumbrado, condiciones insalubres en cementerios o rastros, etc.), de la misma manera, busca transparentar para beneficio del ciudadano, la atención y seguimiento de dichos reportes hasta su resolución, esto, mediante la consulta del estatus de los reportes y las notificaciones de las acciones de seguimiento.

2.1 LOS SERVICIOS PÚBLICOS

Podemos definir el concepto de servicios públicos como “la actividad destinada a satisfacer una necesidad colectiva de carácter material, económico o cultural, mediante prestaciones concretas por parte del Estado, de particulares o ambos, sujetos a un régimen jurídico que les imponga adecuación, regularidad y uniformidad, con fines sociales” [1]. En el contexto de nuestra sociedad, se han determinado principalmente los siguientes tipos de servicios públicos [2]:

- Agua potable. Instalación y mantenimiento de las tuberías de agua potable, así como su extracción y potabilización.
- Alumbrado público. Mantenimiento de las luminarias públicas.
- Alcantarillado. Mantenimiento de las redes subterráneas de desagüe por las cuales son conducidas las aguas del drenaje sanitario y pluvial hacia plantas de tratamiento especializadas.

- Mantenimiento de calles. Construcción, mantenimiento y señalización de las vías públicas.
- Limpia. Recolección, almacenamiento y tratamiento de la basura y desechos que son generados por la ciudadanía con el fin de mantener una ciudad limpia.
- Mercados. Administración y mantenimiento de los mercados y centros de distribución de alimentos.
- Cementerios. Administración y mantenimiento de los espacios destinados para el entierro de los restos mortales.
- Parques. Mantenimiento de jardines, fuentes, parques y lugares públicos de esparcimiento.
- Rastros. Vigilar que se cumplan las medidas de higiene requeridas en estos centros para el sacrificio y tratamiento de los animales.
- Seguridad pública. Vigilar la prevención del delito y aplicación de la ley.

2.2 EL GOBIERNO Y LAS TIC'S

Actualmente, es imposible concebir el funcionamiento de la administración pública sin el uso de las Tecnologías de la Información y Comunicación (TIC's). El aporte de las TIC's se centra en la automatización y sistematización de los procesos administrativos para una toma de decisiones eficiente. La rápida evolución de la informática les ha permitido una fácil integración a los procesos gubernamentales, dando como resultado el nacimiento del *e-government* (e-gobierno o gobierno electrónico), el cual, de acuerdo con la Organización de Estados Americanos se define como: la aplicación de las tecnologías de la información y la comunicación (TIC's) al funcionamiento del sector público, con el objetivo de incrementar la eficiencia, la transparencia y la participación ciudadana [3].

Como parte de esta transformación de las TIC's ha surgido el cómputo móvil y ubicuo, un ejemplo de su aplicación es la aparición de los *Smartphone* (teléfonos inteligentes) que le han permitido al gobierno un acercamiento como nunca hacia los

ciudadanos estableciendo un nuevo canal de comunicación más directo y transparente. Las aplicaciones (*app*) desarrolladas para estos dispositivos móviles han permitido el empoderamiento de los ciudadanos mediante el fomento de la participación, vigilancia y exigencia de cumplimiento de las obligaciones gubernamentales para con la ciudadanía.

La idea de utilizar una *app* que permita reportar desperfectos en los servicios públicos se ha implementado en proyectos alrededor de la república mexicana. En el municipio de Othón P. Blanco aún no se cuenta con aplicaciones de este tipo.

Una característica de este tipo de proyectos es que su alcance está delimitado por las fronteras geopolíticas de las ciudades o municipios que las implementan.

Entre los proyectos de este tipo que se han desarrollado en México tenemos la *app* “Bache 24” mediante la cual, en la Ciudad de México se reportan los baches, el gobierno recibe el reporte y una cuadrilla es asignada para atender y solucionar este desperfecto en la calle dentro de las siguientes 24 horas.

El proyecto “072 Reportes” está diseñado no solo para el reporte de baches, sino también para el reporte de cualquier otro inconveniente con cualquiera de los servicios públicos prestados por el Ayuntamiento de Othón P. Blanco (muchos de ellos listados anteriormente en esta sección). Por ejemplo: coladeras tapadas, parques con necesidad de mantenimiento, luminarias deficientes, entre otros.

Al momento de desarrollar este trabajo, se ha investigado sobre este tipo de proyectos en México encontrando proyectos como las *apps* “Servicios Públicos Tlajo App” en el municipio de Tlajomulco, Jalisco; “En contacto con Guadalupe” del municipio de Guadalupe, Zacatecas; “Marca el cambio” del municipio de Chihuahua, Chihuahua; “CiudadApp” del Ayuntamiento de Guadalajara, Jalisco y “Mérida móvil” del Ayuntamiento de Mérida, Yucatán.

Son pocos los proyectos de este tipo en relación con la gran cantidad de municipios en nuestro país en donde existen las condiciones adecuadas para una implantación de este tipo, tales como: cantidad de habitantes, infraestructura en telecomunicaciones disponible,

ciudadanos con acceso a internet y uso de *Smartphone*, entre otros. En la región sureste del país (Chiapas, Tabasco, Campeche, Yucatán y Quintana Roo) es el segundo proyecto después del implementado por el Ayuntamiento de Mérida “Mérida móvil”.

El proyecto “072 Reportes” además de servir para la notificación de fallas en servicios públicos también cuenta con un apartado para la recepción de noticias de interés y avisos referentes a servicios públicos que el ayuntamiento emita y considere que los ciudadanos necesiten saber, como por ejemplo la limpieza de coladeras en temporada de lluvias para evitar encharcamientos, la poda de árboles grandes en temporada de huracanes, cierres de avenidas por motivos de reparación, entre otros.

Este tipo de comunicados llegarán a los ciudadanos a través de notificaciones *push* a la aplicación, de la misma manera los ciudadanos podrán enviar sugerencias al municipio mediante la aplicación con el fin de mejorar la calidad del servicio.

2.3 EL SISTEMA ANTERIOR 072 AYUNTATEL

El proyecto “072 Reportes” nace de la necesidad de modernizar el antiguo sistema “072 Ayuntatel”. 072 Ayuntatel es un sistema que surge en el año 2007 como una herramienta que automatizaba el proceso de registro y seguimiento de los reportes del mal funcionamiento de los servicios públicos que ingresaban mediante llamadas al número telefónico 072 del municipio de Othón P. Blanco.

Este sistema fue desarrollado con el lenguaje de programación PHP 5.1.6 y usando el motor de base de datos MySQL 5.0.95 montados en el sistema operativo Linux. En su momento este sistema cubría las necesidades del personal del ayuntamiento para la realización de su trabajo, sin embargo, con el transcurso del tiempo las modificaciones al sistema ya no fueron suficientes para satisfacer dichas necesidades.

Las actualizaciones en el software de desarrollo (PHP y MySQL) y de sistema (Linux) que se liberaban eran incompatibles con el código fuente, debido a que no se contaba con los programadores iniciales, dio la pauta para desarrollar un nuevo sistema

adecuado a los requerimientos actuales de los usuarios, un sistema desarrollado con versiones modernas que aprovecha las bondades de las nuevas tecnologías informáticas, todo esto con el fin de brindar un mejor servicio a los usuarios así como a la ciudadanía.

Por lo antes descrito el proyecto 072 Reportes se crea para sustituir al anterior sistema 072 Ayuntatel y de la misma manera ampliar su cobertura de servicio al involucrar por primera vez a los ciudadanos directamente mediante el desarrollo y uso de una aplicación para *Smartphone*.

Este es el contexto bajo el cual se desarrolla el proyecto denominado “072 Reportes” que busca un acercamiento real entre gobierno y ciudadanía con el objetivo de volver más eficiente, rápido y transparente la atención y resolución de desperfectos en los servicios públicos municipales.

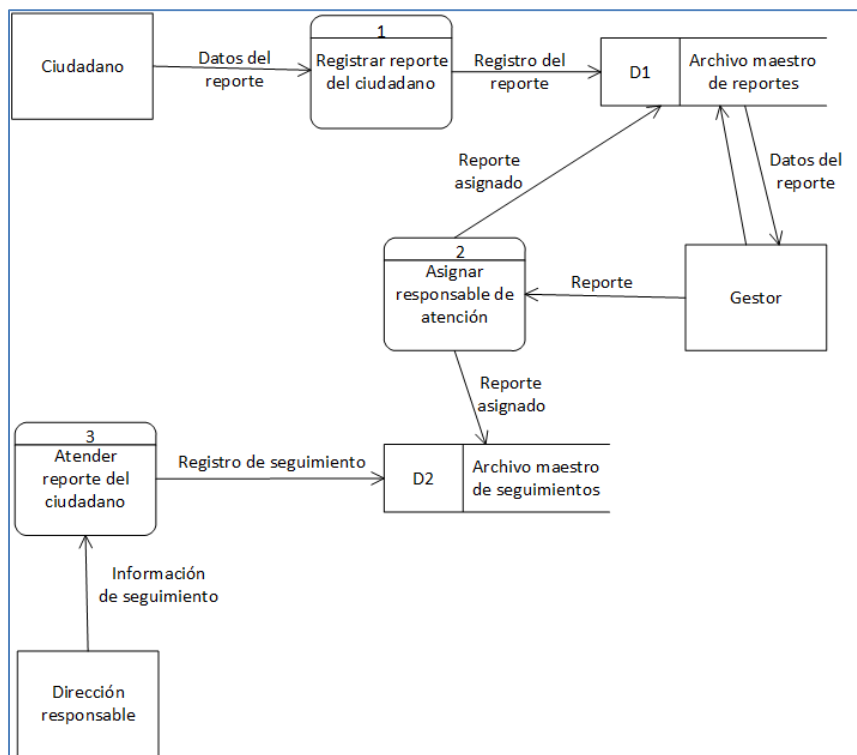


Figura 1: Funcionamiento del sistema actual [autoría propia].

En la figura anterior se describe el funcionamiento del sistema actual (072 Ayuntatel) en donde el ciudadano vía telefónica se comunica con una operadora y levanta su reporte, en ese momento el reporte es guardado y un usuario tipo gestor o administrador se encarga de asignar el reporte a un área para su seguimiento y resolución. La dirección encargada de resolverlo conforme va dándole seguimiento va registrando en el sistema estos avances (seguimientos) y así sucesivamente hasta darle una solución final, es entonces cuando el reporte se da por concluido.

Capítulo 3: Justificación

Se ha realizado el proyecto “072 Reportes” el cual busca la aplicación del cómputo móvil y ubicuo en el desarrollo de una solución informática para el reporte, seguimiento y atención de las deficiencias en los servicios públicos municipales. El proyecto consta de dos componentes: una aplicación móvil y un sistema web visible en una intranet para uso exclusivo del personal del ayuntamiento.

Actualmente el proceso de solución de reportes ciudadanos en el municipio de Othón P. Blanco presenta problemas de coordinación entre las áreas involucradas. Los problemas principales son la poca participación de la población en la realización de los reportes, los reportes falsos que implican el desperdicio de recursos como tiempo, combustible y personal, y por último la precisión de la ubicación geográfica de los reportes realizados.

Este proyecto nace de la necesidad de una mejora al canal de comunicación entre el gobierno municipal y la ciudadanía en cuanto al reporte de deficiencias en los servicios públicos. Actualmente el reporte de estos problemas se hace a través del número telefónico 072. Existe la necesidad de modernizar dicho canal de comunicación, y una forma de hacerlo es mediante la aplicación y el uso de las TIC's. Otra necesidad detectada es una automatización del proceso de seguimiento y solución de los reportes realizados por los ciudadanos.

La puesta en marcha de este proyecto ayudará al ayuntamiento a llevar un mejor control en la atención de reportes de servicio público. Los ciudadanos podrán levantar un reporte de manera más rápida y exacta (geográficamente hablando), adjuntando hasta tres imágenes como prueba del desperfecto a atender, propiciando así la participación ciudadana, y la generación de reportes y estadísticas de la información en tiempo real y automática.

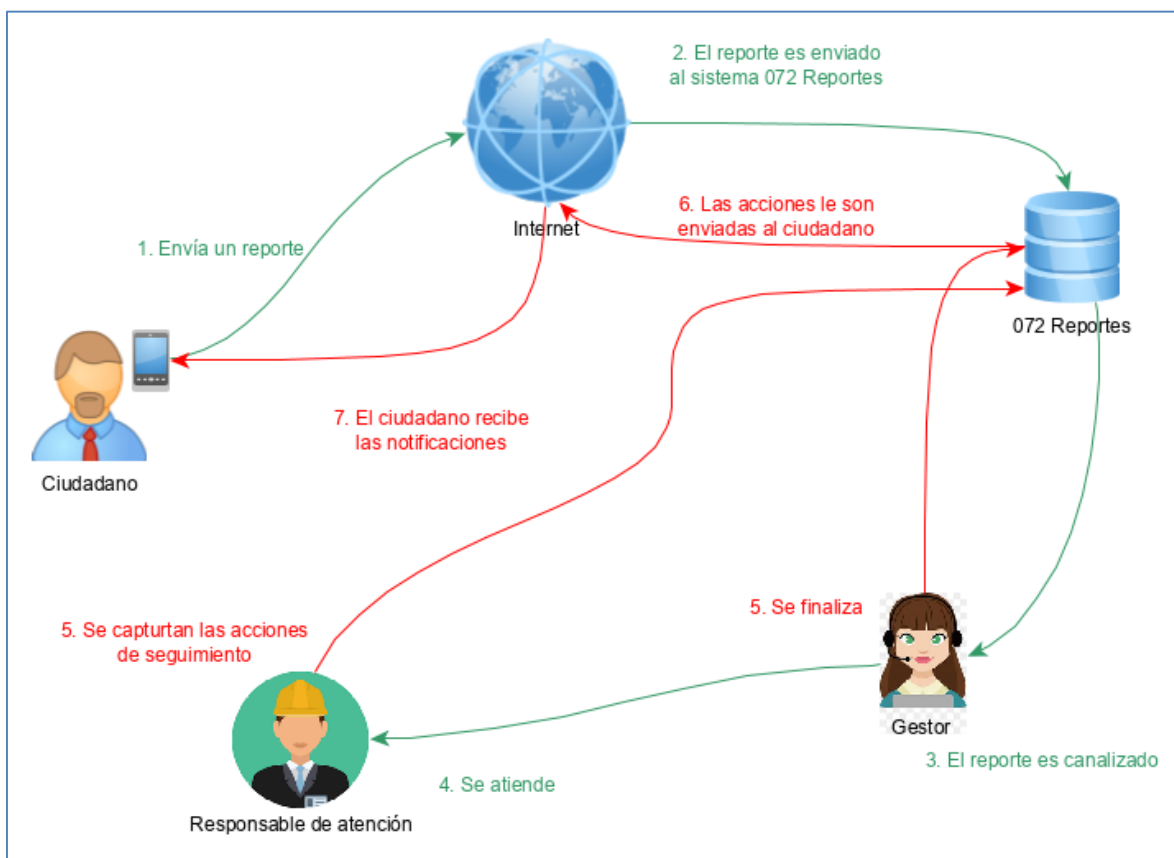


Figura 2: Funcionamiento del proyecto 072 Reportes [autoría propia].

3.1 DELIMITACIÓN DEL PROBLEMA

El presente proyecto se ha realizado en beneficio de la población del Municipio de Othón P. Blanco, Quintana Roo, así como en beneficio de las autoridades municipales.

Para garantizar el buen funcionamiento del proyecto, las condiciones que se deben cumplir en su implementación son:

- En el caso del sistema web, deberá correr dentro de una intranet, estar instalado en un servidor de preferencia Linux, con versiones de PHP 7.1, MySQL 5+ y Apache 2+. Al tratarse de una intranet no es necesario que las computadoras de los usuarios tengan acceso a internet, y con respecto al navegador de internet el sistema puede visualizarse sin ningún problema en cualquiera que el usuario elija ya que el sistema está desarrollado con estándares de diseño web. Es necesario que los usuarios cuenten con sus credenciales de acceso al sistema para poder usarlo.

- Para el caso de la aplicación móvil, los ciudadanos deberán contar con un *Smartphone* con sistema operativo Android con versiones entre 4.4 (Kit Kat) y 9 (Pie) ya que dicha *app* fue probada con dispositivos entre ese rango de versiones con éxito, sin embargo es probable que también funcione con versiones superiores a la 9. Adicionalmente, los dispositivos deberán contar con una cámara integrada y una antena gps debido a que este hardware es usado por la aplicación. Cabe mencionar que el usuario, al momento de abrir por primera vez la *app* deberá aceptar el uso de los siguientes permisos: almacenamiento interno, uso de cámara, acceso a la red y ubicación. Por último, la *app* también requiere que el usuario cuente con conexión a internet (datos o wifi) al momento de levantar su reporte.

Capítulo 4: Fundamentación teórica

El proyecto aquí expuesto describe el desarrollo de un software (administración del sistema vía web y una *app*) para reporte y seguimiento de deficiencias en los servicios públicos municipales. A continuación, se abordan los conceptos computacionales que intervienen en el desarrollo del proyecto desde la etapa del análisis hasta el mantenimiento, pasando por la definición de lo que es un sistema hasta conceptos técnicos como la arquitectura cliente - servidor.

4.1 CONCEPTOS GENERALES

4.1.1 Sistema informático

Para entender lo que un sistema representa desde el punto de vista informático, primero debemos remitirnos a la Teoría General de Sistemas (TGS), la cual surgió a partir de las siguientes ideas atribuidas a George Wilhem Friedrich Hegel (1770 – 1831):

- El todo es más que la suma de sus partes.
- El todo determina la naturaleza de las partes.
- Las partes no pueden comprenderse si se consideran en forma aislada del todo.
- Las partes están dinámicamente interrelacionadas o son interdependientes.

Estas ideas son tomadas por Ludwig Von Bertalanffy para que, en 1969 en su libro titulado “Teoría General de Sistemas” definiera un sistema como: “Un conjunto organizado de cosas o partes interactuantes e interdependientes, que se relacionan formando un todo unitario y complejo.” [4]. El sistema consta de tres partes:

- Entradas. Representa la materia prima con la cual funcionará el sistema, estas entradas pueden ser, materiales, recursos humanos o datos.

- **Proceso.** Es el mecanismo que transforma las entradas en salidas, previa aplicación de una acción. El proceso puede ser, una máquina, una tarea, un algoritmo computacional, etc.
- **Salidas.** Es el resultado que se obtienen del proceso de las entradas y que son de utilidad para los usuarios. Así, una salida puede ser un producto, un servicio o información.

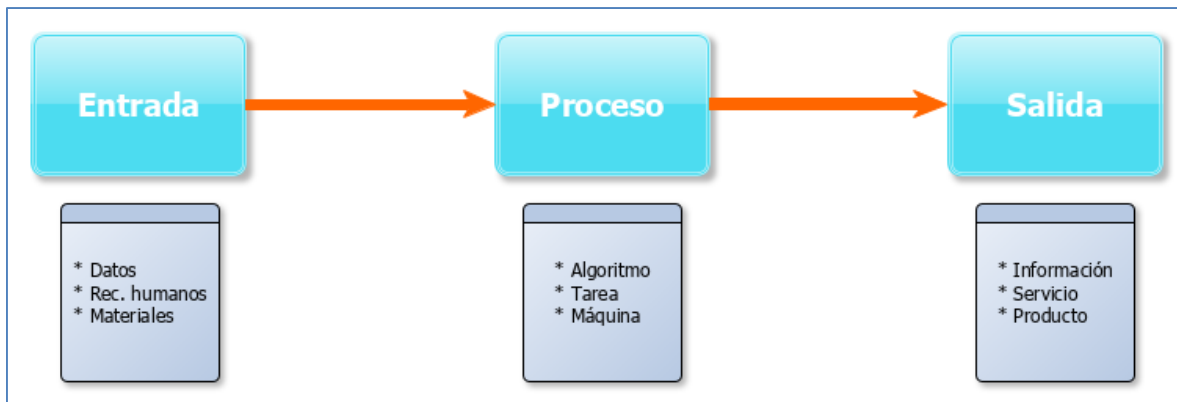


Figura 3: Partes de un sistema [autoría propia].

En el contexto informático, un sistema es un conjunto de piezas de software interrelacionadas con un objetivo en común, sus entradas son datos que por sí mismos no representan nada, a los cuales el sistema les aplica algoritmos (procesos) para posteriormente producir una salida la cual es representada por la información generada.

Los sistemas informáticos son usados para la esquematización, agilización y automatización de tareas y procesos dentro de las organizaciones, y con ello ser una herramienta confiable para la toma de decisiones.

Tradicionalmente un sistema informático estaba conformado por la aplicación (software) y una base de datos, ahora con el desarrollo del cómputo móvil, las aplicaciones móviles pueden considerarse parte de un sistema informático.

4.1.2 Software

La literatura de manera general define al software como “el equipamiento lógico o soporte lógico de un sistema informático; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas” [5]. Esto quiere decir que el software son todos aquellos programas de los cuales hacen uso los equipos de cómputo para cumplir con sus funciones.

El software se clasifica en:

- Software de sistemas. Son aquellas aplicaciones que permiten la administración de los recursos físicos de una computadora, como lo es el procesador, memoria, monitor, es decir del hardware sobre el cual se ejecutan, y al mismo tiempo sirven de plataforma para ejecutar otro tipo de software (aplicaciones). El software de sistemas se clasifica en sistemas operativos monousuario y multiusuarios. Ejemplo de este tipo de software son sistemas operativos, utilerías de diagnóstico, *drivers*, entre otros.
- Software de aplicación. Este tipo de software está construido para dar solución a tareas específicas que benefician al usuario final y que se ejecutan sobre el software de sistemas. Incluye aplicaciones de ofimática, antivirus, software educativo, etc.
- Software de programación. Son herramientas que permiten a los programadores desarrollar aplicaciones mediante el uso de lenguajes de programación. Con este tipo de software se crean los tipos de software de sistemas y de aplicación. Ejemplo de esta categoría son compiladores, intérpretes, lenguajes de programación, bibliotecas de funciones, entre otros.

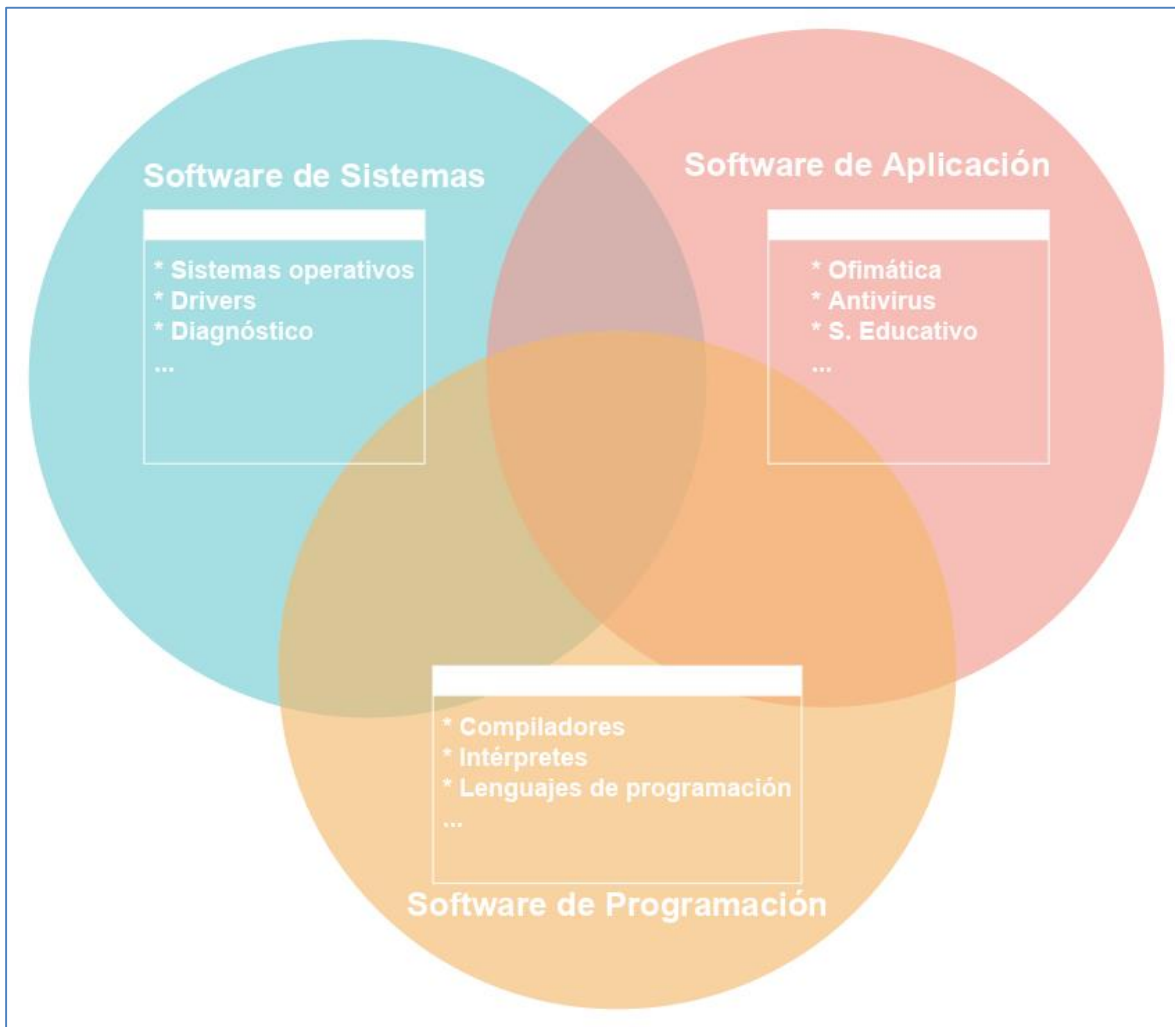


Figura 4: Clasificación del software [autoría propia].

4.1.3 Ingeniería de software

De acuerdo con la IEEE la ingeniería de software es “es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software”, de esta forma la ingeniería aporta sus principios y métodos para la creación de software.

Los objetivos de la ingeniería de software son [6]:

- Diseñar aplicaciones informáticas que se ajusten a las necesidades de las organizaciones.

- Dirigir y coordinar el desarrollo de aplicaciones complejas.
- Intervenir en todas las fases del ciclo de vida de un producto.
- Estimar los costes de un proyecto y determinar los tiempos de desarrollo.
- Hacer el seguimiento de costes y plazos.
- Dirigir equipos de trabajo de desarrollo software.
- Organizar la realización de pruebas que verifiquen el correcto funcionamiento de los programas y que se ajusten a los requisitos de análisis y diseño.
- Diseñar, construir y administrar bases de datos.
- Dirigir y asesorar a los programadores durante el desarrollo de aplicaciones.
- Introducir procedimientos de calidad en los sistemas, evaluando métricas e indicadores y controlando la calidad del software producido.
- Organizar y supervisar el trabajo de su equipo, de los técnicos de mantenimiento y los ingenieros de sistemas y redes.

Dentro de la ingeniería de software un aspecto importante son las metodologías de desarrollo, estas se definen como el conjunto de herramientas, técnicas y métodos a seguir durante el desarrollo de software. A continuación se citan tres de las más populares:

- Metodología en cascada. Tal vez la metodología de desarrollo de software más antigua, básicamente nos indica que debemos avanzar fase por fase, es decir, para poder avanzar a la siguiente fase se debe concluir con la actual, y una vez se avanza de fase no hay vuelta atrás. Las fases de esta metodología son:
 - Análisis de los requisitos del cliente.
 - Diseño del sistema.
 - Programación.
 - Integración.
 - Pruebas.
 - Documentación.

- Implementación.
- Mantenimiento.

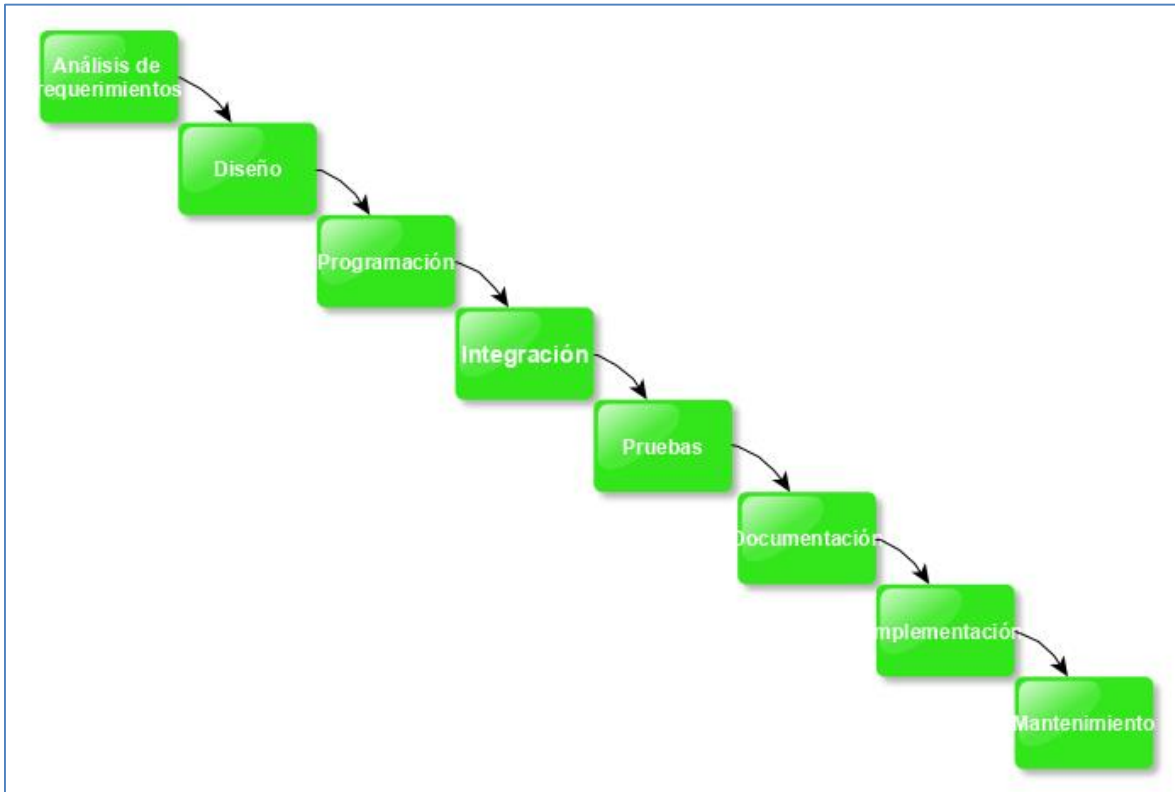


Figura 5: Metodología en cascada [autoría propia].

- Metodología incremental. Toma elementos del modelo en cascada y los repite a lo largo del tiempo de desarrollo. Al finalizar cada ciclo se obtiene un entregable. De esta forma el modelo incremental aplica secuencias lineales de forma escalonada. Cada secuencia lineal produce un incremento del software y cada incremento tiene su propio ciclo de vida. Este modelo busca en cada incremento agregar funcionalidad adicional al software o bien la corrección de errores, y también busca diseñar software que pueda ir entregándose por piezas y de esta forma los usuarios no tienen que esperar a que el software esté terminado en su totalidad para poder empezar a usarlo.

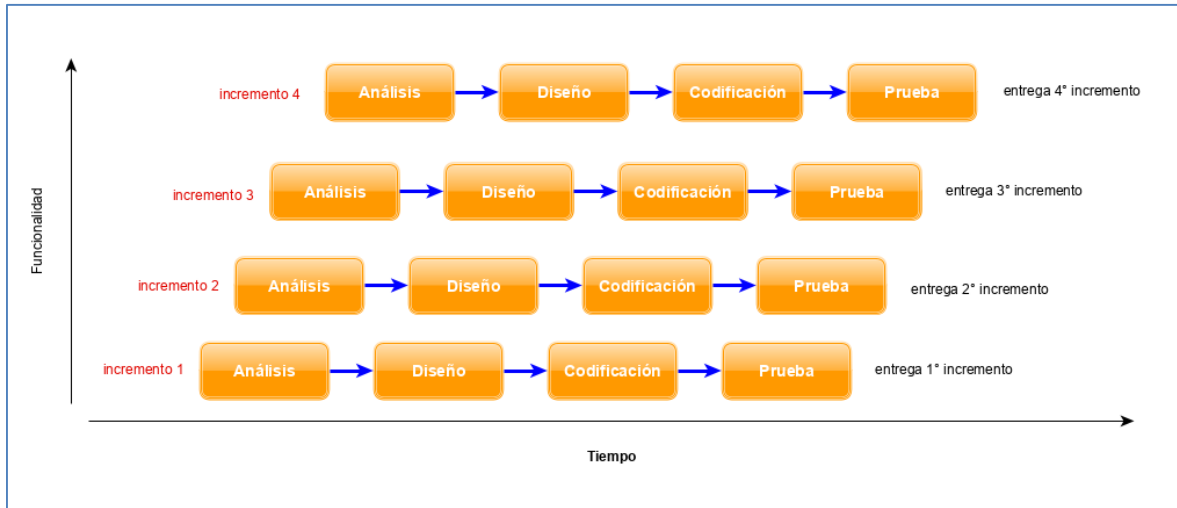


Figura 6: Metodología incremental [autoría propia].

- Metodología orientada a objetos. Fue propuesta por Grady Booch y este enfoque modela un sistema como un grupo de objetos que interactúan entre sí, cada objeto tiene una identidad y comportamiento propio, los cuales interactúan entre sí para lograr el objetivo del sistema. En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML).

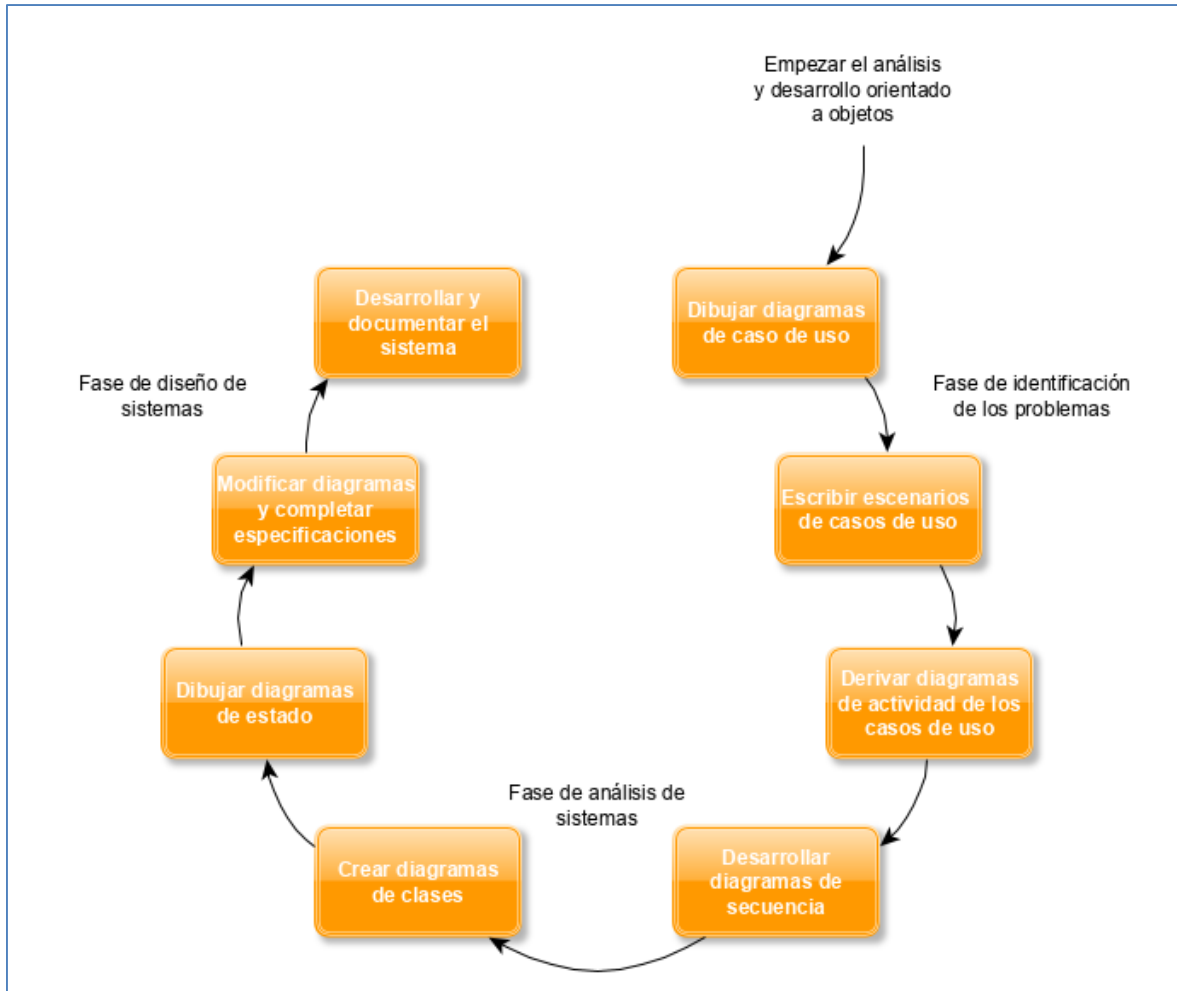


Figura 7: Metodología orientada a objetos [18].

4.1.4 Internet

También conocida como la red de redes, se inició en los 60s dentro del proyecto llamado DARPA con el objetivo de interconectar a las pocas computadoras de aquel entonces localizadas en algunas instalaciones universitarias y otras mayormente en instalaciones gubernamentales en los Estados Unidos, esto para buscar mejores maneras de usar el equipo de cómputo disponible en ese entonces.

De esta manera, la red internet es una interconexión de redes informáticas, y desde entonces ha crecido de manera exponencial hasta alcanzar más de 4,000 millones de usuarios conectados en 2019 [13].

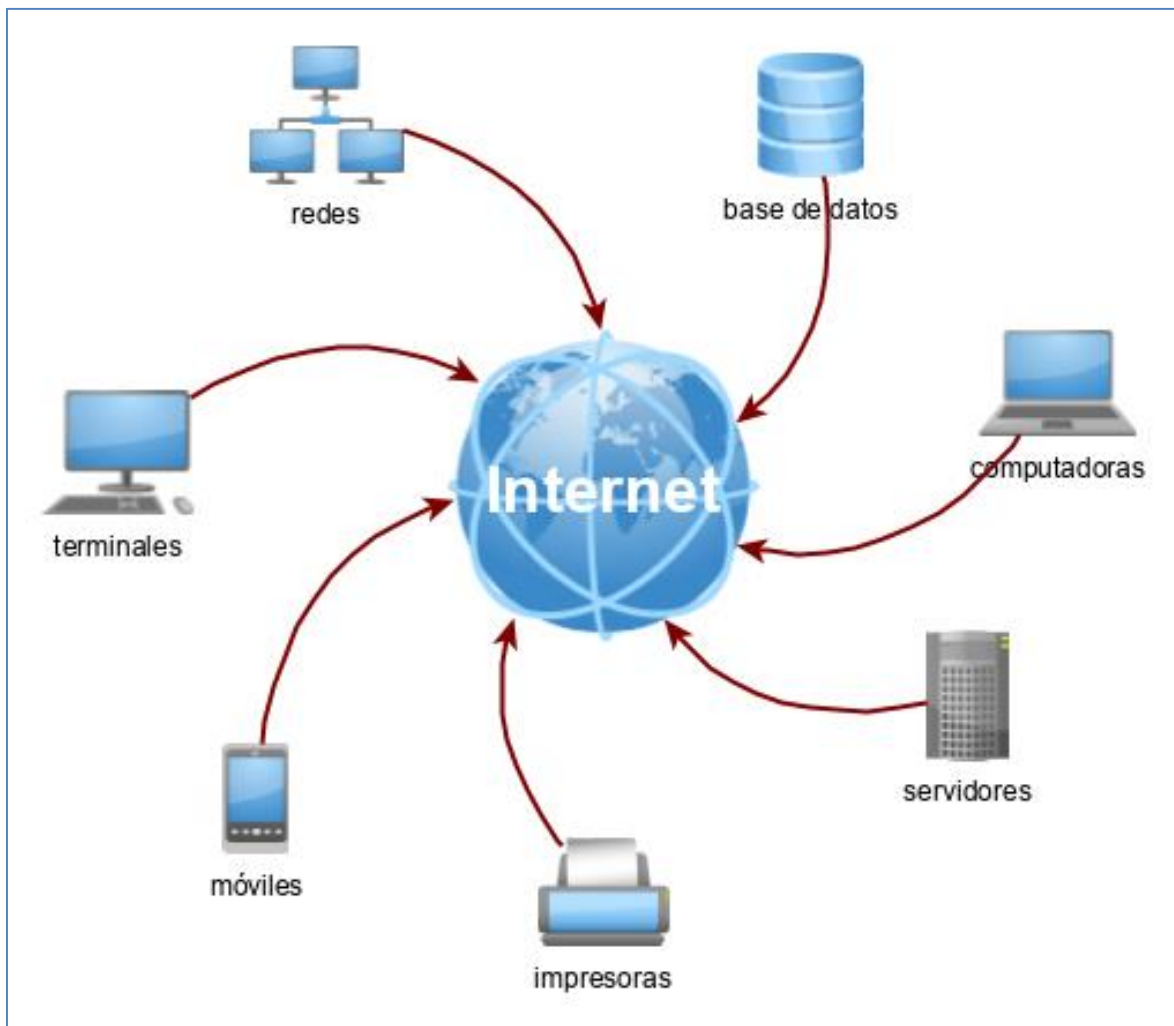


Figura 8: La internet [autoría propia].

La comunicación en internet se realiza mediante el uso de protocolos de comunicación, siendo el protocolo TCP/IP el principal sobre el cual corren otros protocolos; TCP/IP hace uso de las direcciones IP, esta dirección es un número único que asigna a cada computadora conectada a la red [9].

Un protocolo de comunicación es un conjunto de reglas establecidas que permiten el envío y recepción garantizada de datos de manera organizada a través de la red. El protocolo más conocido de internet es http y https que son la base del funcionamiento de la *web* y permiten la visualización de páginas *web*. Otros servicios populares en internet son el correo electrónico y la compartición de archivos directa mediante el protocolo FTP.

Tabla 1: Principales protocolos de internet.

Protocolo	Descripción
SMTP	Protocolo Simple de Transferencia de Correo. Es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.
POP3	Protocolo de Oficina de Correos. Permite recoger el correo electrónico en un servidor remoto.
IMAP	Protocolo de Acceso a Mensajes de Internet. Protocolo alternativo a POP3 pero que ofrece más posibilidades, Permite accesos simultáneos a diferentes clientes a una misma cuenta de correo electrónico.
FTP, SFTP	Protocolo de Transferencia de Archivos. Permite que equipos remotos puedan compartir archivos de una manera eficaz, necesitándose configurar un servidor FTP de archivos que es accedido mediante un cliente FTP.
HTTP, HTTPS	Permite la transferencia de archivos (principalmente en formato HTML) entre un servidor web y un cliente (navegador web), los recursos son localizados mediante una URL (Localizador Uniforme de Recursos).
TELNET	<i>Telecommunication Network</i> , permite emular una terminal (consola en modo texto) de una computadora de manera remota, de tal manera que un usuario puede acceder de manera distante a una computadora usando una interface que parece ser parte del sistema local del usuario.
DHCP	El Protocolo de Configuración Dinámica de Host es un protocolo de red de tipo cliente/servidor mediante el cual un servidor DHCP asigna dinámicamente una dirección IP y otros parámetros de configuración de red a cada dispositivo en una red para que puedan comunicarse con otras redes IP.

Los usuarios se conectan a la red a través de diferentes medios físicos, entre los más usados están vía líneas telefónicas (ADSL), por cable (coaxial), por fibra óptica, vía satélite y vía inalámbrica.

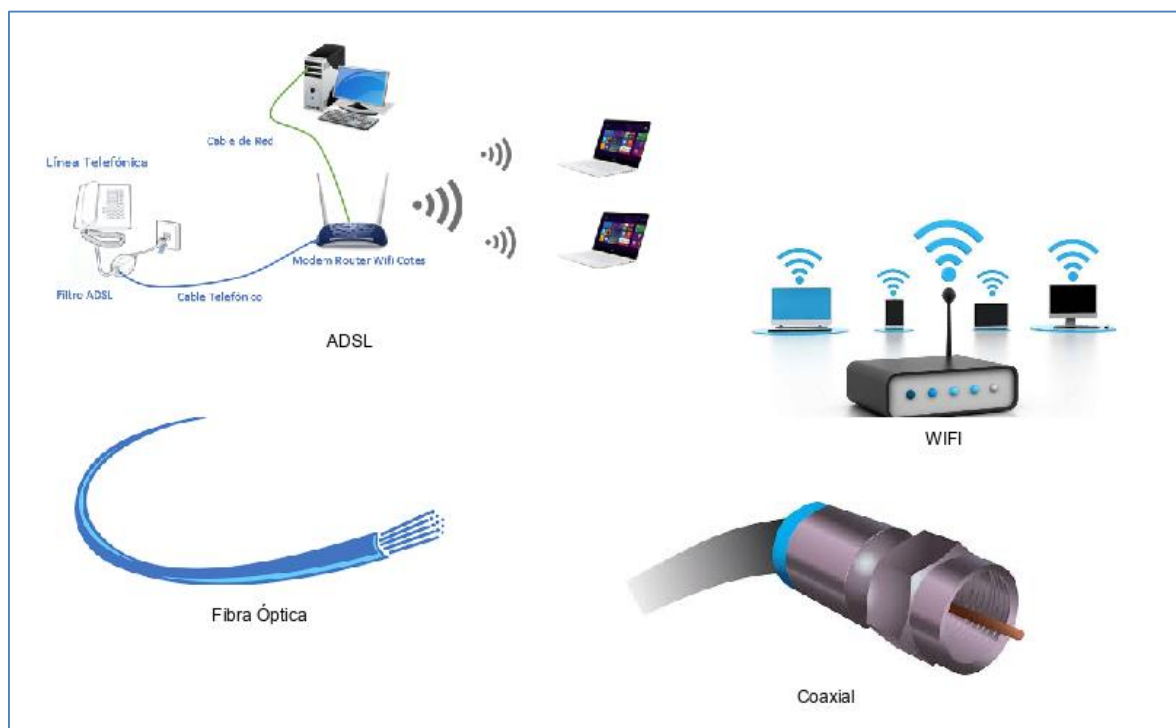


Figura 9: Medios físicos de conexión [autoría propia].

Actualmente el internet es el principal medio de comunicación del mundo, debido a las múltiples formas de comunicación que ofrece, así tenemos que se pueden hacer llamadas telefónicas, video conferencia, transmisiones en vivo, televisión, compartición de archivos, paginas multimedia, radio y demás formas, todo sobre internet agregando un importante factor: la comunicación directa entre el usuario y el emisor de la información.

Un factor que ha impulsado el desarrollo y expansión de internet ha sido el cómputo móvil mediante el uso de *Smartphone*, dando pie al desarrollo de nuevas tecnologías basadas en internet como es la comunicación asíncrona en el esquema cliente – servidor, notificaciones, etc.

El cambio continuo de internet nos lleva a su nueva etapa: el internet de las cosas (IoT), en donde, básicamente se busca que cada objeto en el hogar u oficina se conecte a la red convirtiéndose así en un objeto “inteligente” con el fin de automatizar y resolver tareas de forma transparente y así facilitar la vida de los humanos.



Figura 10: El internet de las cosas [19].

4.2 CONCEPTOS TÉCNICOS

4.2.1 *Lenguaje de programación*

Se define como el conjunto de reglas sintácticas y semánticas y símbolos que se utilizan para la creación de algoritmos (software) que le indican a la computadora que procesos realizar a fin de llevar a cabo una tarea específica.

A lo largo de la historia de la computación se han desarrollado múltiples lenguajes de programación, sin embargo, estos se dividen en dos categorías muy diferenciadas: lenguajes de bajo nivel y lenguajes de alto nivel [10].

- Lenguajes de bajo nivel. Son lenguajes diseñados especialmente para un tipo de hardware, por lo tanto, son altamente dependiente de la máquina sobre la cual se ejecutan. La ventaja es que aprovechan al máximo los recursos hardware donde se ejecutan, además que se ejecutan con mayor rapidez que los lenguajes de alto nivel. Una desventaja es que producen programas enormes donde encontrar un error es bastante difícil. Un ejemplo de estos lenguajes es el lenguaje ensamblador.
- Lenguajes de alto nivel. Son lenguajes en donde sus instrucciones son muy semejantes al lenguaje natural humano en vez del lenguaje máquina de ceros y unos. Estos lenguajes son independientes del hardware subyacente, por lo tanto, fácilmente pueden ejecutarse en una máquina o en otra. Le permiten al programador enfocarse en el desarrollo de la solución dejando a un lado la comprensión del funcionamiento interno de la máquina. Ejemplos de estos lenguajes son PHP, C, Pascal, entre otros.

4.2.2 *Framework de desarrollo*

Un *framework* es un conjunto de conceptos y reglas estandarizados para el desarrollo y organización de programas de software. Estas reglas definen nomenclaturas de nombres de archivos, de nombres de tablas, estructuras de los archivos de código fuente, así como la metodología de desarrollo de software. El objetivo principal de los *framework* de desarrollo es la facilitación, organización y agilización de la escritura de código,

implementar la reutilización de código, así como promover las buenas prácticas del desarrollo de software [11]. Entre las ventajas del uso de *frameworks* están:

- Proporcionan una estructura de archivos y directorios a los que el programador debe ajustarse, por lo tanto, el programador no necesita plantearse esta cuestión y se enfoca más en el desarrollo de la solución.
- Facilita la colaboración. Al existir reglas y nomenclaturas definidas para la creación del proyecto, es más fácil lidiar con código ajeno y el mantenimiento es más fácil de esta manera.
- Rutinas de código como utilidades, librerías o *plugins* pueden encontrarse ya desarrollados por lo tanto el desarrollo se agiliza y se vuelve más fácil.

El uso de un *framework* de desarrollo implica la aplicación de la lógica de desarrollo dada por el mismo. La gran mayoría aplica la lógica de MVC (modelo – vista – controlador), esto es, la estructura del software es dividida en tres capas:

- Modelo. Los archivos de esta capa se encargan del acceso a datos, en ellos se define la tabla a la que tienen acceso, su llave primaria y la relación de la tabla que representan con respecto a las demás dentro de la base de datos, reglas, restricciones de tabla, entre otros atributos.
- Controlador. Los archivos de esta capa se encargan del procesamiento de los datos obtenidos de la base de datos o bien de los ingresados mediante las vistas. Esta capa intermedia sirve de enlace entre la vista (usuario) y el modelo (acceso a datos).
- Vista. Esta capa es a la que el usuario visualiza (interfaz gráfica), carece del procesamiento de datos y es encargada de mostrarle al usuario los datos formateados recuperados mediante el modelo y procesados mediante el controlador.

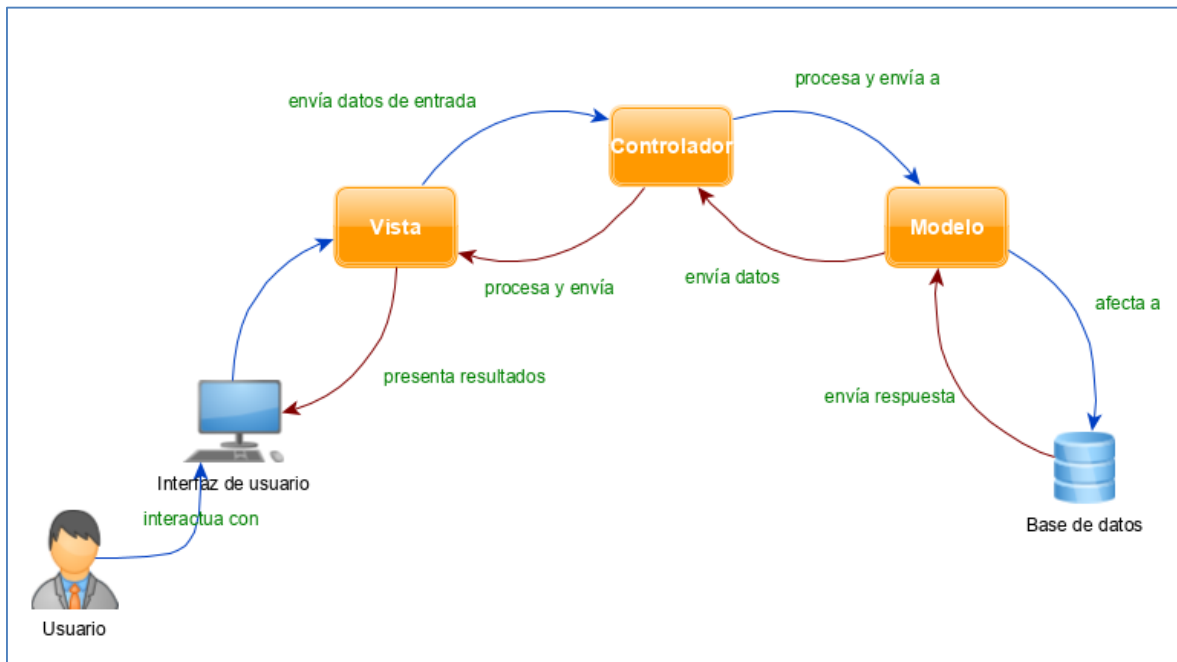


Figura 11: El funcionamiento de la metodología modelo – vista - controlador [autoría propia].

Gran parte de los lenguajes de programación, sobre todo los orientados al desarrollo de sistemas basados en *web*, cuentan con *frameworks* de desarrollo, ya sea oficiales o desarrollados por su comunidad de programadores. En el caso de PHP los más populares son Zend Framework (*framework* oficial), Cakephp, Laravel, Codeigniter, Yii, entre otros.

4.2.3 Paradigmas de programación

Un paradigma se define como un modelo o patrón de técnicas para diseñar soluciones. En el contexto del desarrollo de software un paradigma de programación lo definimos como una propuesta tecnológica para el desarrollo de software aprobada, usada y continuamente mejorada por un grupo de programadores con el objetivo de diseñar soluciones en lo que se refiere a la automatización y esquematización de procesos. Actualmente son dos los paradigmas de programación usados: programación estructurada y programación orientada a objetos.

- Programación estructurada. Surge a finales de la década de los 70s, una de sus mejoras era la eliminación de la etiqueta “*goto*” que es una instrucción de transferencia

incondicional, esto es, que permite realizar “saltos dentro del código” durante la ejecución de un programa, consecuencia del uso de esta instrucción son programas difíciles de entender y mantener (código espagueti). La programación estructurada omite esta práctica y sostiene que los scripts pueden ejecutarse usando únicamente tres tipos de instrucciones (Teorema del Programa Estructurado, Böhm-Jacopini, 1966): instrucciones de secuencia, condicionales y de iteración condicionales (ciclos). La programación estructurada también permite la definición y uso de funciones y procedimientos y algunos lenguajes permiten la inclusión externa de scripts. Ejemplo de lenguajes que soportan programación estructurada son FORTRAN, C, Pascal, entre otros.

Estas son sus ventajas y desventajas [7]:

Ventajas:

- Los programas son más fáciles de leer.
- La estructura de los programas es clara.
- Reducción de esfuerzo en las pruebas.
- Se incrementa el rendimiento del programador.

Desventajas:

- Se genera un único bloque de programa que cuando se hace demasiado grande puede ser problemático de entender y modificar.
- Programación orientada a objetos (POO). Surge en la década de los 70s, se basa en la utilización de objetos para la organización y ejecución del código en un programa. Un objeto es definido como una abstracción de algún hecho o ente del mundo real, con atributos que representan sus características o propiedades, y métodos que emulan su comportamiento o actividad [8]. El paradigma de POO está definido por las siguientes características:

- Abstracción. Denota las características esenciales de un objeto, donde se capturan sus comportamientos.
- Encapsulamiento. Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.
- Ocultación. Asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas.
- Polimorfismo. Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.
- Herencia. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.
- Recolección de basura. Es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos.

Las ventajas y desventajas de este paradigma son:

Ventajas:

- Reusabilidad.
- Mantenibilidad.
- Modificabilidad.
- Fiabilidad.

Desventajas:

- La ejecución de programas orientados a objetos tiende a ser más lenta.
- Se necesita aprendizaje y entrenamiento para usar bibliotecas de clases.

- Costo en tiempo y esfuerzo para la comprensión de la abstracción del paradigma de POO.

4.2.4 *Aplicación móvil*

Una aplicación móvil comúnmente conocida como “*app*” es una aplicación informática desarrollada exclusivamente para ser ejecutada en teléfonos o dispositivos inteligentes, y debido a esto son capaces de explotar al máximo los recursos de hardware que estos dispositivos ofrecen.

En el mundo de los dispositivos inteligentes son tres las plataformas que dominan el mercado: Android, IOS y Windows Phone, estos tres sistemas operativos móviles concentran más del 90% del mercado mundial. Por mucho Android es quien domina el mercado, sus aplicaciones son desarrolladas en el lenguaje de programación Java y son distribuidas de manera oficial en su repositorio en línea llamado Google Play. En el caso de IOS el lenguaje usado para el desarrollo de *apps* es Swift y Objective C y son distribuidas en la App Store de Apple. C++ es el lenguaje usado para el desarrollo de *apps* para Windows Phone y estas son distribuidas en la tienda virtual Windows Phone Store. Además de estos sitios propietarios en línea también existen otros terceros donde los desarrolladores pueden alojar sus *apps* para que los usuarios los descarguen, un ejemplo de estos sitios no oficiales es Amazon AppStore [9].

Existe otro concepto relacionado con los dispositivos móviles, las llamadas “*web app*”, estas no son otra cosa sino versiones de las *web* tradicionales adaptadas para una óptima visualización en las pantallas de los dispositivos móviles. Actualmente no es necesario hacer dos versiones de la misma página, gracias a librerías de estilos CSS como Bootstrap se puede hacer un sitio responsivo, esto es, que se ajuste automáticamente a la pantalla donde se despliega ya sea una computadora o un teléfono móvil. Una *web app* no debe confundirse con una aplicación móvil nativa, ya que esta última es capaz de hacer un uso más profundo del hardware del móvil.

Las ventajas de una aplicación nativa son:

- Acceso completo al hardware del dispositivo.
- Mejora en la experiencia de usabilidad que experimenta el usuario.
- Envío de notificaciones a los usuarios.
- Actualizaciones automáticas de la aplicación.

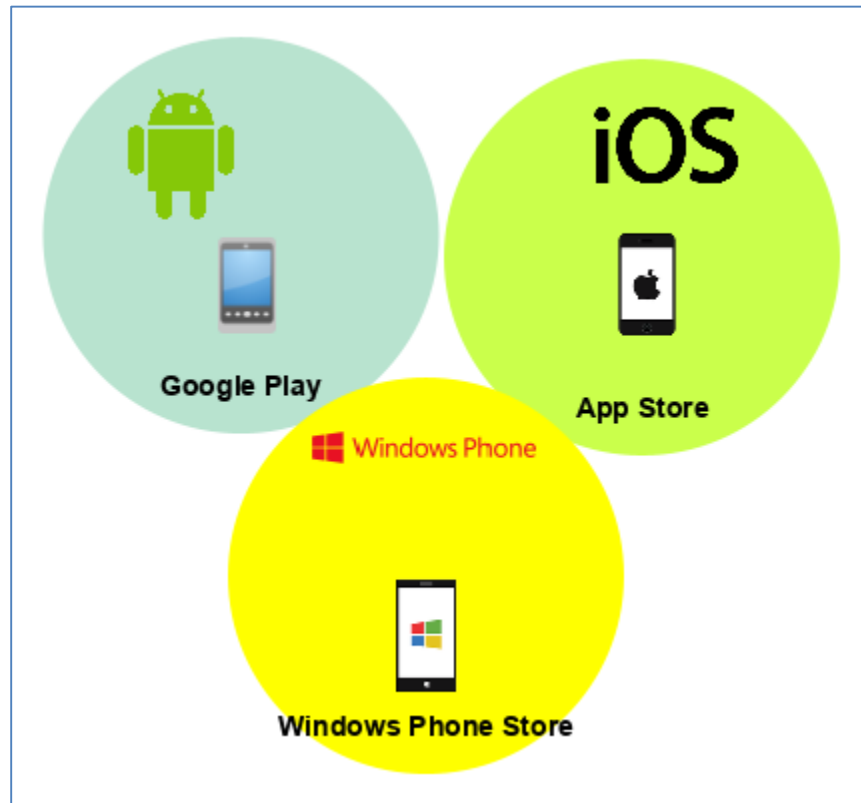


Figura 12: Principales plataformas de las aplicaciones móviles [autoría propia].

4.2.5 Modelos de comunicación de datos

4.2.5.1 Cliente - servidor

Se trata de un modelo de comunicación integrado por dos partes: el proveedor de servicios llamado “servidor” y el demandante de dichos servicios llamado “cliente”. Tanto el cliente como el servidor son software que asumen sus funciones propias, por lo tanto, pueden estar en la misma máquina sin embargo por lo general cliente como servidor se

encuentran en máquinas separadas. Un ejemplo de la aplicación de este modelo al contexto informático son los servidores de impresiones, el correo electrónico y el más importante de todos: el internet.

La comunicación en este modelo siempre inicia cuando el cliente envía una solicitud de información al servidor, este responde informando si está o no disponible, en caso de estarlo, mediante un protocolo de comunicación se negocia y establece una conexión entre ambas máquinas, iniciando de esta manera la transferencia de información, entonces el cliente recibe la información y por lo general la muestra al usuario [12].

Ventajas:

- Centralización de los recursos. Estos al estar centralizados son más fáciles de proteger de accesos no deseados.
- Escalabilidad. La capacidad de clientes o servidores pueden ser aumentados sin afectar al sistema.
- Permite el intercambio de información entre sistemas de diferente plataforma (sistema operativo). Al llevar a cabo la comunicación sobre protocolos (basados en estándares) es fácil y confiable la comunicación entre cliente y servidor.

Desventajas:

- Susceptible a congestión por tráfico. Debido a la centralización de las peticiones un servidor puede verse afectado en su funcionamiento cuando múltiples clientes realizan peticiones, esto llevado a gran escala y hecho de manera mal intencionado por un atacante es lo que se llama ataque DOS (denegación de servicio).
- Continuidad del servicio. Esta desventaja se debe a la centralización de los servicios y recursos, en un dado caso de falla en el servidor, ningún cliente podrá acceder a sus servicios, esto puede ser solucionado mediante la implementación de servidores espejo que entran en servicio tras la caída de otro, sin embargo esto eleva el costo económico de implementación además de implicar más complejidad de configuración.

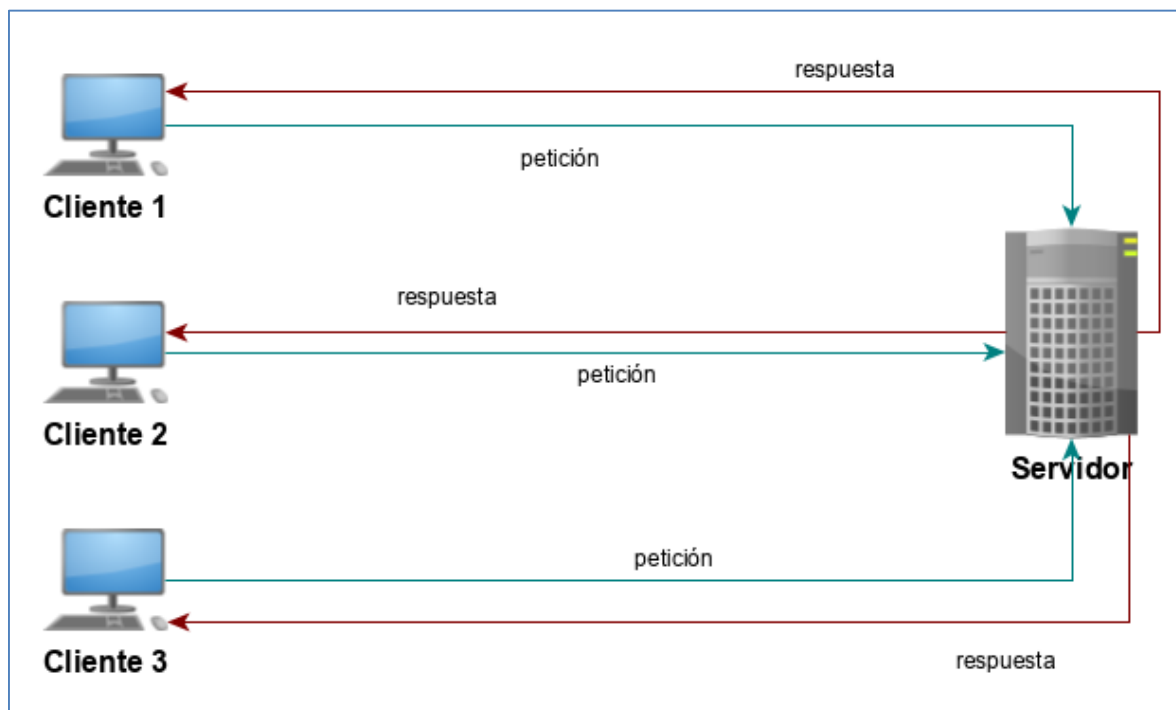


Figura 13: Modelo Cliente - Servidor [autoría propia].

4.2.5.2 Maestro - esclavo

También conocido como modelo maestro – trabajador, en este modelo de comunicación es el maestro quien inicia el proceso de comunicación y quien tiene un control unidireccional sobre los esclavos. Este requerimiento de comunicación puede ser hacia uno o varios esclavos y la ubicación de los datos o de las aplicaciones o equipos es totalmente transparente para el maestro. Maestro y esclavo representan a cualquier recurso de cómputo o proceso que intervienen en la comunicación [14].

Este paradigma es adecuado, generalmente, para arquitecturas tanto de memoria compartida como memoria distribuida y se usa cuando se tienen dos o más procesos y se requieren ejecutar simultáneamente pero a velocidades diferentes.

Un ejemplo claro de este modelo lo podemos ver en la replicación de bases de datos, en donde la base de datos maestra es considerada como el recurso maestro y las bases en donde se replica la información consideradas como esclavas y que se sincronizan a petición del recurso maestro.

Ventajas:

- Ventajoso para aplicaciones multitarea.
- Le da un enfoque más modular al desarrollo de la aplicación.

Desventajas:

- El proceso / equipo maestro puede convertirse en un cuello de botella, esto sucedería si la carga de trabajo de cada esclavo es pequeña generando muchas comunicaciones con el maestro.
- Posible pérdida de datos y falla en su integridad cuando el maestro falla.
- Cada nuevo esclavo añade más carga de trabajo al maestro.
- La aplicación podría ser reseteada.

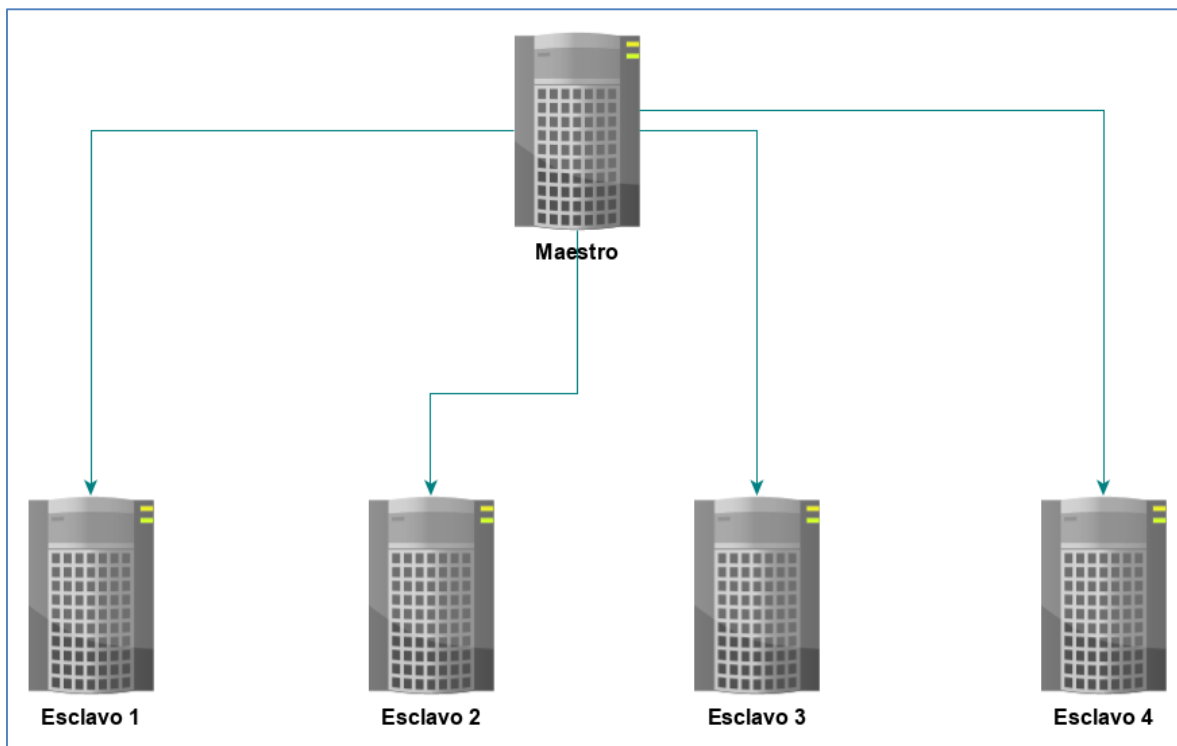


Figura 14: Modelo Maestro - Esclavo [autoría propia].

4.2.5.3 Productor - consumidor

Este esquema consiste en uno o más productores y uno o más consumidores. Productores y consumidores pueden representar dispositivos de cómputo o procesos. Estos productores y consumidores tienen acceso a un medio en común, podría tratarse, por ejemplo, de un buffer de datos. El productor produce algún dato y lo añade al buffer, los consumidores esperan por algo que sea añadido al buffer, leen el elemento y lo procesan [15].

Este es un ejemplo clásico de un problema de sincronización de multiprocesos. El problema es asegurarse que el productor no trate de añadir datos en el buffer si éste está lleno y que el consumidor no trate de recuperar datos de un buffer vacío.

La solución para el productor es dormir o descartar datos si el buffer está lleno. Una vez que el consumidor toma el dato del buffer, se le notifica al productor quien comienza a llenar el buffer nuevamente. De la misma manera el consumidor se pone a dormir si encuentra que el buffer está vacío. Una vez que el productor pone datos en el buffer el consumidor es despertado. Esta solución puede ser lograda mediante la implementación de algoritmos del tipo semáforos.

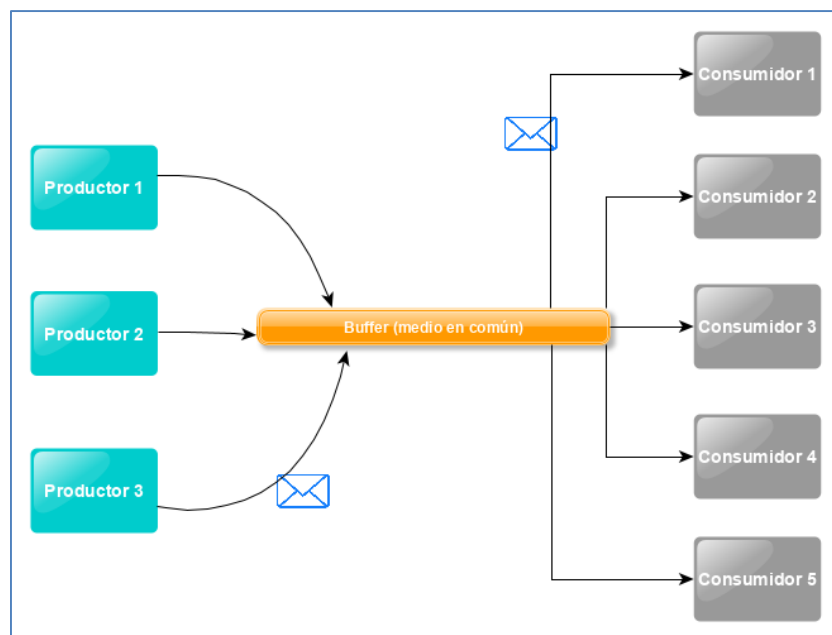


Figura 15: Modelo Productor - Consumidor [autoría propia].

4.2.5.4 Publicador - suscriptor

El sistema *publish/subscribe* es un paradigma de mensajes asíncronos donde los que envían (*publisher*) mensajes no están programados para enviar sus mensajes a receptores específicos (*subscribers*), sino que se envían a algún tipo de servidor. Los mensajes publicados se caracterizan por clases o tópicos, sin tener constancia de los suscriptores que pueda haber. Los suscriptores expresan interés en una o más clases, y solo reciben mensajes de ese mismo interés, sin tener constancia de qué publicadores hay [16].

En este modelo los suscriptores reciben un sub conjunto del total de mensajes publicados. Este proceso de selección de mensajes para recepción y procesamiento es llamado *filtering* (filtrado), hay dos formas de *filtering*:

- Sistema basado en tópicos: los mensajes son publicados a tópicos o canales nombrados lógicos. Los suscriptores en un sistema basado en tópicos recibirán todos los mensajes publicados a los tópicos a los cuales ellos están suscritos, todos los suscriptores a un tópico recibirán el mismo mensaje. El publicador es responsable de definir la clase de mensaje al cual los suscriptores pueden suscribirse.
- Sistema basado en contenido: los mensajes son únicamente entregados a los suscriptores si el atributo o contenido de aquéllos mensajes coinciden con las restricciones definidas por el suscriptor. El suscriptor es responsable por la clasificación de los mensajes.

A continuación se listan algunas ventajas y desventajas de este modelo de comunicación [17]:

Ventajas:

- Bajo acoplamiento. Los publicadores están acoplados libremente a los suscriptores e incluso no necesitan saber de su existencia. Cada uno puede continuar funcionando según lo normal independientemente del otro.

- Escalabilidad. Publicador / suscriptor proporciona una oportunidad de mejor escalabilidad que el tradicional esquema cliente – servidor, a través de operaciones paralelas, cacheo de mensajes, etc. Este paradigma ha probado su escalabilidad proveyendo mensajería distribuida en internet a través de protocolos de sindicación *web* como RSS y Atom.

Desventajas:

- Problemas de entrega de mensajes.
 - El intermediario en un sistema publicador / suscriptor puede ser diseñado para entregar mensajes por un tiempo específico, pero luego deja de intentar realizar la entrega, ya sea que haya recibido o no confirmación de que todos los suscriptores recibieron el mensaje con éxito.
 - Un publicador puede asumir que un suscriptor está escuchando, cuando en realidad no lo está.
 - El modelo publicador / suscriptor se adapta bien a redes pequeñas con un número reducido de nodos de publicador y suscriptor y un volumen de mensajes bajo. Sin embargo, a medida que aumenta el número de nodos y mensajes, aumenta la probabilidad de inestabilidad, lo que limita la escalabilidad máxima de una red publicador / suscriptor.

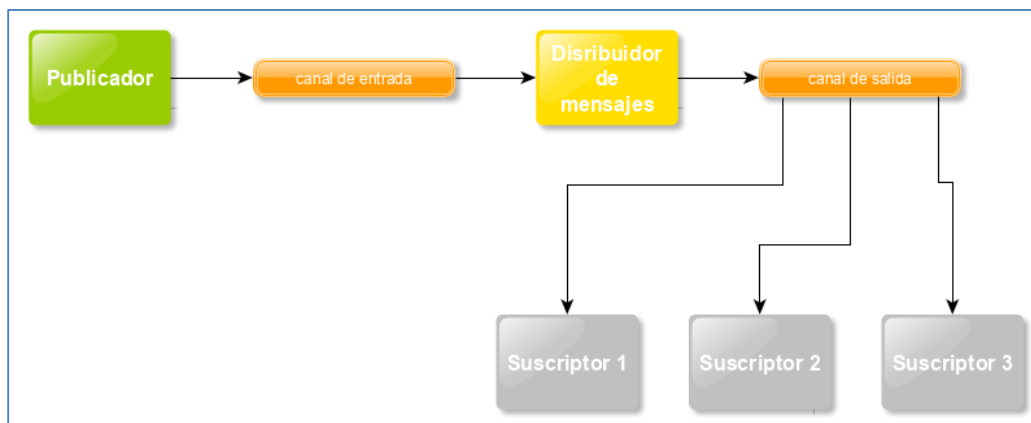


Figura 16: Modelo Publicador - Suscriptor [autoría propia].

Capítulo 5: Objetivos

5.1 OBJETIVO GENERAL

Desarrollar una plataforma tecnológica digital para el reporte, seguimiento y resolución de deficiencias en los servicios públicos del municipio de Othón P. Blanco, Quintana Roo, con el fin de optimizar dichos procesos mediante la aplicación e integración de las TIC's.

5.2 OBJETIVOS ESPECÍFICOS

- Analizar los procesos municipales para la determinación de los requerimientos del sistema *web* y de la aplicación móvil.
- Desarrollar un sistema de administración vía *web* basado en el análisis aprobado por el municipio de Othón P. Blanco.
- Desarrollar una aplicación móvil para Android basada en el análisis aprobado por el municipio de Othón P. Blanco.
- Llevar a cabo una evaluación funcional y documentación del sistema de administración vía *web* y de la aplicación móvil.

Capítulo 6: Hipótesis

El sistema integral para el reporte y seguimiento de deficiencias del servicio público del municipio de Othón P. Blanco, Quintana Roo permite registrar un reporte de manera más rápida y con información más exacta, de igual manera permite al ciudadano conocer en todo momento el estatus de su reporte y las acciones de seguimiento realizadas por parte del Ayuntamiento para su resolución. De esta manera se mejora el canal de comunicación entre el ciudadano y el Ayuntamiento en lo que se refiere al reporte y seguimiento de problemas en los servicios públicos municipales.

Capítulo 7: Metodología

1. Selección del tema de investigación. Realizar un análisis de las carencias y necesidades del proceso de seguimiento y resolución de los reportes de deficiencias en los servicios públicos en el municipio de Othón P. Blanco, Quintana Roo.
2. Revisión de la literatura. Investigación y revisión de los trabajos e investigaciones realizados acerca de soluciones informáticas en el ámbito de e-gobierno que sirvan como base para el desarrollo exitoso del proyecto que se plantea.
3. Análisis de requerimientos. Aplicación de técnicas de análisis de sistemas que permitan el levantamiento de la información necesaria para el desarrollo de una solución informática óptima que cumpla con los objetivos del presente proyecto.
4. Desarrollo del sistema de administración vía *web* para uso exclusivo del personal del ayuntamiento. Con base en el análisis realizado, desarrollar un sistema *web* que solucione las necesidades detectadas en el proceso.
5. Desarrollo de la aplicación móvil para uso de los ciudadanos. Crear una aplicación móvil para el ciudadano que realice los reportes y permita ver el seguimiento y resolución de estos.
6. Fase de pruebas. Periodo de realización de pruebas tanto del sistema *web* como de la aplicación móvil a fin de detectar posibles fallos y en su caso corregirlos.
7. Redacción de la tesis. Documentación del presente proyecto de acuerdo con el desarrollo realizado.

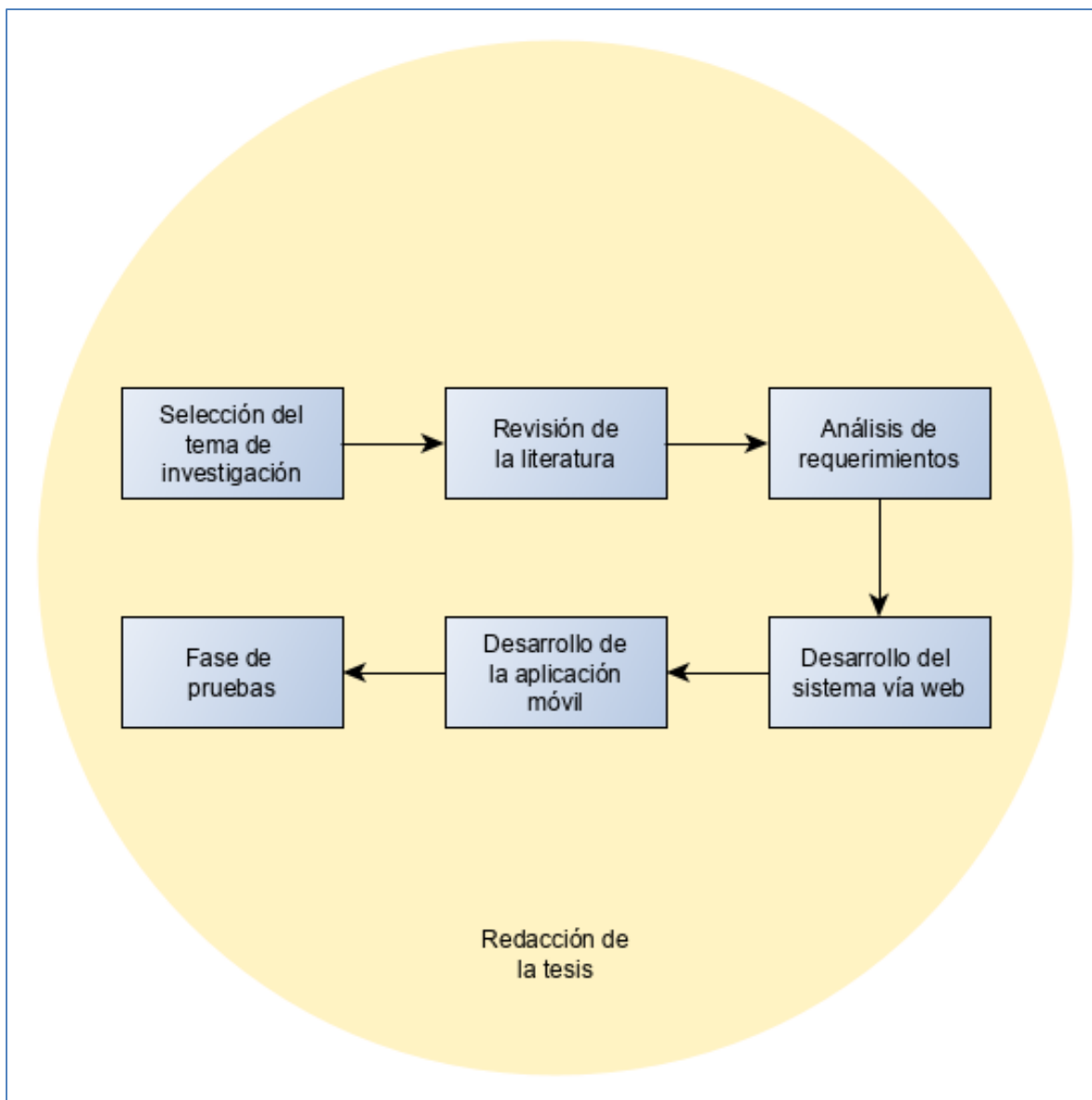


Figura 17: Metodología del desarrollo de tesis [autoría propia].

Capítulo 8: Resultados

8.1 RESULTADOS

Objetivo específico 1: “Analizar los procesos municipales para la determinación de los requerimientos del sistema web y de la aplicación móvil”.

Como primer resultado del análisis tenemos la siguiente figura que representa el diagrama de flujo de cómo aún opera el sistema 072 Ayuntatel, mismo que será reemplazado por el sistema 072 Reportes.

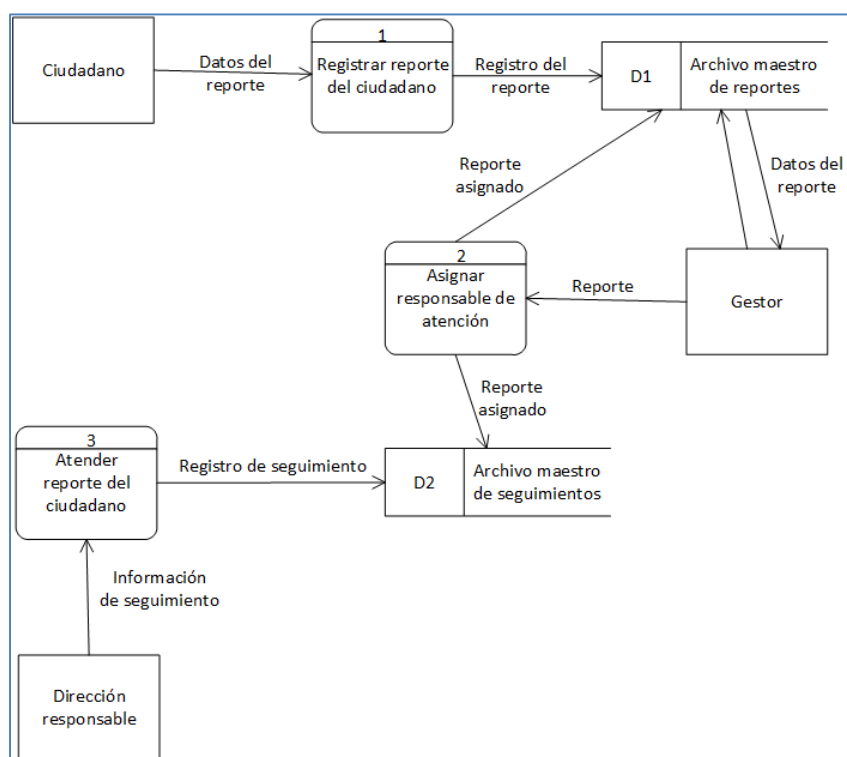


Figura 18: Funcionamiento del sistema actual [autoría propia].

El diagrama 0 del nuevo sistema 072 Reportes, es similar a la figura anterior, sin embargo, la diferencia está en el detalle de cada uno de los tres procesos, además de que para el registro de incidencias se agrega el nuevo canal de reporte vía dispositivos móviles (siguiente figura).

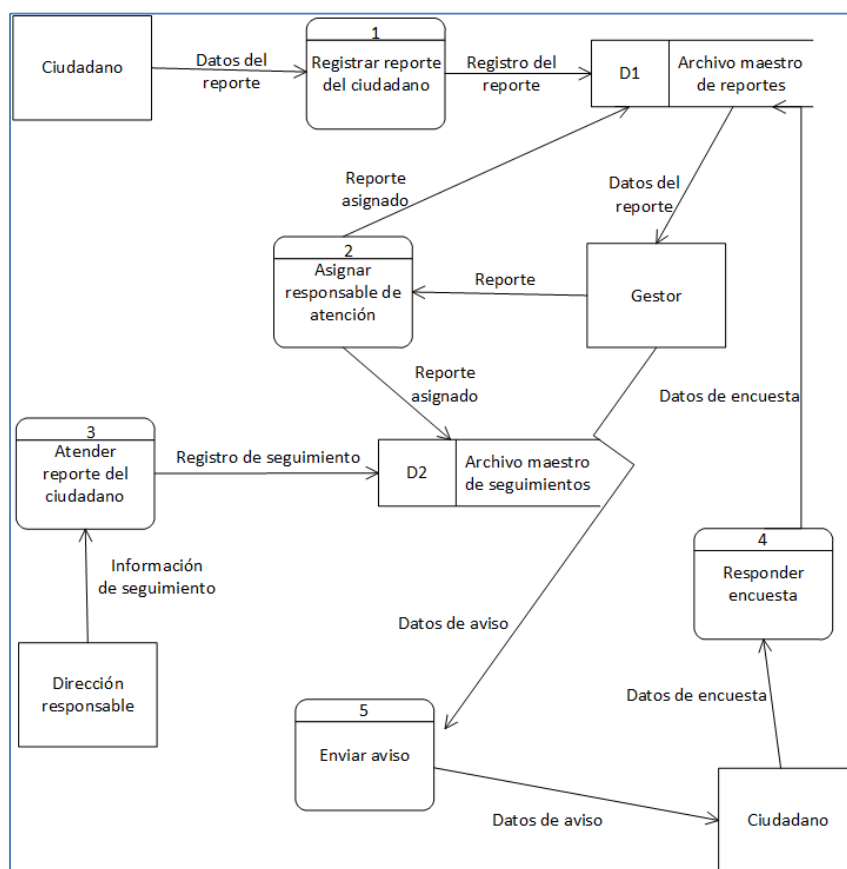


Figura 19: Diagrama 0 del nuevo sistema 072 Reportes [autoría propia].

La figura anterior muestra el diagrama 0 del nuevo sistema en donde también se agregan los procesos para la contestación de la encuesta de servicios desde la aplicación móvil además de la capacidad de enviar avisos de información desde el sistema y recibirlos en la aplicación.

Así tenemos que para el proceso 1, el diagrama de flujo de datos en el nuevo sistema es como se muestra a continuación:

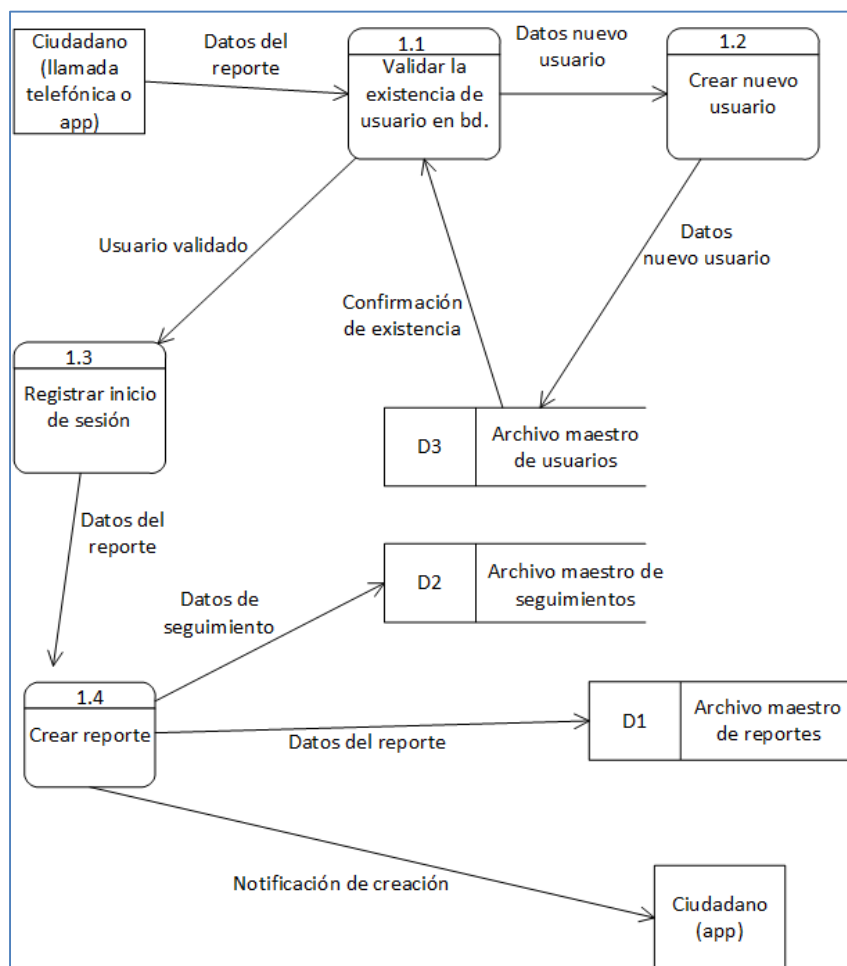


Figura 20: Diagrama del proceso 1 del nuevo sistema 072 Reportes [autoría propia].

Este proceso comienza cuando el ciudadano levanta un reporte ya sea vía telefónica o mediante el uso de la *app*, los datos del usuario son validados para descartar que ya exista en la tabla de usuarios (1.1), en caso de no existir se crea como nuevo usuario (1.2). Posteriormente se registra su inicio de sesión (1.3) y el registro del reporte es dado de alta (1.4), en este punto se afecta la tabla de reportes y se agrega un registro en la tabla de seguimientos, instantáneamente una respuesta de creación exitosa en el sistema es enviada a la *app* (en caso de que haya sido creada desde este medio) y es entonces cuando se finaliza el proceso de registro de reporte.

A continuación se muestra el diagrama de flujo correspondiente al proceso 2 del diagrama 0.

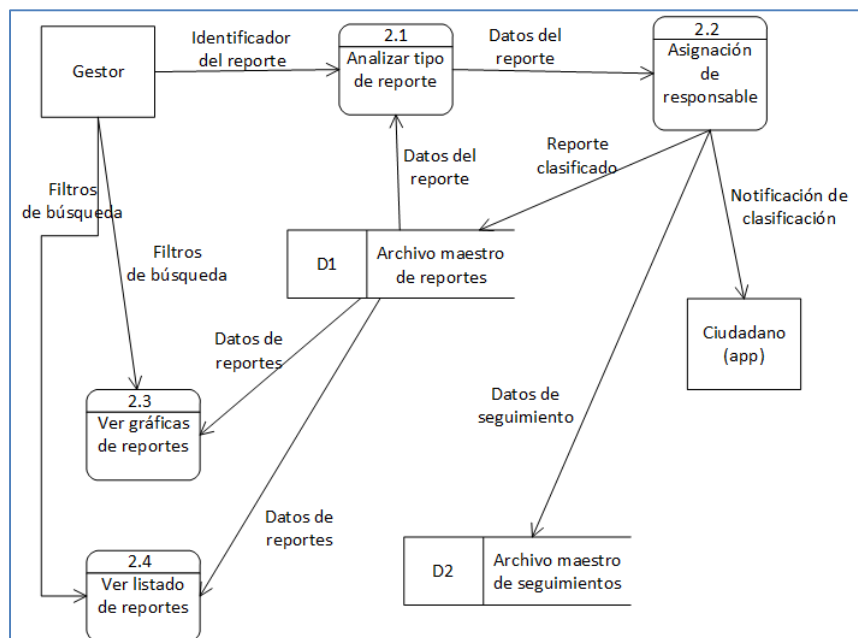


Figura 21: Diagrama del proceso 2 del nuevo sistema 072 Reportes [autoría propia].

El proceso 2 corresponde a asignar el responsable de atención de un reporte, el encargado de esta tarea es el usuario “Gestor”. El gestor analiza el tipo de reporte (2.1) que atiende y con base en su naturaleza le asigna un responsable (área) de atención (2.2), en este punto las tablas de reportes y seguimientos son afectadas y una notificación es enviada al ciudadano (en caso de que el reporte haya sido hecho desde la *app*) informándole que el reporte ha sido turnado a un área para su atención. El proceso 2 también abarca los subprocesos “ver graficas de reportes” (2.3) y “ver listado de reportes” (2.4) los cuales reciben parámetros de búsqueda, y consisten el primero en ver un listado de reportes y el segundo gráficas.

La figura siguiente muestra el diagrama para el proceso 3 “Atender reporte del ciudadano”, el cual consiste en registrar las acciones de seguimiento llevadas a cabo en el proceso de resolución, así como la solución final dada al reporte. El proceso comienza cuando el “Responsable de atención” revisa el reporte que le fue asignado por el gestor (3.1), una vez que las acciones de resolución van siendo hechas, estas son capturadas como acciones de seguimiento (3.2) insertando un registro por cada una en la tabla de

seguimientos, al mismo tiempo una notificación de informe de seguimiento es enviada a la *app* del ciudadano (en caso de que el reporte haya sido hecho desde la *app*).

De la misma manera en que se registran las acciones de seguimiento se registra la solución final dada al reporte (3.3): el responsable de atención registra la solución final en el sistema, las tablas de reportes y seguimiento son afectadas y una notificación de solución final es enviada a la *app* en caso de que el reporte haya sido levantado desde un dispositivo móvil. El proceso 3 se muestra en la siguiente figura:

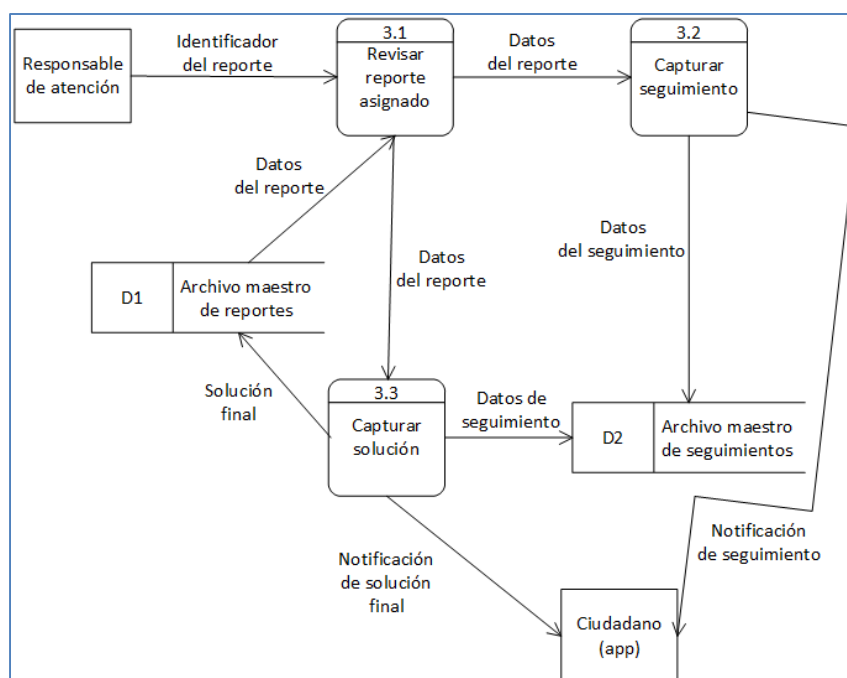


Figura 22: Diagrama del proceso 3 del nuevo sistema 072 Reportes [autoría propia].

El proceso 4 se lleva a cabo en la aplicación móvil y corresponde a la contestación de una encuesta de calidad del servicio una vez que el reporte ha sido concluido. Comienza con la notificación de solución final de reporte que llega a la *app* lo cual cambia el estado del reporte a “concluido”, en este punto las tablas de reportes y seguimiento (en el móvil) son afectadas. El ciudadano entonces puede ingresar al detalle del reporte y contestar la encuesta de satisfacción asociada a dicho reporte. Las respuestas ingresadas son guardadas

de manera local así como enviadas a la base de datos del sistema en internet. El proceso se muestra en la siguiente figura.

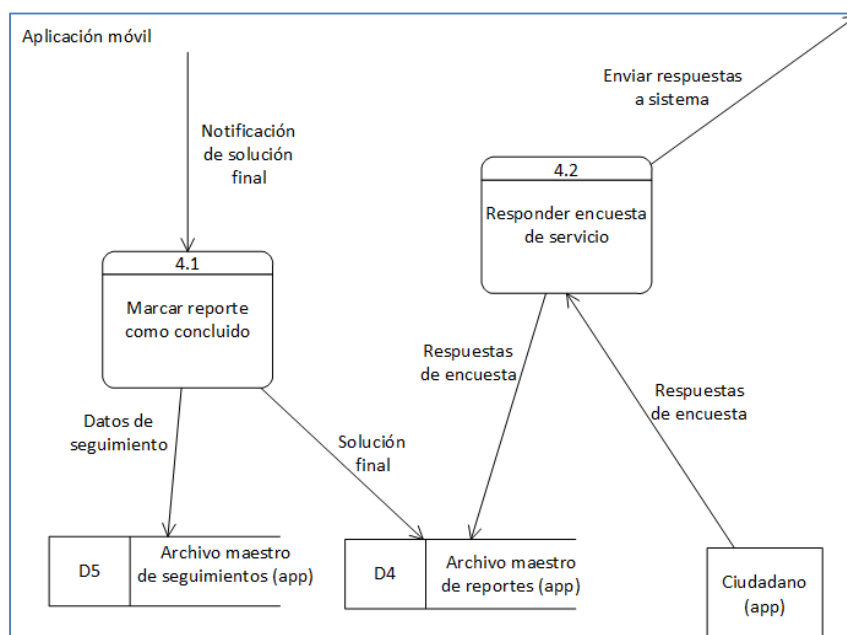


Figura 23: Diagrama del proceso 4 del nuevo sistema 072 Reportes [autoría propia].

El proceso 5 se refiere al envío de avisos de información general (que el ayuntamiento considera que la población necesita saber) a los dispositivos móviles. Comienza cuando el gestor ingresa al sistema los datos del nuevo aviso (texto e imagen) (5.1) en ese momento un nuevo registro es agregado a la tabla de avisos del sistema. Posteriormente el gestor procede a hacer el envío del aviso (5.2) el cual llega a la *app* del ciudadano y entonces la tabla de avisos en local es afectada guardando en ella los datos del nuevo aviso (5.3). El proceso se muestra en la siguiente figura.

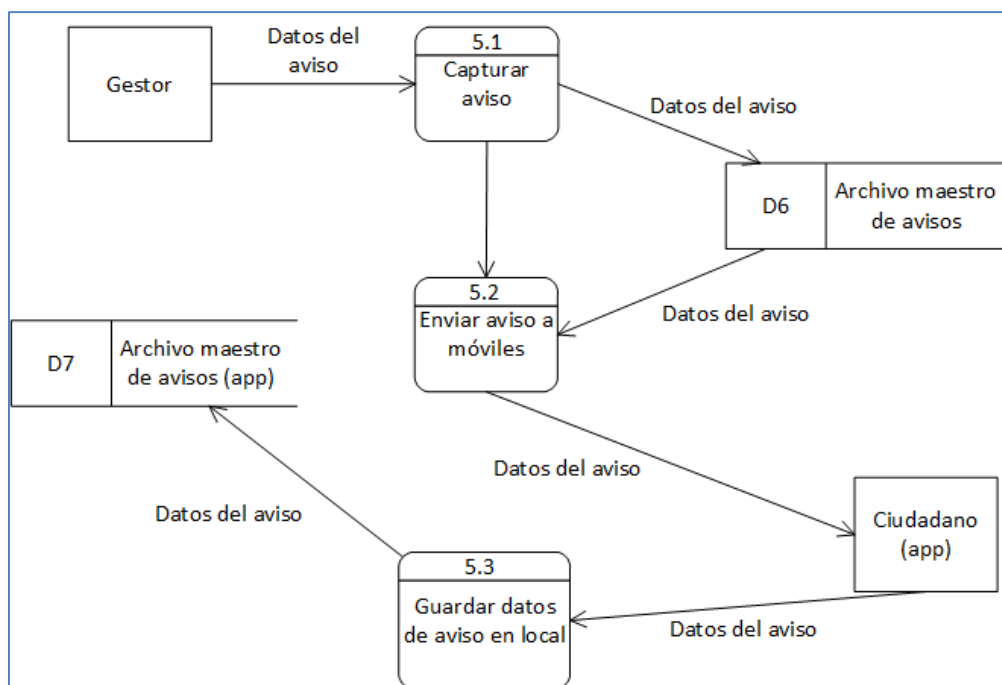


Figura 24: Diagrama del proceso 5 del nuevo sistema 072 Reportes [autoría propia].

Un segundo resultado de esta etapa de análisis son los diagramas entidad relación de las bases de datos requeridas por el nuevo proyecto, estas bases de datos son las del sistema web administrado por personal del Ayuntamiento y la base de datos local que es usada por la *app* en cada dispositivo móvil. El modelo entidad relación de la base de datos necesaria para la aplicación móvil es como se muestra en la figura 25.

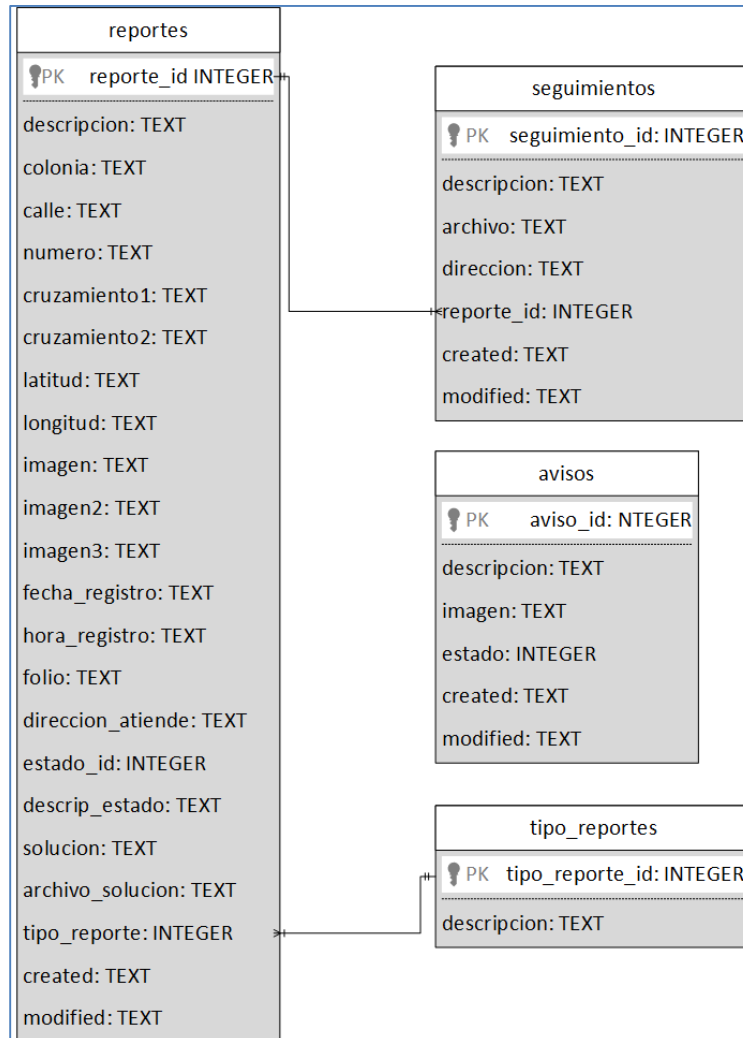


Figura 25: Diagrama ER de la base de datos en el dispositivo móvil [autoría propia].

En la siguiente figura se detalla el diagrama entidad – relación para la base de datos implementada por el sistema web utilizado por el personal del Ayuntamiento:

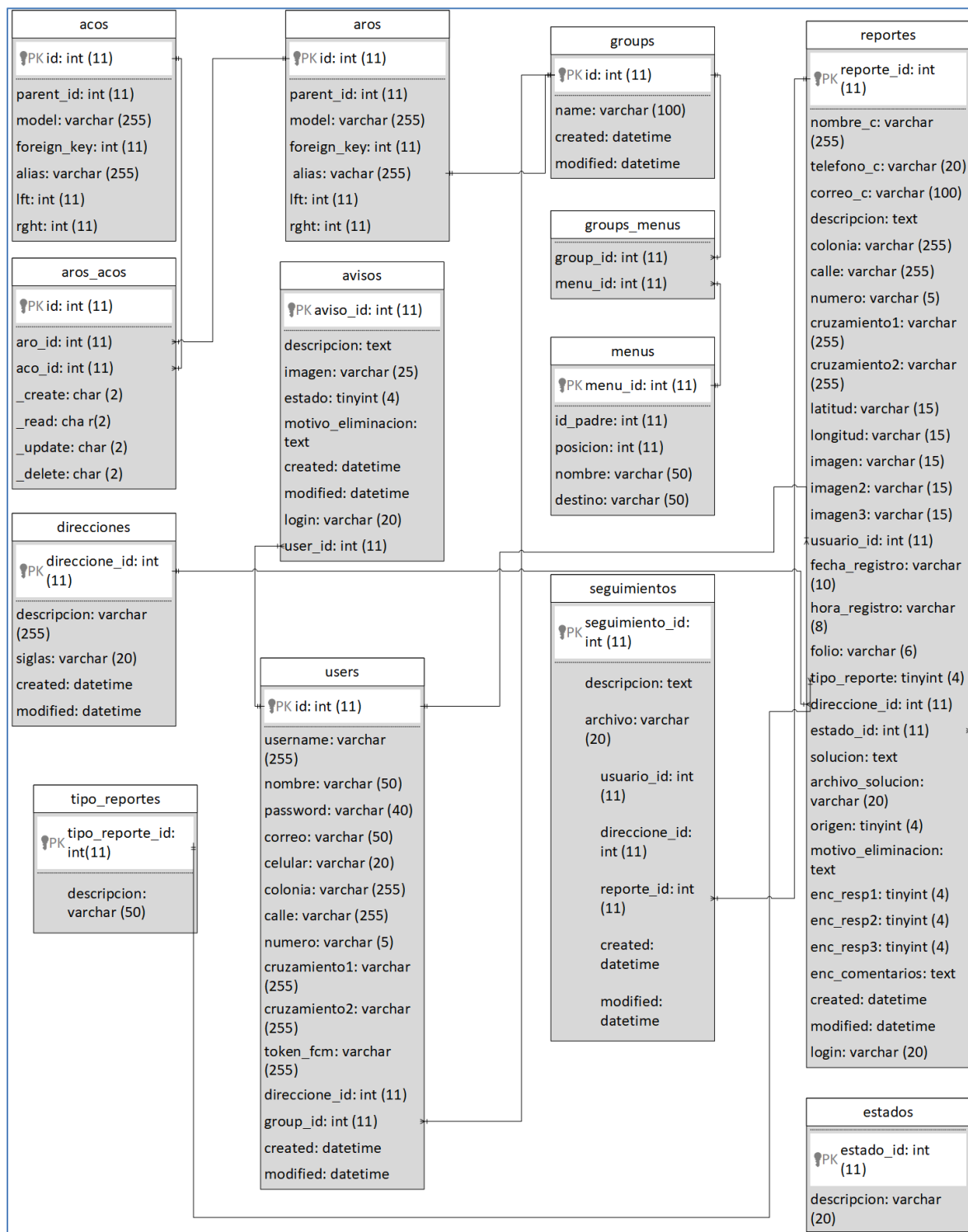


Figura 26: Diagrama ER de la base de datos del sistema web [autoría propia].

Producto del análisis se encontró la siguiente lista de requerimientos funcionales que el sistema web y la aplicación móvil deberían de cubrir:

Tabla 2: Requerimientos del proyecto 072 Reportes.

No.	Descripción
1	El acceso al sistema será mediante un usuario y contraseña entregados a todos los usuarios que necesiten tener acceso a él.
2	El diseño de la interfaz del sistema deberá ser responsiva para que pueda mostrarse correctamente en pantallas de diferente tamaño.
3	La sesión en el sistema deberá tener un periodo máximo de validez, después del cual deberá terminarse.
4	Todos los listados de registros deberán ser ordenados por fecha de creación en orden descendente.
5	Todos los campos requeridos para la creación de un reporte, seguimiento o solución final deberán ser obligatorios.
6	En todos los formularios antes de guardar, editar o eliminar deberá mostrarse una ventana de confirmación para que el usuario autorice la acción que está por realizarse.
7	Los archivos adjuntos que se permiten en todo el sistema deberán ser imágenes en formato jpg.
8	Si el reporte fue realizado desde la <i>app</i> se enviará una notificación <i>push</i> al dispositivo móvil cuando: se abra el reporte por primera vez, cuando capture un nuevo seguimiento o bien, cuando se capture la solución final.
9	Se enviará una notificación <i>push</i> al móvil cuando se capture y envíe un nuevo aviso.
10	Deberá existir en el sistema un mapa en donde se indiquen los puntos geográficos

	en donde se hicieron los reportes.
11	En el mapa anterior los marcadores deberán tener códigos de color para saber el estado en que se encuentran los reportes.
12	Deberán existir en el sistema tres tipos de gráficas: por estado del reporte, por unidad administrativa que atiende y por año de registro del reporte.
13	El aviso que se envía a los teléfonos móviles deberá consistir de un texto descriptivo y una imagen opcional en formato jpg.
14	El sistema contará con tres tipos de usuario: administrador, gestor y unidad responsable.
15	Los integrantes del grupo administrador tendrán acceso total a todas las funcionalidades del sistema.
16	Los del grupo gestor podrán crear un reporte, designar responsable de atención, capturar acciones de seguimiento y resolución final del reporte, también podrán eliminar un reporte si así se requiere. Tendrán acceso al mapa y gráficas, así como a la creación y envío de avisos.
17	El sistema deberá ser codificado en el lenguaje PHP usando el motor de base de datos MySQL y un <i>framework</i> de desarrollo para garantizar un desarrollo rápido.
18	El sistema deberá ser accedido vía web mediante el uso de un navegador de internet. Deberá desarrollarse con estándares de diseño como html y css que garantice su correcto funcionamiento en cualquier navegador web.
19	Todos los datos que se capturen en la aplicación o que se reciban deberán ser almacenados en forma local en una base de datos.
20	La aplicación permitirá hasta 3 imágenes adjuntas al momento de enviar el reporte al sistema.
21	En la aplicación los datos obligatorios para levantar un reporte serán: descripción del reporte, colonia, calle, número de lote, cruzamiento 1, cruzamiento 2 y tipo de

	reporte.
23	Al momento de enviar el reporte al sistema se deberán recuperar las coordenadas geográficas del usuario y ser enviadas también.
24	La aplicación además del apartado de captura de reportes deberá tener un apartado de consulta en mapa los repostes hechos, otro apartado de consulta en listado de los reportes realizados y por último un apartado de avisos, que es donde podrá localizar los avisos enviados desde el sistema a los dispositivos móviles.
25	Aviso de privacidad y términos y condiciones de uso

Objetivo específico 2: “Desarrollar un sistema de administración vía web basado en el análisis aprobado por el municipio de Othón P. Blanco”.

En este punto se muestra a continuación el diagrama de navegación entre las pantallas del sistema de acuerdo al orden en que son accedidas por el usuario.

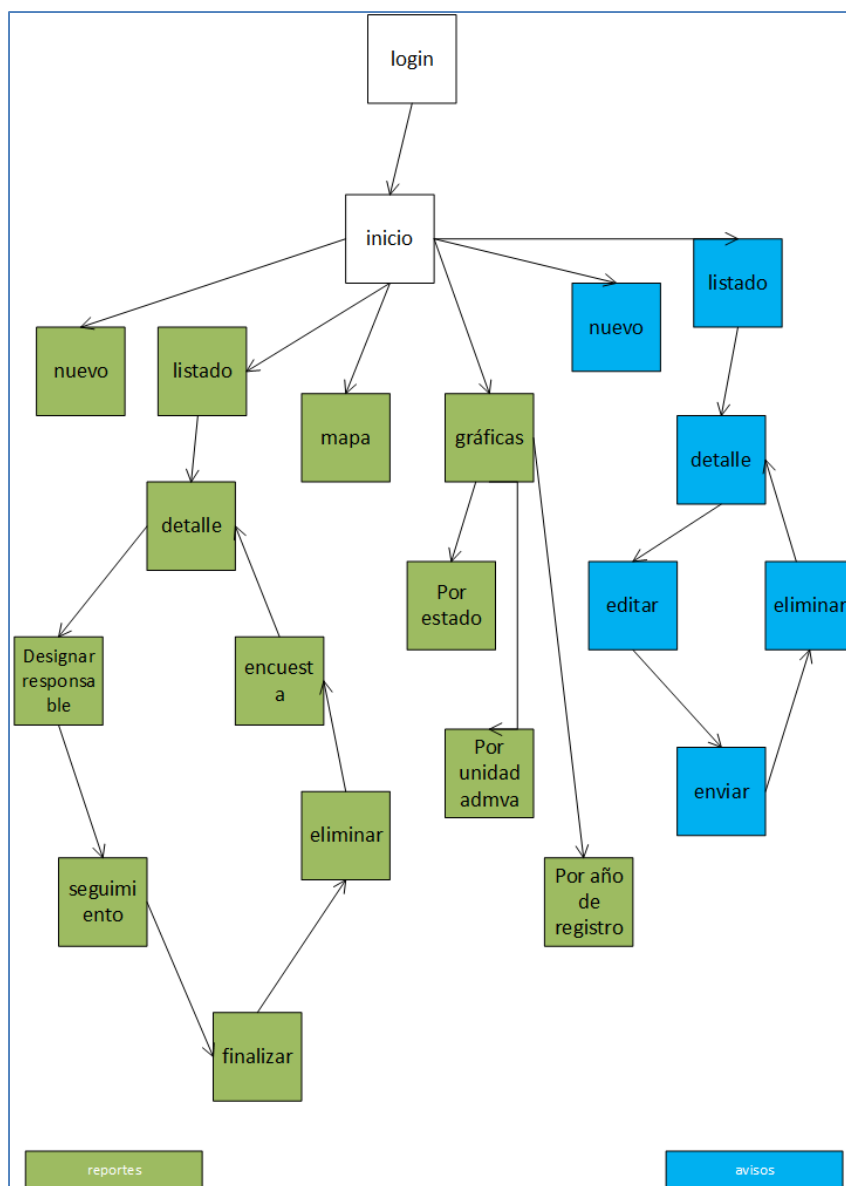


Figura 27: Diagrama de navegación del sistema web [autoría propia].

A continuación se describen las pantallas más importantes del sistema web.



Seguimiento y Atención de Reportes de
Servicios Públicos

nombre de usuario

password

Ingresar

Othón P. Blanco | Quintana Roo

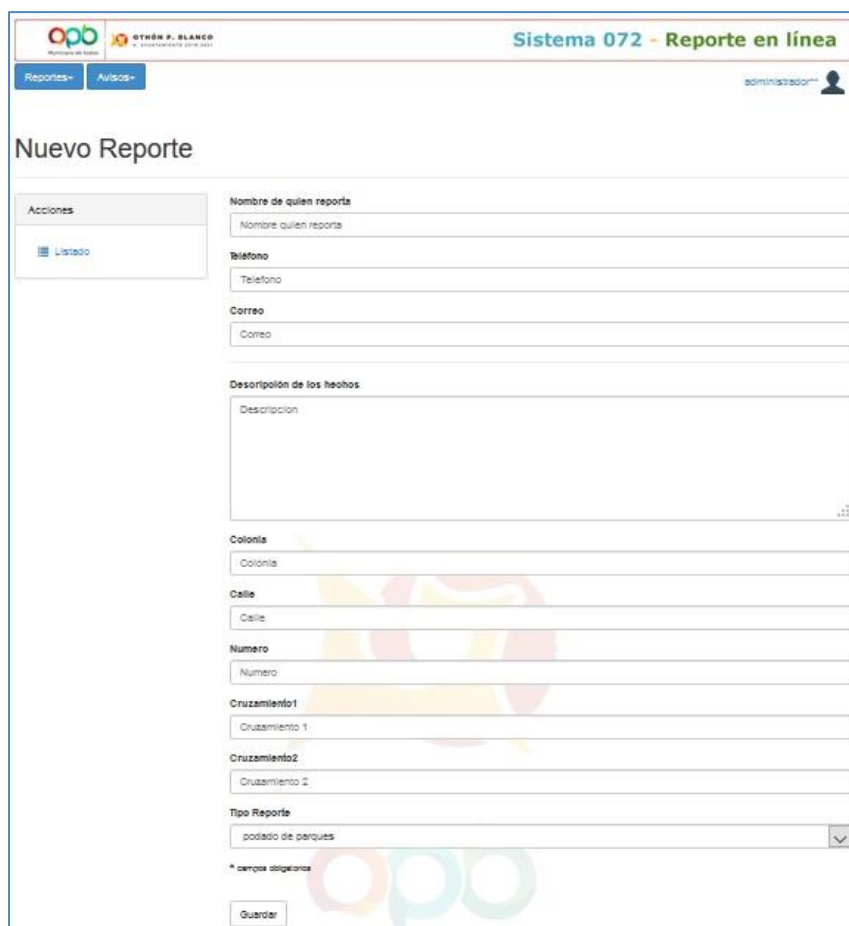
opo

Municipio de todos

Figura 28: Ventana de *logueo* [autoría propia].

En la pantalla anterior el usuario ingresa sus credenciales de acceso para ingresar al sistema.

Para crear un reporte en caso de ser recibido vía llamada telefónica el operador debe ingresar a la opción “Reportes – Nuevo”, entonces se le presenta el siguiente formulario:



The screenshot shows a web application interface for reporting incidents. At the top, there is a header with the 'opo' logo, the text 'GOBIERNO P. BLANCO', and the title 'Sistema 072 - Reporte en línea'. Below the header, there are tabs for 'Reportes' and 'Avisos', and a user profile icon labeled 'administrador'. The main section is titled 'Nuevo Reporte'. On the left, there is a sidebar with 'Acciones' and a 'Listado' button. The main form area contains several input fields: 'Nombre de quien reporta', 'Telefono', 'Correo', 'Descripción de los hechos' (a large text area), 'Colonia', 'Calle', 'Numero', 'Cruzamiento1', 'Cruzamiento2', and 'Tipo Reporte' (a dropdown menu). At the bottom, there is a 'Guardar' button and a note '* campos obligatorios'.

Figura 29: Formulario de captura para un nuevo reporte [autoría propia].

Para dar de alta un nuevo reporte todos los campos son obligatorios.

Al ingresar al detalle del reporte se muestra la siguiente pantalla en donde se muestra toda la información del reporte incluyendo sus seguimientos y solución final:

Detalle

Acciones

- Detalle
- Designar responsable
- Seguimiento
- Finalizar
- Eliminar
- Listado

Datos de contacto

Nombre de quien reporta	angel trejo rojas
Celular	9831149421
Correo	angel.trejo@groo.gob.mx

Reporte

Id reporte	54
Folio reporte	T2909J
Descripción	coladera abierta
Imagen adjunta	T2909J.jpg
Fecha Registro	2019-08-14
Hora Registro	18:11:23
Latitud	18.49743
Longitud	-88.30372166666
Unidad Administrativa responsable de atender	sin asignar responsable de atención
Estado	Seguimiento
Solución	sin solución aún
Creado	2019-08-14 18:11:23
Modificado	2019-08-14 18:31:24

Seguimiento

Descripción	Archivo	Unidad Administrativa	Fecha - Hora
Reporte REGISTRADO.			2019-08-14 18:11:23
Reporte en SEGUIMIENTO			2019-08-14 18:31:24

Figura 30: Detalle del reporte [autoría propia].

La opción “Designar responsable” permite asignar en el sistema un área responsable de dar seguimiento y resolución al reporte, es decir, será la encargada de subir los avances (seguimientos) y solución final del reporte, la siguiente imagen nos muestra su interface:



OTHOÓN P. BLANCO
Municipio de Totolá

Sistema 072 - Reporte en línea

Reportes- Avisos-

administrador**

Designar responsable

Acciones

- Detalle
- Designar responsable
- Seguimiento
- Finalizar
- Eliminar
- Listado

Folio: T2909J
Descripción: coladera abierta
Registro: 2019-08-14 18:11:23

Unidad Administrativa que atiende

DEPARTAMENTO DE ADMINISTRACIÓN E IMAGEN URBANA

Guardar

Figura 31: Designación del área responsable de atención [autoría propia].

Para dar de alta los seguimientos de cada reporte se ingresa a la opción “Seguimiento” en donde el área responsable ingresa de manera obligatoria un texto describiendo el seguimiento realizado y además puede adjuntar un archivo a modo de evidencia. La interface de este formulario es la siguiente:

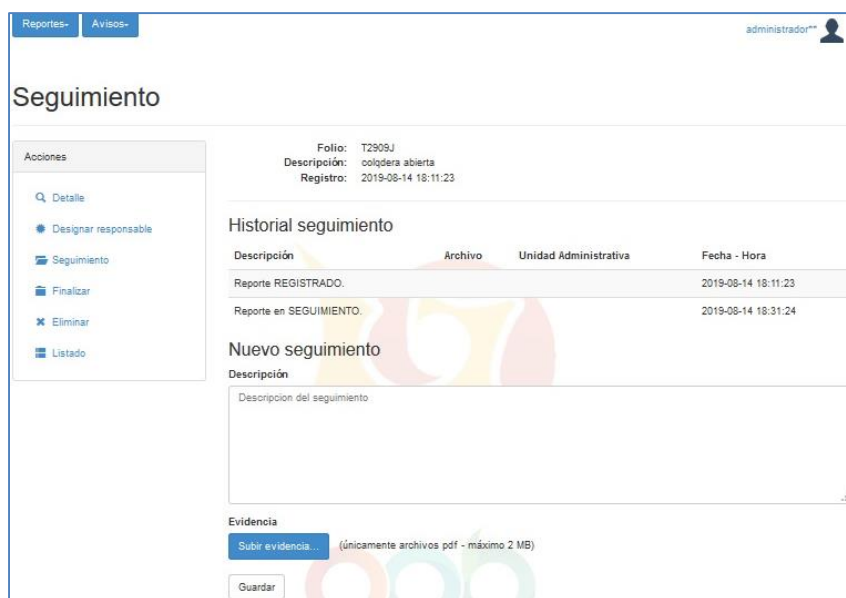


Figura 32: Pantalla de seguimiento [autoría propia].

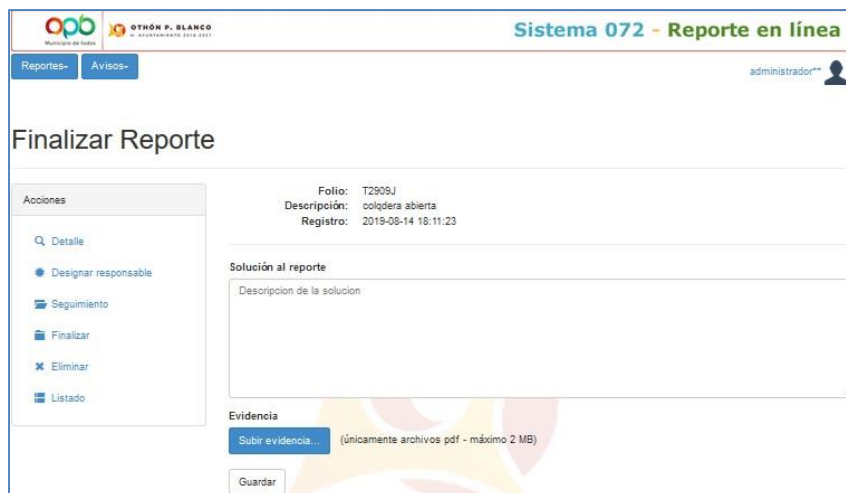


Figura 33: Pantalla de finalización de reporte [autoría propia].

La figura anterior muestra el formulario para ingresar la solución final del reporte, se solicita una descripción de la misma y de manera opcional un archivo adjunto como evidencia.

La opción “Reportes - mapa” nos permite visualizar en un mapa los marcadores y la descripción de los reportes hechos en un año - mes determinado como se muestra a continuación:

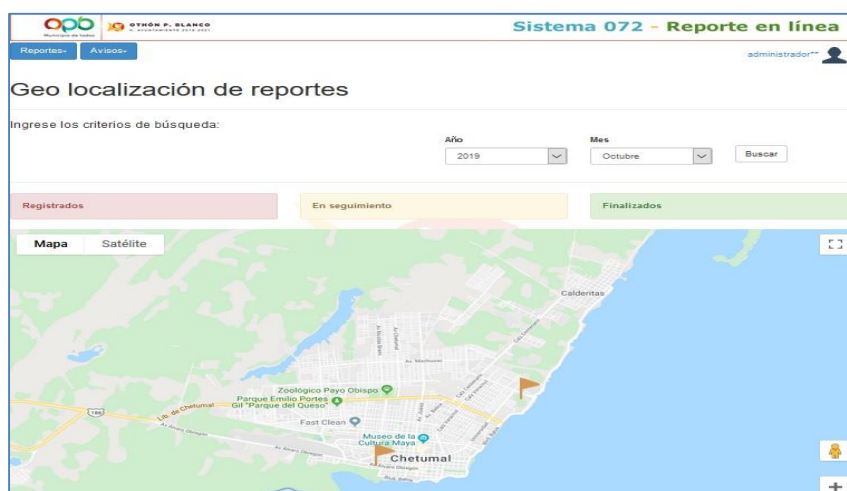


Figura 34: Mapa de reportes registrados [autoría propia].

La opción “Reportes – gráficas” nos permite el acceso a las gráficas proporcionadas por el sistema, todas las gráficas permiten elegir un año y mes específico para mostrar resultados. Las siguientes figuras muestran las gráficas correspondientes:



Figura 35: Gráfica “por estado del reporte” [autoría propia].



Figura 36: Gráfica “por unidad administrativa” [autoría propia].



Figura 37: Gráfica “por año de registro” [autoría propia].

En lo que se refiere a los avisos, se debe ingresar a la opción “Avisos - nuevo” para crear uno nuevo, la descripción es obligatoria y la imagen adjunta (jpg) es opcional. La pantalla de captura es la siguiente:

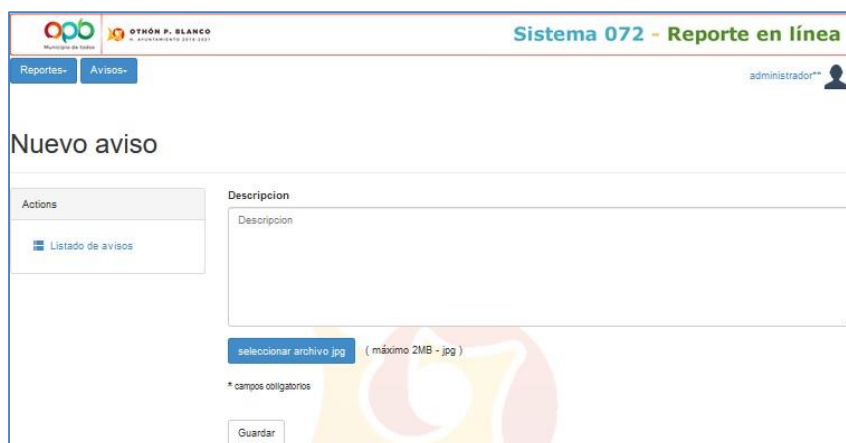


Figura 38: Pantalla para crear un nuevo aviso [autoría propia].

En la opción “editar” se permite la modificación de la descripción y/o imagen que componen a un reporte, esta es la pantalla correspondiente:



Figura 39: Edición de un aviso [autoría propia].

En la opción “enviar” solamente se le pide al usuario dar clic en un botón para que el aviso sea enviado a modo de notificación *push* a todos los dispositivos móviles que tengan la *app* instalada:

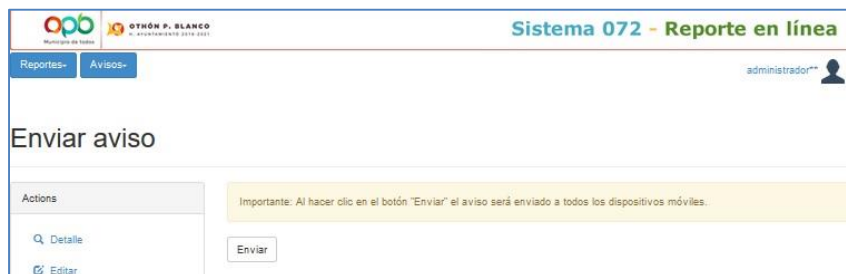


Figura 40: Envío de aviso [autoría propia].

Estas han sido las pantallas más importantes del sistema web de uso exclusivo del personal del municipio, para ver el código fuente de los módulos de este sistema ir al anexo 1.

Objetivo específico 3: “Desarrollar una aplicación móvil para Android basada en el análisis aprobado por el municipio de Othón P. Blanco.”

Para este objetivo el primer resultado es el diagrama de interfaces en donde se aprecia la relación del flujo de navegación entre ellas:

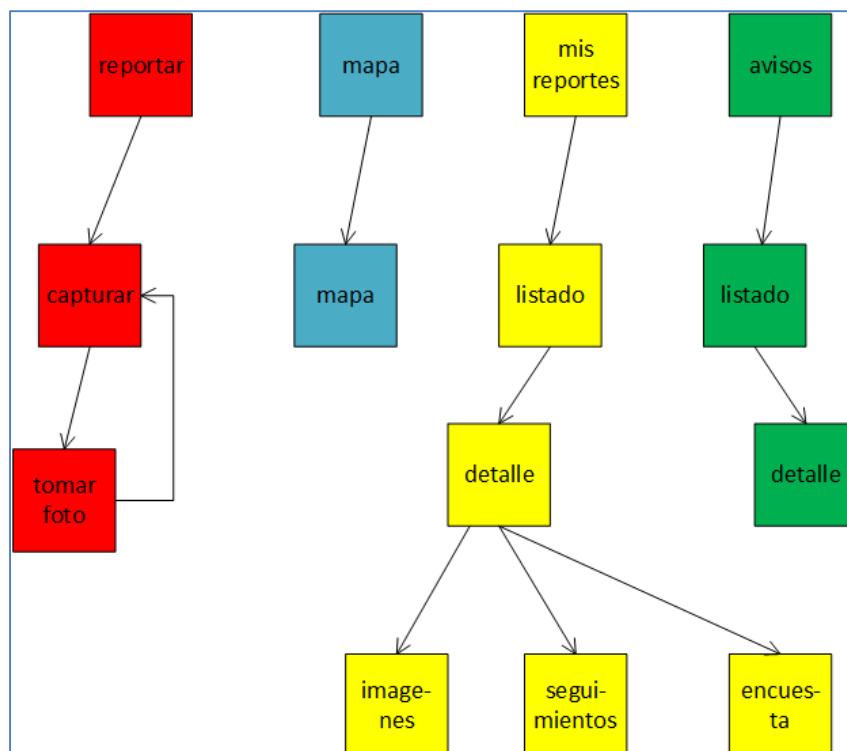


Figura 41: Diagrama de navegación de la *app* [autoría propia].

Al abrir la aplicación, la pantalla principal muestra las 4 principales opciones: reporta, mapa, mis reportes y avisos, como se muestra continuación:



Figura 42: Pantalla principal de la aplicación [autoría propia].

En la opción “reporta”, después de tomar la primera imagen de evidencia se presenta el formulario de captura del reporte en donde el ciudadano puede agregar dos fotos más y proporcionar la información relevante del suceso. Dicho formulario de captura es el siguiente:

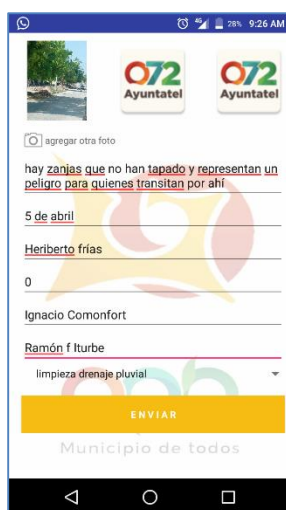


Figura 43: Formulario de captura [autoría propia].

La opción “mapa” muestra la geo localización de los últimos reportes hechos por el ciudadano junto a su descripción:

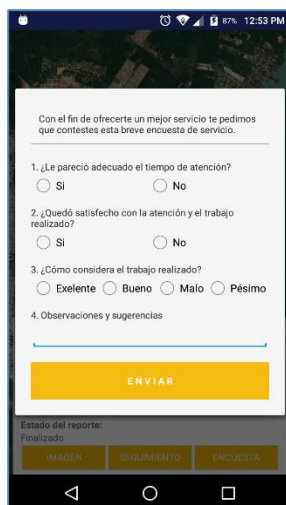


Figura 44: Mapa de reportes realizados [autoría propia].

La opción “mis reportes” muestra un listado de los reportes hechos por el ciudadano, cuando se elige uno de ellos se muestra un mapa con su ubicación y el detalle de información así como tres botones para ver las imágenes relacionadas al reporte, sus seguimientos y la posibilidad de responder una encuesta de calidad una vez el reporte ha sido atendido y concluido:



Figura 45: Detalle del reporte [autoría propia].



Con el fin de ofrecerte un mejor servicio te pedimos que contestes esta breve encuesta de servicio.

1. ¿Le pareció adecuado el tiempo de atención?
☐ SI ☐ No
2. ¿Quedó satisfecho con la atención y el trabajo realizado?
☐ SI ☐ No
3. ¿Cómo considera el trabajo realizado?
☐ Excelente ☐ Bueno ☐ Malo ☐ Pésimo
4. Observaciones y sugerencias

ENVIAR

Estado del reporte:
Finalizado

INAGUO **COMPLETADO** ENCUESTA

Figura 46: Encuesta de calidad del servicio [autoría propia].

En la opción “avisos” se muestran los comunicados que el ayuntamiento ha enviado a los ciudadanos a través de la aplicación. Cuando se elige un aviso del listado se muestra el detalle del mismo como se muestra a continuación:



Figura 47: Detalle del aviso [autoría propia].

Par ver el código fuente de la aplicación móvil ver el anexo 2.

Objetivo específico 4: “Llevar a cabo una evaluación funcional y documentación del sistema de administración vía web y de la aplicación móvil”.

Para el caso del sistema web las pruebas consistieron en evaluar el correcto funcionamiento de sus servicios web (API) que son los que permiten la comunicación con los dispositivos móviles mediante la recepción de toda la información que es enviada desde y hacia la *app*, esta información es, por ejemplo, los reportes que son recibidos desde la *app* así como las notificaciones *push* que son enviadas hacia los dispositivos móviles. De la misma manera, se evaluó el correcto funcionamiento del resto del sistema, permitiendo corroborar que se mantiene la integridad de la información tanto en la base de datos del sistema como en la base de datos de los dispositivos móviles después de ejecutar todos los procesos que abarca este proyecto.

En las siguientes imágenes se muestra a personal del área “072 Ayuntatel” realizando pruebas del sistema web.

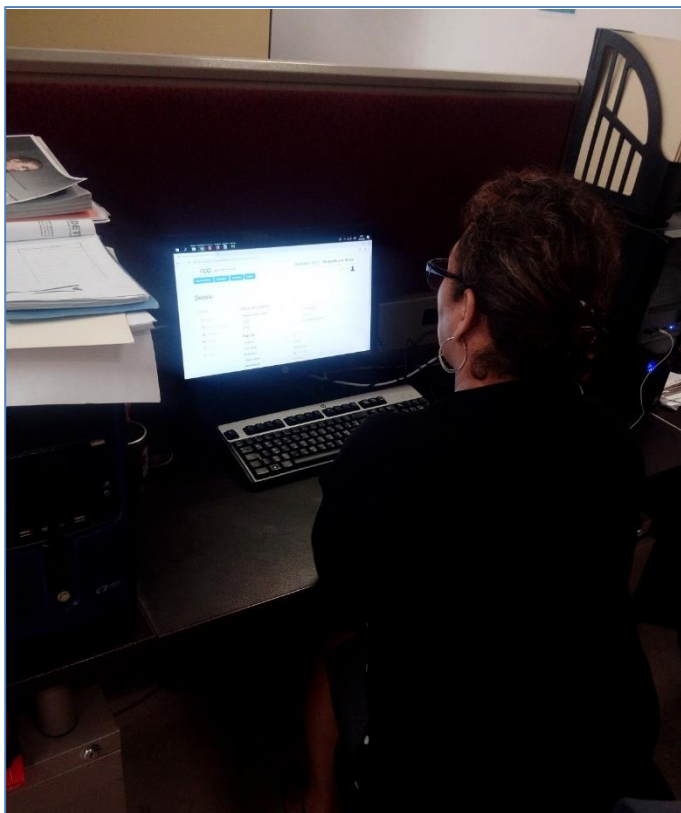


Figura 48: Evaluación del sistema web [autoría propia].

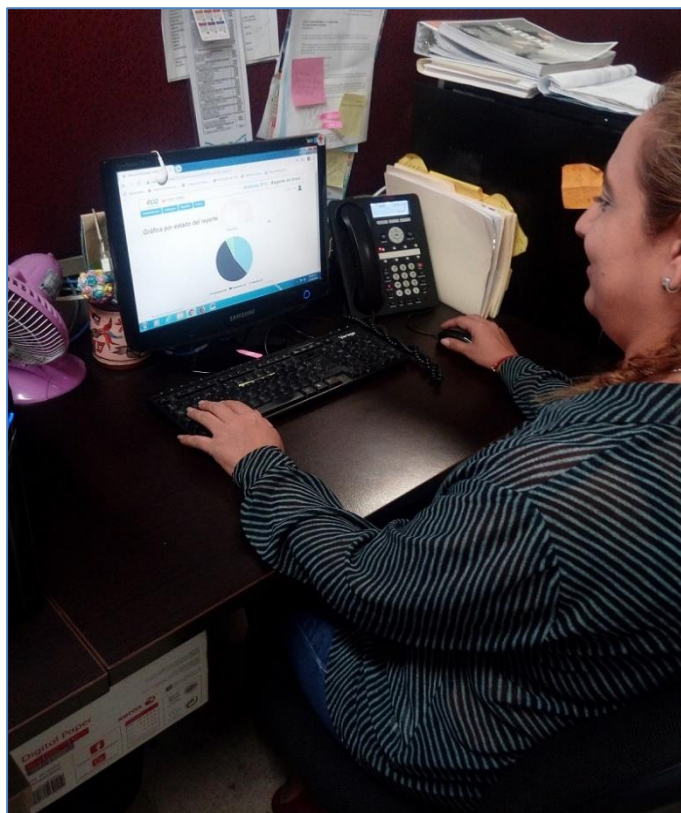


Figura 49: Evaluación del sistema web [autoría propia].

Con respecto a la aplicación móvil se procedió a realizar pruebas de funcionamiento que consistieron en la realización de reportes de prueba con diferentes usuarios en diferentes ubicaciones, de igual forma los dispositivos usados fueron de diferente modelo y diferente versión del sistema operativo Android. El objetivo fue probar la compatibilidad de funcionamiento de la aplicación en dispositivos con diferente versión de Android encontrándose que la aplicación es compatible con versiones de Android desde 4.4 hasta la versión 9. De igual forma se comprobó que la aplicación conserva la integridad de los datos con respecto a los que son enviados al sistema web, también que recibe las notificaciones con éxito y en general no presenta fallas de funcionamiento.

En las siguientes imágenes se muestra a usuarios realizando pruebas con la aplicación móvil:

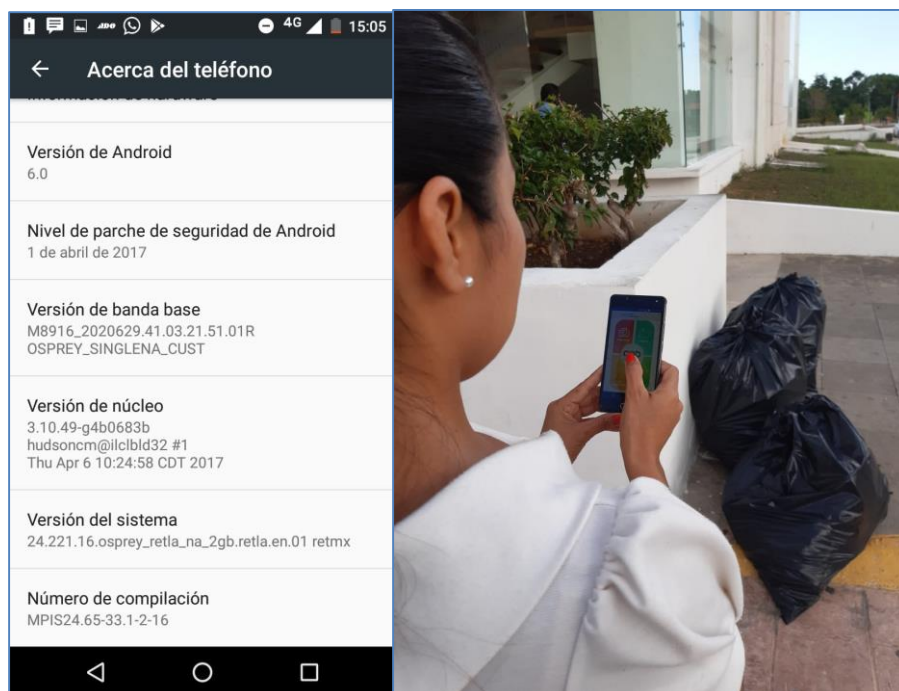


Figura 50: Evaluación de la aplicación móvil [autoría propia].

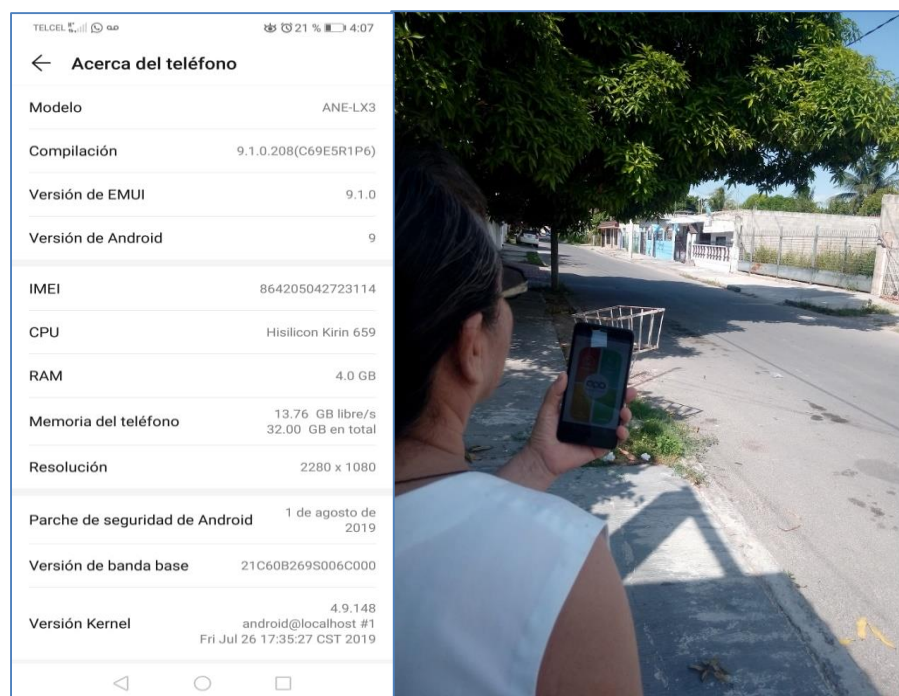


Figura 51: Evaluación de la aplicación móvil [autoría propia].

Como parte de la evaluación el día 22 de Enero de 2019 se presentó ante directores de las distintas áreas del Ayuntamiento este proyecto a fin de obtener sus opiniones con el propósito de mejorarlo. Resultado de esta exposición fue la aprobación de las funcionalidades propuestas y la adición de la funcionalidad de “Avisos”. A continuación se muestran algunas imágenes de dicha reunión.

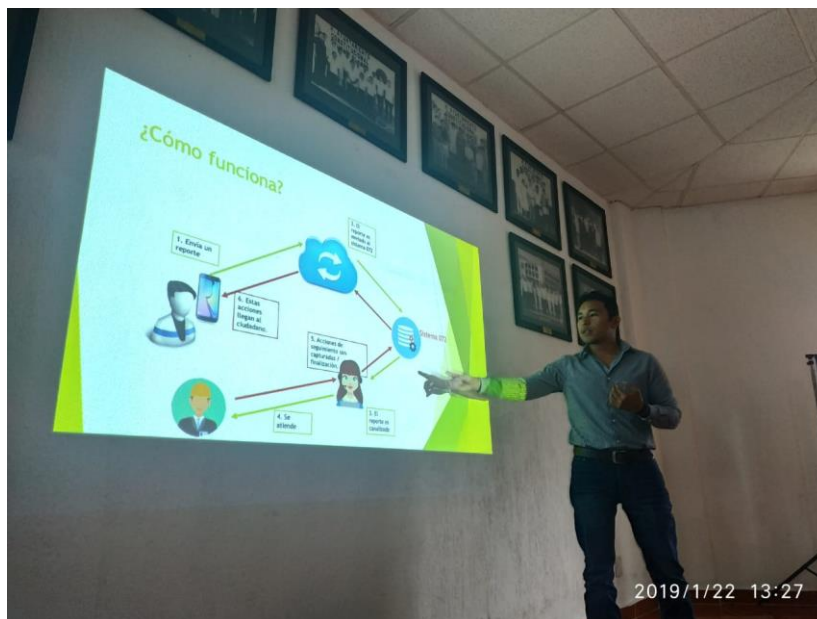


Figura 52: Presentación del proyecto a directores de área [autoría propia].

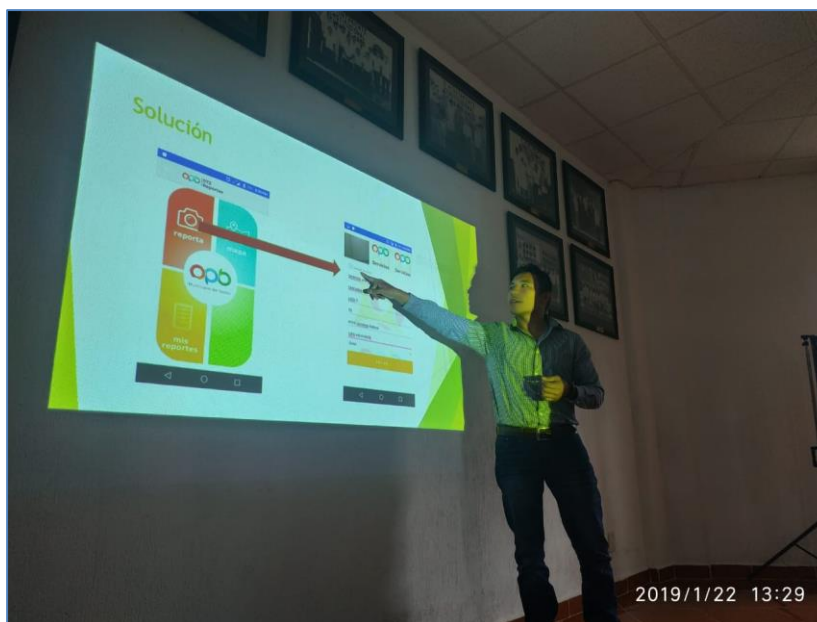


Figura 53: Presentación del proyecto a directores de área [autoría propia].

Después de un proceso de evaluación del proyecto se logró la aprobación del mismo por parte de la oficina de la presidencia, como consecuencia se implementó el uso del sistema web para los trabajadores municipales del área “072 Ayuntatel” y por otro lado se publicó la *app* en la tienda de Google, todo esto a partir del día 17 de Septiembre de 2019. A continuación se muestran las capturas de pantalla del sistema en producción y de la aplicación publicada.

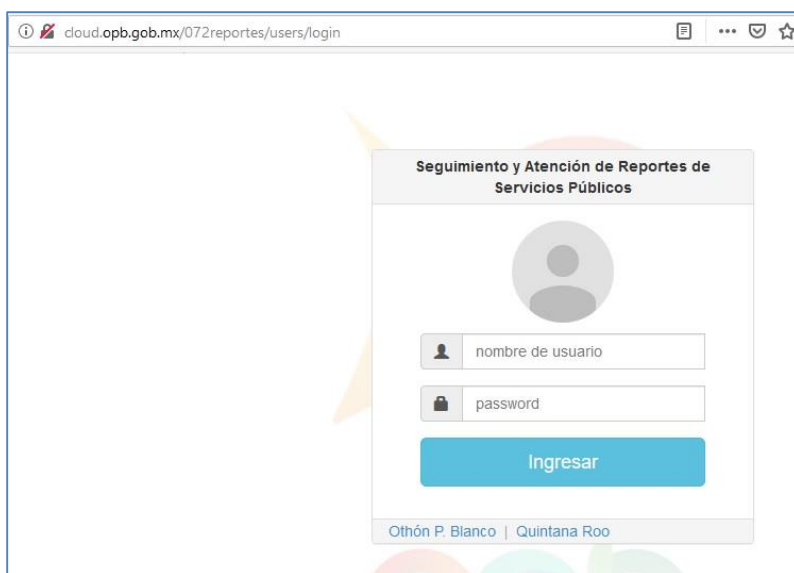


Figura 54: El sistema web en producción [autoría propia].

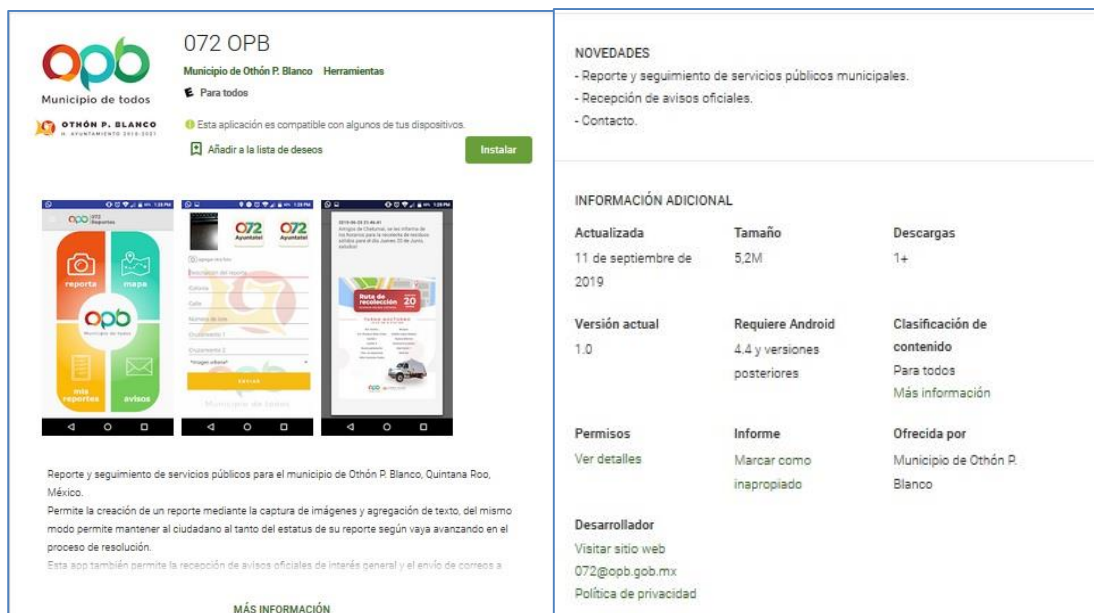


Figura 55: La aplicación móvil publicada [autoría propia].

Capítulo 9: Conclusiones

Se ha desarrollado el proyecto “072 Reportes” el cual consiste de un sistema web para uso exclusivo del personal municipal, así como de una aplicación móvil para uso de la ciudadanía en general, mediante la cual se realiza el levantamiento de reportes sobre deficiencias en los servicios públicos. Este proyecto ha permitido la optimización de los procedimientos de reporte y seguimiento mediante la aplicación e integración de las TIC’s.

Para el óptimo desarrollo de este proyecto se investigó en internet el estado actual de este tipo de proyectos en México, encontrándose que son pocos los ayuntamientos que cuentan con proyectos similares en relación a la gran cantidad de municipios existentes en nuestro país (que cuentan con las condiciones necesarias para el desarrollo de este tipo de proyectos). Es importante destacar que en la región sureste de México solo el ayuntamiento de Mérida (Yucatán) cuenta con un desarrollo de este tipo, siendo así el proyecto “072 Reportes” el segundo en su tipo en esta región.

Para una mejor comprensión de este trabajo por parte del lector se han explicado también los conceptos tecnológicos pertinentes.

Mediante el análisis de los procesos municipales correspondientes a la recepción, seguimiento y conclusión de los reportes ciudadanos se logró determinar los requerimientos de información necesarios, de esta forma se logró el desarrollo del sistema web y de la aplicación móvil. Es importante mencionar que gracias a las evaluaciones realizadas al proyecto se comprobó el cumplimiento de las necesidades de los usuarios, esto es, por parte del sistema web ser fácil de usar y arrojar los reportes y gráficas necesarias para la toma de decisiones por parte de las autoridades municipales; por parte de la aplicación móvil, las evaluaciones sirvieron para tener una *app* compatible con las versiones de Android desde 4.4 (Kit Kat) hasta la versión 9 (Pie) siendo una *app* intuitiva y fácil de usar.

La implementación de este proyecto tiene gran beneficio para ambas partes, por un lado, el gobierno municipal obtiene rapidez en la recepción de reportes hasta el punto de generar un mapa en tiempo real de los mismos, por otra parte, el proceso de seguimiento y

resolución es completamente automatizado reduciendo los tiempos de espera en cuanto a la consulta de información se refiere. La concentración de los datos en este sistema permite la generación de reportes y gráficas integrales que arrojan información exacta y en tiempo real permite una acertada toma de decisiones para quien así necesite hacerlo.

El ciudadano por otra parte se ve beneficiado al permitirle levantar un reporte con gran simplicidad teniendo únicamente que tomar una foto, agregar una breve información complementaria y enviar su reporte, todo desde su teléfono móvil. Todas las acciones de seguimiento y resolución que sean realizadas por el municipio le son enviadas al ciudadano en forma de notificaciones, ganando con esto confiabilidad sobre las instituciones.

Este nuevo canal de comunicación también permite mantener informada a la ciudadanía de manera rápida y directa al permitirles mediante la aplicación móvil recibir avisos y comunicados que el Ayuntamiento crea pertinente (cierre de calles por mantenimiento, limpieza de coladeras, entre otros) en cuanto a los servicios públicos se refiere.

Este proyecto proporciona una gran oportunidad para el gobierno municipal de modernizarse internamente en lo que se refiere a su procedimiento de recepción, seguimiento y solución de reportes de servicios públicos, y de cara al ciudadano le permite ganar una imagen moderna y confiable. Para el ciudadano esta herramienta le permite una participación más activa en el mantenimiento de su comunidad y una comunicación directa con las autoridades.

En este proyecto, tanto el sistema web como la aplicación móvil quedan abiertos a futuras funcionalidades que se requieran ser adicionadas (escalabilidad), esto gracias a la documentación generada durante el diseño y a la tecnología de codificación usada, en el caso del sistema web se usó la metodología modelo – vista – controlador a través del *framework* de desarrollo CakePHP.

Capítulo 10: Trabajo futuro

A continuación se enlistan las tareas que representan una oportunidad de mejora de cara al futuro del proyecto 072 Reportes. De esta manera, las actividades que se consideran necesarias para la continuación y mejoramiento del proyecto son:

- Versión para iOS. La versión de la *app* actualmente es únicamente para el sistema operativo Android, para tener un mayor alcance de ciudadanos beneficiados se necesita desarrollar la versión de la *app* para el sistema operativo iOS de Apple. Para ello será necesario contar con el equipo de cómputo de desarrollo adecuado (apple) y con el conocimiento necesario del lenguaje de programación Objective C o Swift.
- Almacenamiento en local por parte de la *app*. Se requerirá la adecuación de la versión en Android (la versión en iOS deberá tomar en cuenta este punto) para que los datos del reporte, en caso de no haber conexión a internet en el momento de hacer el reporte, sean guardados en el almacenamiento local del dispositivo móvil del ciudadano. De esta forma cuando se cuente con una conexión a internet, el ciudadano podrá enviar su reporte previamente guardado.
- Versión web de la *app*. Para lograr un mayor alcance de usuarios, se deberá desarrollar una versión web de la *app* que sea diseñada en modo responsivo para que se pueda adaptar a diferentes medidas de pantallas. Es importante mencionar que esta versión no podría almacenar datos en modo local a diferencia de las versiones nativas (Android y iOS) para dispositivos móviles.
- Toda aquella funcionalidad que necesite ser agregada al proyecto podrá ser desarrollada debido a que a que el proyecto está diseñado modularmente, esto garantiza su crecimiento y amplitud de beneficios para el ciudadano y/o usuarios del ayuntamiento.

Referencias

- [1]. Ibarra Mares, Alberto. (2010). Introducción a las finanzas públicas. Tomado el 23 de Septiembre de 2017, de Eumed.net. Url: <http://www.eumed.net/libros-gratis/2010a/665/CARACTERISTICAS%20DE%20LOS%20SERVICIOS%20PUBLICOS.htm>
- [2] (2015). Guía de Servicios Públicos Municipales. Tomado el 29 de Septiembre de 2017, de Instituto Nacional para el Federalismo y el Desarrollo Municipal. Url: http://www.adm.gob.mx/work/models/ADM/Resource/210/1/images/Guia_de_servicios_publicos_municipales_2015.pdf
- [3]. Organización de los Estados Americanos. (2010). Sobre e-Gobierno. Tomado el 23 de Septiembre de 2017, de Organización de los Estados Americanos. Url: <http://portal.oas.org/Portal/Sector/SAP/DptodeModernizaci3nDelEstadoyGobernabilidad/NPA/SobreProgramadeeGobierno/tabid/811/Default.aspx>
- [4]. Román Tapia, Bonifacio. Cuadernos de Investigación de Operaciones. Tomado el 25 de Octubre de 2017, de Universidad Autónoma de México. Url: <http://www.ingenieria.unam.mx/industriales/descargas/documentos/catedra/investigacionopboni.pdf>
- [5]. Dávila Silva, Pablo. Software y Hardware. Tomado el 25 de Octubre de 2017, de Universidad Autónoma de México. Url: http://www.paginaspersonales.unam.mx/files/490/SOFTWARE_Y_HARDWARE.pdf
- [6]. (2017). Objetivos de Ingeniería del Software. Tomado el 25 de Octubre de 2017, de Universidad Politécnica de Madrid. Url: <http://www.etsisi.upm.es/estudios/grados/software/objetivos>
- [7]. Baltazar Guzmán, Francisco. (2016). Programación Estructurada. Tomado el 26 de Octubre de 2017, de Universidad Autónoma del Estado de Hidalgo. Url: https://www.uaeh.edu.mx/docencia/P_Presentaciones/huejutla/sistemas/programa_estruc/pro_estructurada.pdf
- [8]. Wikipedia. (2017). Programación Orientada a Objetos. Tomado el 26 de Octubre de 2017, de Fundación Wikimedia. Url: https://es.wikipedia.org/wiki/Programaci3n_orientada_a_objetos

- [9]. Laudon, Kenneth C.; Guercio Travel, Carol. (2014). E-commerce 2013. Negocios, tecnología, sociedad. México: Pearson Educación.

- [10]. Pérez, Felipe U. (2015). Clasificación de los lenguajes de programación. Tomado el 25 de Octubre de 2017, de La Revista Informática. Url: <http://larevistainformatica.com/clasificacion-de-los-lenguajes-de-programacion.html>

- [11]. CakePHP. (2017). CakePHP de un vistazo. Tomado el 25 de Octubre de 2017, de CakePHP. Url: <https://book.cakephp.org/3.0/es/intro.html>

- [12]. Cubillos, Claudio. Arquitectura cliente / servidor. Tomado el 26 de Octubre de 2017, de Pontificia Universidad Católica de Valparaíso. Url: <http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>

- [13]. (2019). World Internet Usage and Population Statistics. Tomado el 06 de Mayo de 2019, de Internet World Stats. Url: <https://www.internetworldstats.com/stats.htm>

- [14]. Meriño Maceira, José Raúl. Pardo Vega, Enrique. (2015). Simulador para el protocolo de comunicación industrial AB Ethernet. Tomado el 04 de Junio de 2019, de Universidad de las Ciencias Informáticas. Url: https://www.researchgate.net/publication/279531120_Simulador_para_el_protocolo_de_comunicacion_industrial_AB_Ethernet

- [15]. Bersvendsen, Jorn. (2011). Control of a multifunction Arm Prosthesis Model. Tomado el 04 de Junio de 2019, de Norwegian University of Science and Technology. Url: https://www.researchgate.net/publication/266185907_Control_of_a_multifunction_Arm_Prosthesis_Model

- [16]. Delgado, Danielle. (2014). Publish – Subscriber. Tomado el 04 de Junio de 2019, de Arquitectura de Software. Url: <http://arquitecturasomos4.blogspot.com/2014/12/publish-subscriber.html>

- [17]. Wikipedia. (2019). Publish – subscribe pattern. Tomado el 05 de Junio de 2019, de Fundación Wikimedia. Url: https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern

- [18]. Kendall, Kenneth E. y Kendall, Julie E. (2011). Análisis y Diseño de Sistemas. México: Pearson Educación. Tomado el 23 de Julio de 2019.
- [19]. Jagtap, Shayaan. (2018). IoT + Machine Learning is Going to Change the World. Tomado el 25 de Julio de 2019, de Towards Data Science. Url: <https://towardsdatascience.com/iot-machine-learning-is-going-to-change-the-world-7c4e0cd7ac32>

Apéndice 1

A continuación se muestra parte del código fuente correspondiente al sistema web destinado al uso exclusivo del personal de municipio. El código fuente mostrado corresponde a algunos de los modelos, funciones de controlador y vistas principales.

```
1 <?php
App::uses('AppModel', 'Model');

class Reporte extends AppModel {
    public $primaryKey = 'reporte_id';
    public $belongsTo = array(
        'User' => array(
            'className' => 'User',
            'foreignKey' => 'usuario_id',
            'conditions' => '',
            'fields' => '',
            'order' => ''
        ),
        'Direcciones' => array(
            'className' => 'Direcciones',
            'foreignKey' => 'direccion_id',
            'conditions' => '',
            'fields' => '',
            'order' => ''
        ),
        'Estado' => array(
            'className' => 'Estado',
            'foreignKey' => 'estado_id',
            'conditions' => '',
            'fields' => '',
            'order' => ''
        )
    );
    public $hasMany = array(
        'Seguimiento' => array(
            'className' => 'Seguimiento',
            'foreignKey' => 'reporte_id',
            'dependent' => false,
            'conditions' => '',
            'fields' => '',
            'order' => '',
            'limit' => '',
            'offset' => '',
            'exclusive' => '',
            'finderQuery' => '',
            'counterQuery' => ''
        )
    );
}
```

Figura 56: Código del modelo de la tabla reportes [autoría propia].

```
public function add() {
    if ($this->request->is('post')) {
        //echo'<pre>';print_r($this->request->data);echo'</pre>';exit;
        $this->request->data['Reporte']['fecha_registro'] = date('Y-m-d');
        $this->request->data['Reporte']['hora_registro'] = date('H:i:s');

        $Repetido = 1;
        while ( $Repetido == 1 ){
            $Folio = $this->GeneraFolio();
            $Q1 = $this->Reporte->find('first',array('fields'=>array('Reporte.reporte_id'),
                'conditions'=>array('Reporte.folio'=>$Folio)));
            if ( empty( $Q1 ) ){
                $Repetido = 0; //el folio no se ha usado y permito salir dle ciclo
            }
        }
        $this->request->data['Reporte']['folio'] = $Folio;
        $this->request->data['Reporte']['estado_id'] = '1'; //registrado
        $this->request->data['Reporte']['origen'] = '2'; //sistema
        $this->request->data['Reporte']['login'] = $this->Auth->user('username');

        $this->Reporte->create();
        if ($this->Reporte->save($this->request->data)) {
            $this->Session->setFlash__( 'El reporte ha sido guardado.', 'default', array('class' => 'alert alert-success'));
            return $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash__( 'Error: El reporte no pudo ser guardado. Intente nuevamente.', 'default', array('class' => 'alert alert-danger'));
        }
    }

    $users = $this->Reporte->User->find('list');
    $direcciones = $this->Reporte->Direccion->find('list');
    $estados = $this->Reporte->Estado->find('list');
    $this->set(compact('users', 'direcciones', 'estados'));
    $this->set('TipoReporte',$this->TipoReporte);
}
```

Figura 57: Función para guardar un nuevo reporte [autoría propia].

```
public function asigna_responsable($IdReporte){
    App::import('Model', 'Direccion');
    $Direccion = new Direccion;

    if (!$this->Reporte->exists($IdReporte)) {
        throw new NotFoundException__( 'Reporte invalido');
    }
    $this->cambia_estado($IdReporte,2);
    if ($this->request->is(array('post', 'put'))) {
        $this->Reporte->id = $this->request->data['Reporte']['IdReporte'];
        $this->Reporte->saveField('direccion_id',$this->request->data['Reporte']['direccion_id']);
        $this->Session->setFlash__( 'Se ha guardado la unidad administrativa responsable de atender el reporte.', 'default', array('class' => 'alert alert-success'));
        return $this->redirect(array('action' => 'index'));
    }

    $DatosEncabezado = $this->DatosReporteEncabezado($IdReporte);
    $DireccionesAdmvas = $Direccion->find('list',array('fields'=>array('Direccion.descripcion'),
        'order'=>array('Direccion.descripcion'=>'ASC')));

    $options = array('conditions' => array('Reporte.' . $this->Reporte->primaryKey => $IdReporte));

    $this->set('reporte', $this->Reporte->find('first', $options));
    $this->set('DatosEncabezado',$DatosEncabezado);
    $this->set('DireccionesAdmvas',$DireccionesAdmvas);
}
```

Figura 58: Función para asignar un área responsable de atención [autoría propia].

```
public function delete($Id = null) {
    if ($this->request->is(array('post', 'put'))) {
        //echo'<pre>';print_r($this->request->data);echo'</pre>';exit;
        $this->Reporte->id = $this->request->data['Reporte']['reporte_id'];
        $this->Reporte->saveField('motivo Eliminacion',$this->request->data['Reporte']['motivo Eliminacion']);
        $this->Reporte->saveField('estado_id,$); //eliminado
        $this->Session->setFlash__( 'El reporte ha sido eliminado.', 'default', array('class' => 'alert alert-success'));
        return $this->redirect(array('action' => 'index'));
    }

    $DatosEncabezado = $this->DatosReporteEncabezado($Id);

    $this->set('IdReporte',$Id);
    $this->set('DatosEncabezado',$DatosEncabezado);
}
```

Figura 59: Función para eliminar un reporte [autoría propia].

```

public function seguimiento($IdReporte){
    if (!isset($IdReporte) || $IdReporte == '') {
        throw new NotFoundException(__('Reporte invalido'));
    }
    $this->cambia_estado($IdReporte,2);
    if ($this->request->is(array('post', 'put'))) {
        $IdReporte = $this->request->data['Reporte']['id'];
        $Descripcion = $this->request->data['Reporte']['descripcion'];
        $IdUsuario = $this->Auth->user('id');
        $IdDireccion = $this->Auth->user('direccion_id');
        $BandInsertar = 0; //no permite insertar por default
        $TmpFileName = ''; //nombre de archivo subido
        $RutaArchivosSeguimiento = Configure::read('RutaArchivosSeguimiento');

        if (isset($this->request->data['Reporte']['archivo']['name']) and ($this->request->data['Reporte']['archivo']['size'] == '0')) {
            $BandInsertar = 1; //permite insertar
        } else {
            if ($this->request->data['Reporte']['archivo']['size'] == '0') { //el adjunto supera el tamaño de upload permitido por el server
                $this->Session->setFlash(__('El tamaño del archivo permitido es de 2 MB, suba un archivo más pequeño'));
                return $this->redirect(array('action' => 'index'));
            } else {
                $TamFile = $this->request->data['Reporte']['archivo']['size'] / 1024;
                if ($TamFile < 2048) {
                    if ($this->request->data['Reporte']['archivo']['type'] == "application/pdf") {
                        move_uploaded_file($this->request->data['Reporte']['archivo']['tmp_name'], $RutaArchivosSeguimiento.$this->request->data['Reporte']['id'].'.pdf');
                        $TmpFileName = $this->request->data['Reporte']['archivo']['name'];
                        $BandInsertar = 1; //permite insertar
                    } else {
                        $this->Session->setFlash(__('Solo se aceptan archivos pdf. El registro no fue guardado.'), 'default', array('class' => 'alert alert-danger'));
                        return $this->redirect(array('action' => 'index'));
                    }
                } else {
                    $this->Session->setFlash(__('El archivo adjunto excede el tamaño permitido de 2 MB. El registro no fue guardado.'), 'default', array('class' => 'alert alert-danger'));
                    return $this->redirect(array('action' => 'index'));
                }
            }
        }
    }
}

```

Figura 60: Parte de la función para guardar un seguimiento [autoría propia].

```

public function notificacion_finalizacion($IdReporte, $IdEstado){
    $Q1 = $this->Reporte->Estado->find('first', array('fields' => array('Estado.descripcion'),
        'conditions' => array('Estado.estado_id' => $IdEstado)));
    $DescripEstado = $Q1['Estado']['descripcion'];
    $Q1 = $this->Reporte->find('first', array('fields' => array('Reporte.usuario_id', 'Reporte.folio', 'Reporte.solucion', 'Reporte.archivo_solucion'),
        'conditions' => array('Reporte.reporte_id' => $IdReporte)));
    $IdUsuario = $Q1['Reporte']['usuario_id'];
    $Folio = $Q1['Reporte']['folio'];
    $Solucion = $Q1['Reporte']['solucion'];
    $ArchivoSolucion = $Q1['Reporte']['archivo_solucion'];
    $Q1 = $this->Reporte->User->find('first', array('fields' => array('User.token_fcm'),
        'conditions' => array('User.id' => $IdUsuario)));
    $TokenFCM = $Q1['User']['token_fcm'];
    $url = $this->Url_FCM;
    $fields = array (
        'registration_ids' => array (
            $TokenFCM
        ),
        'data' => array (
            "BandNotificacion" => '3', //finalizacion del reporte
            "IdReporte" => $IdReporte,
            "Folio" => $Folio,
            "Solucion" => $Solucion,
            "ArchivoSolucion" => $ArchivoSolucion,
            "DescripEstado" => $DescripEstado,
            "IdEstado" => $IdEstado
        )
    );
    $fields = json_encode ( $fields );
    $headers = array (
        'Authorization: key=' . "AAAAy0zHWUg:APA91bG4azS8R2cCvilt5cypbHrj_phJ3mp_d6NmNEMW_g4pG7sB0rhQXTf8gWJ-hy38xxHEdYXsE_Cm25hFnM48bTy9QFCxo8mY",
        'Content-Type: application/json'
    );
    $ch = curl_init ();
    curl_setopt ( $ch, CURLOPT_URL, $url );
    curl_setopt ( $ch, CURLOPT_POST, true );
    curl_setopt ( $ch, CURLOPT_HTTPHEADER, $headers );
    curl_setopt ( $ch, CURLOPT_RETURNTRANSFER, true );
}

```

Figura 61: Parte de la función para enviar una notificación de finalización de reporte [autoría propia].

```
public function add_aviso(){
    App::import('Model', 'Aviso');
    $Aviso = new Aviso;
    if ($this->request->is('post')) {
        $RutaArchivosAvisos = Configure::read('RutaArchivosAvisos');
        $BandInsertar = 0; //por default evita el insert
        $FechaHoy = date('Y-m-d');
        $HoraActual = date('H:i:s');
        $NombreNuevoArchivo = '';
        if ($this->request->data['Reporte']['archivo']['name'] != "") { //hay archivo adjunto
            $TamFile = $this->request->data['Reporte']['archivo']['size'] / 1024;
            if ($TamFile < 2000) {
                if ($this->request->data['Reporte']['archivo']['type'] == "image/jpeg") {
                    if ($this->request->data['Reporte']['archivo']['error'] == 0 && $this->request->data['Reporte']['archivo']['size'] > 0) { //se
                        $NombreNuevoArchivo = $FechaHoy.'_'.$HoraActual.'.jpg';
                        $NombreNuevoArchivo = str_replace(':', '-', $NombreNuevoArchivo);
                        move_uploaded_file($this->request->data['Reporte']['archivo']['tmp_name'], $RutaArchivosAvisos.$NombreNuevoArchivo);
                        $BandInsertar = 1; //permitimos la insercion en la tabla del contenido del archivo
                    } else {
                        $this->Session->setFlash(__('Error: Ocurrió un error al subir el archivo, intente nuevamente.'), 'default', array('class' => 'alert alert-danger'));
                        return $this->redirect(array('action' => 'add_aviso'));
                    }
                } else {
                    $this->Session->setFlash(__('Error: Se aceptan archivos adjuntos del tipo jpg. El archivo seleccionado no es de este tipo.'), 'default', array('class' => 'alert alert-danger'));
                    return $this->redirect(array('action' => 'add_aviso'));
                }
            } else {
                $this->Session->setFlash(__('Error: El archivo excede el tamaño máximo que es de 2 MB. Intente con otro archivo.'), 'default', array('class' => 'alert alert-danger'));
                return $this->redirect(array('action' => 'add_aviso'));
            }
        } else { //no hubo imagen adjunta
            $BandInsertar = 1;
        }
        if ($BandInsertar == 1) { //insertamos en tabla
            $Data = array();
            $Data['Aviso']['descripcion'] = $this->request->data['Reporte']['descripcion'];
            $Data['Aviso']['imagen'] = $NombreNuevoArchivo;
        }
    }
}
```

Figura 62: Parte de la función para crear un nuevo aviso [autoría propia].

```
public function delete_aviso($IdAviso) {
    App::import('Model', 'Aviso');
    $Aviso = new Aviso;
    if ($this->request->is('post')) {
        $Aviso->id = $this->request->data['Reporte']['aviso_id'];
        $Aviso->saveField('motivo_eliminacion', $this->request->data['Reporte']['descripcion']);
        $Aviso->saveField('estado', '3'); //eliminado
        $this->Session->setFlash(__('El aviso ha sido eliminado.'), 'default', array('class' => 'alert alert-success'));
        return $this->redirect(array('action' => 'index_aviso', $IdAviso));
    }
    $this->set('IdAviso', $IdAviso);
}
```

Figura 63: Función para eliminar un aviso [autoría propia].

```
public function m_registra_encuesta() { //registra los resultados de la encuesta de calidad proporcionados por el ciudadano a repres con
    $ArrayTemp = array();
    $ArrayFinal = array();
    $outputArr = array();
    $IdReporte = $_POST['IdReporte'];
    $Resp1 = $_POST['Resp1'];
    $Resp2 = $_POST['Resp2'];
    $Resp3 = $_POST['Resp3'];
    $Comentarios = utf8_encode($_POST['Comentarios']);
    $this->Reporte->id = $IdReporte;
    $this->Reporte->saveField('enc_resp1', $Resp1);
    $this->Reporte->saveField('enc_resp2', $Resp2);
    $this->Reporte->saveField('enc_resp3', $Resp3);
    $this->Reporte->saveField('enc_comentarios', $Comentarios);
    $ArrayTemp['Bandera'] = '1';
    $ArrayFinal[] = $ArrayTemp;
    $outputArr['Datos'] = $ArrayFinal;
    print_r(json_encode($outputArr));
    exit;
}
```

Figura 64: Función para registrar los resultados de la encuesta de calidad del servicio [autoría propia].

```
<style type="text/css">
    #mapa { height: 600px; }
</style>
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCxs69bkkf3DeVFSBUFoEgAcutMTWS11k">
</script>
<script type="text/javascript">
    function initialize() {
        var marcadores = [
            <?php
                foreach ( $Reportes as $Reporte){
                    $Txt = $Reporte['Reporte']['descripcion']. "<br />";
                    $Txt .= $Reporte['Reporte']['fecha_registro'];
                    echo["'".$Txt."', ".$Reporte['Reporte']['latitud'].", ".$Reporte['Reporte']['longitud'].", ".$Reporte['Reporte']['fecha_registro'].", ".$Reporte['Reporte']['estado'].", ".$Reporte['Reporte']['tipo_reporte'].', '];
                }
            <?php
        ];
        var map = new google.maps.Map(document.getElementById('mapa'), {
            zoom: 13,
            center: new google.maps.LatLng(18.526032, -88.292984),
            mapTypeId: google.maps.MapTypeId.ROADMAP
        });
        var infowindow = new google.maps.InfoWindow();
        var marker, i, ruta_marker;
        for (i = 0; i < marcadores.length; i++) {
            if ( marcadores[i][3] == 1 ) { //registrado
                ruta_marker = "<?php echo $Dominio.\"072reportes/app/webroot/img/markers/marker_registrado.png\"; ?>";
            }
            if ( marcadores[i][3] == 2 ) { //seguimiento
                ruta_marker = "<?php echo $Dominio.\"072reportes/app/webroot/img/markers/marker_seguimiento.png\"; ?>";
            }
            if ( marcadores[i][3] == 3 ) { //finalizado
                ruta_marker = "<?php echo $Dominio.\"072reportes/app/webroot/img/markers/marker_finalizado.png\"; ?>";
            }
            marker = new google.maps.Marker({
                position: new google.maps.LatLng(marcadores[i][1], marcadores[i][2]),
            });
            map.add(marker);
            infowindow.setContent(marcadores[i][0]);
            infowindow.open(map, marker);
        }
    }
    initialize();
</script>
</body>
</html>
```

Figura 65: Parte del código de la vista para mostrar el mapa de reportes realizados [autoría propia].

```
<ul class="nav nav-pills nav-stacked">  
    <li><?php echo $this->Html->link(__('<span class="glyphicon glyphicon-search"></span>\n\n';  
    <li><?php echo $this->Html->link(__('<span class="glyphicon glyphicon-edit"></span>\n\n'; Editar  
    <li><?php echo $this->Html->link(__('<span class="glyphicon glyphicon-send"></span>\n\n'; Enviar  
    <li><?php echo $this->Html->link(__('<span class="glyphicon glyphicon-remove"></span>\n\n'; Eliminar  
    <li><?php echo $this->Html->link(__('<span class="glyphicon glyphicon-list"></span>\n\n'; Listado  
</ul>  
</div>  
</div>  
</div><!-- end col md 3 -->  
<div class="col-md-9">  
    <?php echo $this->Form->create('Reporte', array('role' => 'form', 'type'=>'file'));  
        echo $this->Form->hidden('aviso_id',array('value'=>$IdAviso)); ?>  
        <div class="alert alert-warning">  
            Importante! Al hacer clic en el bot&oacute;n "Enviar" el aviso ser&aacute; enviado a todos los dispositivos m&oacute;biles.  
        </div>  
        <div class="form-group">  
            <?php echo $this->Form->submit(__('Enviar'), array('class' => 'btn btn-default', 'id'=>'BtnSubmit', 'onclick'=>'enviar();'))  
        </div>  
    <?php echo $this->Form->end() ?>  
</div><!-- end col md 12 -->  
</div><!-- end row -->  
</div>  
  
div class="modal fade id="myModal" role="dialog">  
    <div class="modal-dialog">  
        <div class="modal-content">  
            <div class="modal-header bg-warning">  
                <button type="button" class="close" data-dismiss="modal" onclick="FNSubmit()">&times;</button>  
                <h4 class="modal-title"><i class="glyphicon glyphicon-warning-sign"></i> \n Confirmaci&oacute;n de evento</h4>  
            </div>  
            <div class="modal-body">  
                <h3>&iquest;Enviar aviso?</h3>
```

Figura 66: Parte del código de la vista para enviar un aviso a los dispositivos móviles [autoría propia].

Apéndice 2

A continuación se muestra parte del código fuente correspondiente a la aplicación móvil destinada al uso por parte de los ciudadanos. El código fuente mostrado corresponde a las principales *activities* de dicha aplicación.

```
public void FEnviar( View view ){
    boolean cancel = false;
    View focusView = null;

    TV_Descripcion = (TextView) findViewById(R.id.descripcion);
    Descripcion = TV_Descripcion.getText().toString();
    TV_Colonia = (TextView) findViewById(R.id.colonia);
    Colonia = TV_Colonia.getText().toString();
    TV_Calle = (TextView) findViewById(R.id.calle);
    Calle = TV_Calle.getText().toString();
    TV_Numero = (TextView) findViewById(R.id.numero);
    Numero = TV_Numero.getText().toString();
    TV_Cruzamiento1 = (TextView) findViewById(R.id.cruzamiento1);
    Cruzamiento1 = TV_Cruzamiento1.getText().toString();
    TV_Cruzamiento2 = (TextView) findViewById(R.id.cruzamiento2);
    Cruzamiento2 = TV_Cruzamiento2.getText().toString();

    if (TextUtils.isEmpty(Descripcion)){
        TV_Descripcion.setError("Porfavor ingrese una descripción para el reporte");
        focusView = TV_Descripcion;
        cancel = true;
    } else if (TextUtils.isEmpty(Colonia)){
        TV_Colonia.setError("Porfavor ingrese la colonia donde sucedieron los hechos");
        focusView = TV_Colonia;
        cancel = true;
    } else if (TextUtils.isEmpty(Calle)) {
        TV_Calle.setError("Porfavor ingrese la calle donde sucedieron los hechos");
        focusView = TV_Calle;
        cancel = true;
    } else if (TextUtils.isEmpty(Numero)) {
        TV_Numero.setError("Porfavor ingrese el número de lote donde sucedieron los hechos");
        focusView = TV_Numero;
        cancel = true;
    } else if (TextUtils.isEmpty(Cruzamiento1)){
        TV_Cruzamiento1.setError("Porfavor ingrese el cruzamiento 1");
        focusView = TV_Cruzamiento1;
        cancel = true;
    } else if (TextUtils.isEmpty(Cruzamiento2)){
        TV_Cruzamiento2.setError("Porfavor ingrese el cruzamiento 2");
        focusView = TV_Cruzamiento2;
        cancel = true;
    }
```

Figura 67: Parte del código para enviar los datos del reporte al servidor [autoría propia].

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.serviciosoph.serviciosoph.CapturaReporte"
    android:background="@mipmap/fondo_app" >

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="368dp"
            android:layout_height="495dp"
            android:orientation="vertical"
            tools:layout_editor_absoluteX="8dp"
            tools:layout_editor_absoluteY="8dp">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="horizontal"
                android:paddingTop="10dp">

                <ImageView
                    android:id="@+id/imageView1"
                    android:layout_width="120dp"
                    android:layout_height="102dp"
                    android:layout_gravity="center"
                    android:layout_weight="1"
                    app:srcCompat="@mipmap/ic_launcher" />

                <ImageView
```

Figura 68: Parte del código para generar el *layout* para la captura del reporte [autoría propia].

```
public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);
        isGPSEnabled = locationManager
            .isProviderEnabled(LocationManager.GPS_PROVIDER);
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
            // No network provider is enabled
        } else {
            this.canGetLocation = true;
            if (isNetworkEnabled) {
                int requestPermissionsCode = 50;
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, mLocationListener);
                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }

        if (isGPSEnabled) {
            if (location == null) {
                if (ContextCompat.checkSelfPermission(this,
                    android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
                    && ActivityCompat.checkSelfPermission(this,
                    android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(activity,
                        new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 50);
                } else {

```

Figura 69: Parte del código para recuperar las coordenadas del ciudadano [autoría propia].

```

        catch(Exception ex) {}
    }
    return null;
}

protected void onPostExecute(Void unused) {
    Dialog.dismiss();
    if (Error != null) {
        Dialog.setMessage(Error);
        //Dialog.show();
    } else {
        String OutputData = "";
        JSONObject jsonResponse;
        try {
            jsonResponse = new JSONObject(Content);
            JSONArray jsonMainNode = jsonResponse.optJSONArray("Datos");
            int lengthJsonArr = jsonMainNode.length();
            for(int i=0; i < lengthJsonArr; i++){
                JSONObject jsonChildNode = jsonMainNode.getJSONObject(i);
                reporte_id = jsonChildNode.optString("reporte_id").toString();
                descripcion = jsonChildNode.optString("descripcion").toString();
                folio = jsonChildNode.optString("folio").toString();
                imagen = jsonChildNode.optString("imagen").toString();
                fecha_registro = jsonChildNode.optString("fecha_registro").toString();
                latitud = jsonChildNode.optString("latitud").toString();
                longitud = jsonChildNode.optString("longitud").toString();
                estado_id = jsonChildNode.optString("estado_id").toString();
                descrip_estado = jsonChildNode.optString("descrip_estado").toString();
                LatLng Chetumal = new LatLng(Double.parseDouble(latitud), Double.parseDouble(longitud));
                MarkerOptions markerOptions = new MarkerOptions();
                markerOptions.position(Chetumal)
                    .title(folio)
                    .snippet(descripcion);
                InfoWindowData info = new InfoWindowData();
                info.setEstado("Estado: " + descrip_estado);
                info.setFechaRegistro("Fecha de registro: " + fecha_registro);
                Marker m = mMap.addMarker(markerOptions);
                m.setTag(info);
                m.showInfoWindow();
            }
        }
    }
}

```

Figura 70: Parte del código para recuperar los reportes y mostrarlos en un mapa [autoría propia].

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:map="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context="com.serviciosopb.serviciosopb.Mapa" />
    <LinearLayout
        android:id="@+id/validity"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:layout_gravity="bottom"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:background="@color/common_google_signin_btn_text_light_default"
        android:orientation="vertical">
        <TextView
            android:id="@+id/textView7"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_vertical|center_horizontal"
            android:text="Reportes realizados en el mes actual"
            android:textColor="@android:color/background_light" />
    </LinearLayout>
</FrameLayout>

```

Figura 71: Código para mostrar el *layout* del mapa de reportes [autoría propia].

```
public void onMapReady(GoogleMap googleMap) {
    String Latitud = null, Longitud = null, Descripcion = null, FechaRegistro = null, HoraRegistro = null, Folio = null, Unida
    Bundle bundle = getIntent().getExtras();
    IdReporte = bundle.getString("IdReporte");

    SQLiteDatabase DBSOPB = DBServiciosOPB.getReadableDatabase();
    Cursor Cl = DBSOPB.rawQuery("SELECT descripcion, latitud, longitud, fecha_registro, hora_registro, folio, direccion_atiend
    if (Cl.moveToFirst()) {
        Descripcion = Cl.getString(0);
        Latitud = Cl.getString(1);
        Longitud = Cl.getString(2);
        FechaRegistro = Cl.getString(3);
        HoraRegistro = Cl.getString(4);
        Folio = Cl.getString(5);
        UnidadAdmva = Cl.getString(6);
        DescripEstado = Cl.getString(7);
        EstadoRep = Cl.getString(8);
    }
    DBSOPB.close(); //cerramos conexion con la db
    TV_Folio = (TextView) findViewById(R.id.TV_Folio);
    TV_FechaRegistro = (TextView) findViewById(R.id.TV_FechaRegistro);
    TV_Descripcion = (TextView) findViewById(R.id.TV_Descripcion);
    TV_UnidadAdmva = (TextView) findViewById(R.id.TV_UnidadAdmva);
    TV_Estado = (TextView) findViewById(R.id.TV_Estado);

    TV_Folio.setText(Folio);
    TV_FechaRegistro.setText(FechaRegistro);
    TV_Descripcion.setText(Descripcion);
    TV_UnidadAdmva.setText(UnidadAdmva);
    TV_Estado.setText(DescripEstado);
    mMap = googleMap;
    LatLng sydney = new LatLng(Double.parseDouble(Latitud), Double.parseDouble(Longitud));
    mMap.addMarker(new MarkerOptions().position(sydney).title(Folio));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 13));
    mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    if (EstadoRep.equals("3")) {
        BtnEncuesta.setEnabled(true); //habilitamos btn para captura de encuesta
        BtnEncuesta.setBackgroundColor( getResources().getColor(R.color.Morena) );
    }
}
```

Figura 72: Parte del código para mostrar el detalle del reporte [autoría propia].

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="false"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true" />
    <LinearLayout
        android:id="@+id/validity"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_gravity="bottom"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:background="@android:color/white"
        android:orientation="vertical">
        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:paddingBottom="5dp"
                android:paddingLeft="10dp"
                android:paddingRight="10dp"
                android:paddingTop="5dp">
                <TextView
```

Figura 73: Parte del código para mostrar el *layout* del detalle del reporte [autoría propia].

```
package com.serviciosopb.serviciosopb;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class BaseDatos extends SQLiteOpenHelper {

    public BaseDatos(Context contexto, String nombre, SQLiteDatabase.CursorFactory factory, int version) {
        super(contexto, nombre, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String sqlCreate = "CREATE TABLE reportes (reporte_id INTEGER, descripcion TEXT, colonia TEXT, calle TEXT, numero TEXT, cruzamiento TEXT, cr";
        db.execSQL(sqlCreate);
        String sqlCreate2 = "CREATE TABLE seguimientos (seguimiento_id INTEGER, descripcion TEXT, archivo TEXT, direccion TEXT, reporte_id INTEGER, cr";
        db.execSQL(sqlCreate2);
        String sqlCreate3 = "CREATE TABLE avisos (aviso_id INTEGER, descripcion TEXT, imagen TEXT, estado INTEGER, created TEXT, modified TEXT)";
        db.execSQL(sqlCreate3);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int versionAnterior, int versionNueva) {
        //para uso de este metodo ver la app de questassist
    }
}
```

Figura 74: Definición de la estructura de la base de datos en el dispositivo móvil [autoría propia].

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.serviciosopb.serviciosopb">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.hardware.location.gps" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-feature
        android:name="android.hardware.camera"
        android:required="false" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".EncuestaReporte"
            android:label=""
            android:theme="@android:style/Theme.Holo.Light.Dialog.NoActionBar" />
        <activity
            android:name=".DetalleAviso"
            android:label=""
            android:theme="@android:style/Theme.Holo.Light.Dialog.NoActionBar" />
        <activity
            android:name=".ListadoAvisos"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar2" />
        <activity
```

Figura 75: Parte del archivo AndroidManifest.xml [autoría propia].



Curriculum Vitae

Licenciado en Informática por el Instituto Tecnológico de Chetumal (2002 - 2006). Con más de 11 años de experiencia laboral como analista – programador de sistemas, de aplicaciones móviles, administrador de bases de datos y de red de voz y datos en la Oficialía Mayor, Secretaría de la Gestión Pública y Secretaría de Turismo del Estado de Quintana Roo. He sido también docente de nivel medio superior en la escuela “Eva Sámano de López Mateos” (2011). Entre mis principales objetivos profesionales está la creación de soluciones informáticas a la medida que permitan el óptimo funcionamiento de los procesos organizacionales, así como una actualización constante en las técnicas y herramientas para el desarrollo de software y de esta manera lograr una superación profesional continua.

Mi correo es angel.8246@gmail .com y angel_t_r@hotmail.com, número celular 9831149426.

Domicilio Permanente: Heriberto Jara #256. Colonia Adolfo López Mateos.
Chetumal, Quintana Roo, C.P. 77010.

Esta tesis fue mecanografiada por Angel Guillermo Trejo Rojas.