



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

**DISEÑO DE UN SISTEMA INFORMÁTICO PARA LA
LOCALIZACIÓN DE AVES EN MÉXICO**

**PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA

HÉCTOR CERÓN GUEVARA



**DIRECTOR DE TESIS:
DR. DANIEL TREJO MEDINA**

CIUDAD UNIVERSITARIA SEPTIEMBRE 2015



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicada a mis padres, mi pilar principal para poder seguir adelante.

“La paciencia es una virtud, no tires en saco roto este conocimiento”

Agradezco a mis maestros que me dieron todo su apoyo para concluir este proyecto, en específico a Jorge Yabra, el mejor maestro y mi amigo.

Agradezco a mi tutor el Dr. Daniel Trejo Medina, que cuando muchos abandonaron este proyecto, él vio futuro en mí y me apoyó en gran medida a la conclusión del mismo.

Agradezco al Dr. Adolfo Navarro y al M.C. Alejandro Gordillo, coordinadores del Museo de Zoología Alfonso L. Herrera de la Facultad de Ciencias de la Universidad Nacional Autónoma de México por haberme brindado la oportunidad de colaborar con ellos y compartir su conocimiento para esta tesis.

Contenido

I. Introducción.....	1
II. Justificación.....	4
III. Objetivo General	5
IV. Hipótesis	5
V. Metodología	6
VI. Antecedentes	8
VII. Marco de referencia	16
VIII. Desarrollo.....	30
1. Introducción	30
1.1. Propósito	30
1.2. Convenciones	30
1.3. Audiencia dirigida.....	30
1.4. Alcance	30
2. Descripción General	31
2.1. Perspectiva del producto.....	31
2.2. Funciones del producto.....	31
2.3. Características del usuario.....	32
2.4. Restricciones.....	32
2.5. Suposiciones y Dependencias	32
2.6. Requisitos futuros.....	32
2.7. Entorno Operativo	32
3. Requerimientos de Interfaz Externa	33
3.1. Interfaz de Usuario	33
3.2. Interfaz de Administrador	34
3.3. Interfaz de Hardware.....	36
3.4. Interfaz de Software	36
4. Características del sistema.....	36
4.1. Funciones de usuario	36
4.2. Funciones de administrador	37
5. Otros requerimientos no funcionales	40
5.1. Requerimientos de Desempeño (Performance Requirement).....	40
5.2. Requerimientos de seguridad (Física).....	41

5.3. Requerimientos de seguridad.....	41
5.4. Atributos de Calidad	41
6. Diseño de interfaz de usuario	42
6.1. Interfaz de usuario.....	42
6.2 Descripción de la interfaz de usuario.....	42
6.3 Pantallas.....	43
6.4 Diagramas de interfaces de usuario	57
IX. Presentación de resultados.....	63
X. Conclusiones.....	65
XI. Referencias	68
XII. Anexos	72
ANEXO 1 CÓDIGO FUENTE.....	72
ANEXO 2 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE.....	76
1. Introducción.....	77
2. Descripción General	78
3. Requerimientos de Interfaz Externa	80
4. Características del sistema	84
5. Otros requerimientos no funcionales	86
ANEXO 3 DESCRIPCIÓN DEL DISEÑO DE SOFTWARE.....	89
1. Introducción.....	90
2. Propósito General.....	91
3. Requerimientos de Interfaz Externa	93
4. Características del sistema	111
5. Otros requerimientos no funcionales	117
ANEXO 4 ACUERDO DE TRABAJO	120
1. Introducción.....	121
2. Alcance General.....	121
3. Actividades y responsabilidades	122
4. Periodo de actividades	123
5. Calendario de eventos.....	124
6. Estándares a ocupar	124
7. Requerimientos especiales	124
8. Criterio y forma de pago	125
ANEXO 5 MANUAL DE USUARIO	127

1. Introducción.....	128
2. Conceptos importantes	129
3. Guía de uso.....	129
ANEXO 6 MANUAL DE ADMINISTRADOR	135
1. Introducción.....	136
2. Conceptos importantes	136
3. Guías de uso	137
ANEXO 7 MANUAL DE INSTALACIÓN	141
1. Introducción.....	142
2. Requerimientos iniciales	143
3. Instalación del sistema	144

Índice de imágenes

Imagen 1. Pantalla de carga. Fuente: propia.....	43
Imagen 2. Pantalla Principal. Fuente: propia.....	44
Imagen 3. Pantalla de Ingreso de Avistamiento. Fuente: propia.....	45
Imagen 4. Pantalla de Ingreso de Avistamiento. Fuente: propia.....	46
Imagen 5. Pantalla Envió de datos. Fuente: propia.....	47
Imagen 6. Pantalla Verificación de Datos. Fuente: propia.....	48
Imagen 7. Pantalla Mapa de Datos. Fuente: propia.....	49
Imagen 8. Pantalla Acerca de. Fuente: propia.....	50
Imagen 9. Pantalla Usuario Verificado. Fuente: propia.....	51
Imagen 10. Pantalla Registros Nuevos. Fuente: propia.....	52
Imagen 11. Pantalla Registros por Especie. Fuente: propia.....	53
Imagen 12. Pantalla Registros por Usuario. Fuente: propia.....	54
Imagen 13. Pantalla Registros por Usuario. Fuente: propia.....	55
Imagen 14. Pantalla Estadísticas. Fuente: propia.....	56
Imagen 15. Pantalla de carga. Fuente: propia.....	96
Imagen 16. Pantalla Principal. Fuente: propia.....	97
Imagen 17. Pantalla de Ingreso de Avistamiento. Fuente: propia.....	98
Imagen 18. Pantalla de Ingreso de Avistamiento. Fuente: propia.....	99
Imagen 19. Pantalla Envió de datos. Fuente: propia.....	100
Imagen 20. Pantalla Verificación de Datos. Fuente: propia.....	101
Imagen 21. Pantalla Mapa de Datos. Fuente: propia.....	102
Imagen 22. Pantalla Acerca de. Fuente: propia.....	103
Imagen 23. Pantalla Usuario Verificado. Fuente: propia.....	104
Imagen 24. Pantalla Registros Nuevos. Fuente: propia.....	105
Imagen 25. Pantalla Registros por Especie. Fuente: propia.....	106
Imagen 26. Pantalla Registros por Usuario. Fuente: propia.....	107
Imagen 27. Pantalla Registros por Usuario. Fuente: propia.....	108
Imagen 28. Pantalla Estadísticas. Fuente: propia.....	109
Imagen 29 Carta de confirmación del uso y aporte valor de la tesis y su producto. Fuente: Carta del Museo de Zoología.....	147

Índice de esquemas

Esquema 1 Proceso de identificación de un ave. Fuente: propia.....	9
Esquema 2. Arquitectura de aplicaciones.	21

Índices de tablas

Tabla 1. Aplicaciones móviles más utilizadas en Android basado en los datos obtenidos de Playstore de Google (Mayo,2015)	12
--	----

Índice de diagramas

Diagrama 1. Diagrama de Interfaces de Usuario. Fuente: propia	35
Diagrama 2. Componentes MVC. Fuente: propia.....	39
Diagrama 3. Diagrama de componentes MVC. Fuente: propia.....	40
Diagrama 4. Diagrama de Interfaces Pantalla Principal. Fuente: propia.....	57
Diagrama 5. Diagrama de Interfaces Pantalla Ingreso de Avistamiento. Fuente: propia.....	58
Diagrama 6. Diagrama de Interfaces Envió de datos. Fuente: propia.....	59
Diagrama 7. Diagrama de Interfaces Verificación de datos. Fuente: propia.....	60
Diagrama 8. Diagrama de Interfaces Verificación de datos. Fuente: propia.....	61
Diagrama 9. Diagrama de Acerca de. Fuente: propia.....	62

I. Introducción

La Universidad Nacional Autónoma de México es reconocida a nivel nacional como la Universidad que genera más investigaciones por año y con los resultados más eficientes gracias a sus Facultades que la conforman. Dentro de estas Facultades se encuentra la Facultad de Ingeniería, donde fue desarrollada esta tesis y la Facultad de Ciencias que será la que recibirá la misma. Esta última a su vez está conformada por varios Institutos, entre ellos el de Biología donde se encuentra el Museo de Zoología Alfonso L. Herrera. El personal que trabaja ahí (investigadores y becarios) se encargan de difundir cada una de sus investigaciones realizadas mediante la observación y análisis del comportamiento animal de diferentes especies entre ellas, las aves.

Para realizar este tipo de análisis, se debe de realizar una observación mediante vista de parte de cada observador y se debe de tener registro de cada uno de éstos. Este tipo de observaciones y anotaciones se lleva haciendo desde hace poco más de 30 años desde su creación del museo, sin embargo la forma de realizar estas anotaciones no ha cambiado mucho, por lo que con el paso del tiempo han comenzado a surgir errores de exactitud acorde con los tiempos actuales, ya que por distintas razones no se han podido actualizar a usar herramientas más exactas. A esto se le junta el problema de que debido a eso, muchos de sus datos no se puedan ser exactos y algunas investigaciones tengan un porcentaje de error que podría afectar tanto la credibilidad del museo como la de la misma Universidad.

Es por esto que con el fin de reducir ese problema y teniendo como premisa la definición de la tesis misma, la cual es resolver un problema de la vida actual, el propósito de esta tesis es presentar el proyecto ante el museo con el fin de darles una herramienta para reducir este tipo de errores, así como para facilitarle la tarea de captura de datos, que si bien no es un problema primordial, les ayudaría a reducir tiempos.

La tesis aplica lo aprendido en las materias de Ingeniería en computación y Administración de proyectos de las cuales se logra hacer la planeación del proyecto como la toma de requerimientos siguiendo los estándares más modernos usados en proyectos de la Ingeniería aplicada; también se usó los conocimientos adquiridos de Bases de datos, Programación avanzada, Algoritmos y estructura de datos y Computación gráfica, todas estas usadas al momento del desarrollo, tomando como base la planeación previamente implementada por todos los requerimientos, ya que sin estos, el proyecto no se podría realizar. Por ultimo pero no menos importante tenemos la materia de redes de datos, la cual nos sirvió para implementar el sistema en tiempo y forma para su uso en el campo y así poder realizar las pruebas sin contratiempos usando la comunicación entre el teléfono inteligente y la aplicación móvil.

Esta tesis sirve como catalizadora del conocimiento de mi carrera y es una forma relevante de hacer difusión del conocimiento aprendido y de divulgación para un estudiante, agradezco a la Universidad y a mis padres el apoyo.

Sin embargo, como se hace notar más adelante no todo fue fácil, ya que existieron problemas y dificultades a la hora del desarrollo de la misma, desde el cambio de requerimientos, pasando por retrasos en tiempo y falta de conocimiento en el área móvil y como punto adicional un poco de tiempo perdido por los trámites lentos que la burocracia provocaba. Y no solo esto, se tuvo un problema grande al no tener conocimiento específico en el área de negocios ya que al momento del cálculo de costos que viene en el anexo Acuerdo de trabajo (Anexo 4) a simple vista, no se contemplan costos que normalmente en el área laboral se consideran.

Durante el desarrollo de la presente tesis obtuve grandes conocimientos en desarrollo móvil y como se relación todas las materias de la universidad al unísono al momento del desarrollo completo de un sistema, ya sea grande o como este caso uno pequeño

II. Justificación

Actualmente, en el Museo de Zoología Alfonso L. Herrera de la Facultad de Ciencias (MZFC) de la Universidad Nacional Autónoma de México (UNAM) se realizan registros periódicos de observaciones de aves en campo abierto. Estos registros se realizan mediante la observación del ave, anotando su nombre en una bitácora. Sin embargo dicha observación no presenta la localización exacta del lugar ya que no se cuenta con equipo de localización satelital profesional que indique donde se encuentran actualmente.

El método actual consiste en, mediante el uso de brújula, decir aproximadamente cuantos metros recorrieron hacia qué dirección desde el lugar de salida. Este método puede presentar una cantidad innumerable de errores ya que, al ser una medición totalmente arbitraria, para un biólogo puede ser una distancia y para otro biólogo puede ser una distancia completamente diferente. Así mismo, a pesar de que las brújulas son precisas, se puede también dar el hecho de que no estén calibradas o simplemente no es exacta la medición. Este tipo de errores generara una tasa de error muy alta al momento de acotar datos.

Tras realizar estas observaciones, los biólogos proceden a anotar en una nueva bitácora todas las observaciones finales en equipo de cómputo de regreso al MZFC ubicado en el Distrito Federal.

Gracias a todos estos errores, junto con los tiempos que se pierden en traslados, anotaciones y registros digitales, el museo está propenso a que las publicaciones generadas por los mismos sean más tardadas en publicar y sobre todo con una cantidad de errores inimaginables que podría comprometer el nombre del museo.

III. Objetivo general

Con el fin de reducir el porcentaje de error y ayudar al personal del MZFC se solicitó desarrollar un sistema con el cual se pueda obtener la localización exacta o muy aproximada de los avistamientos de aves observados en el instante y poder almacenarlas para un futuro análisis de forma rápida y sencilla.

Así mismo se lograra reducir los tiempos desde la observación hasta el análisis con el cual el MZFC podrá realizar este proceso más rápidamente y dar un alcance mayor en tema de exactitud avalado por tecnología de última generación.

IV. Hipótesis

- H1: Si se desarrolla un sistema informático para el manejo de información sobre aves y auxilie en el campo para localizar y almacenar en forma aproximada la ubicación facilitara la administración científica de datos generados por el MZFC.

- H2: El sistema informático reducirá costos de tiempo así como facilitar el análisis futuro de estos datos con mayor exactitud a la que se tenía originalmente.

V. Metodología

El presente trabajo inicio con una metodología cualitativa la cual se define como el modo en que enfocamos los problemas con base a datos descriptivos y buscamos la respuesta más apropiada para resolverlo en forma práctica. (Taylor, 1987).

Tomando en cuenta la problemática base de este trabajo se siguió lo aprendido dentro de la Facultad de Ingeniería de la UNAM (FI) en las diversas materias de la carrera de ingeniero en computación para toma de requerimientos como lo es la materia de ingeniería de software y administración de proyectos.

Para la toma de requerimientos de este proyecto se siguió la metodología del Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers, IEEE por sus siglas en ingles) en el estándar 830 llamado Especificación de Requerimientos de Software (Software Requirements Specification, SRS por sus siglas en ingles). Este estándar especifica la descripción del sistema de software que será desarrollado especificando cada requerimiento funcional y no funcional del sistema y debe incluir los casos de uso que describen las interacciones entre el usuario y el sistema.

El SRS permite una evaluación rigurosa de los requerimientos antes de que se pueda comenzar con el diseño del mismo sistema y reduce el tiempo de rediseño en caso de que existan cambios. También provee de una base realista para la estimación de costos del producto, riesgos y tiempo de trabajo (Bourque y Fairley, 2014). El SRS también enlista la documentación suficiente y necesaria que será requerida para el desarrollo del proyecto (Pressman, 2010).

Posteriormente se realizó un diseño detallado con la arquitectura del sistema y un diseño de la solución conforme al estándar 1016 de la IEEE titulado Descripción de Diseño de Software (Software Design Description, SDD por sus siglas en ingles). Es creado con el fin de dar a conocer el diseño del sistema así como de sus procesos

específicos. En este estándar se describe en forma detallada cada una de las funciones de cada componente del sistema (Bourque y Fairley, 2014).

Al igual que el SRS, el SDD nos permitirá tener los tiempos, riesgos y costos del desarrollo del sistema, previo a ser desarrollado con el fin de evitar contrariedades y contratiempos.

Gracias a estos estándares aprendidos en la FI, se hizo un acuerdo de trabajo para determinar las especificaciones del proyecto con juntas entre el desarrollador y el usuario final, en este caso el MZFC.

VI. Antecedentes

Para la realización del proyecto es relevante mencionar algunos conceptos que se debieron definir para que en el ámbito de la ingeniería se diera mejor comprensión. A continuación se enlistan.

La ornitología es la rama de la zoología que se dedica al estudio de las aves. (Alvarado y Gómez, 2010). La ornitología es desarrollada en su gran mayoría por biólogos. (Navarro, 1989). Una de las actividades que se realizan en la ornitología es la observación de aves y registros de ellas en bitácoras para realización de censos, conteos y biogeografía (Navarro, 1989; Alvarado y Gómez, 2010).

Pero no es necesario ser un biólogo para poder ayudar a la ornitología, ya que cualquier persona con binoculares, una guía de aves y ropa cómoda puede hacer observaciones. (Guerrero, 2004). Con lo anterior, existen 2 tipos de observadores, los científicos dedicados al estudio de aves y los que las observan de forma recreativa (Alvarado y Gómez, 2010; Navarro, Ortiz y Townsend, 2008).

Es por esto que el sistema está orientado a cualquier tipo de observador. Sin embargo el administrador de este sistema una vez implementado será el MZFC para un uso interno del mismo en un principio y público más adelante.

Pero antes de seguir, tenemos que describir y saber a qué se dedica el personal del MZFC. En la Facultad de Ciencias de la UNAM (FC) para el estudio de la ornitología se creó el MZFC en el año 1978 a partir de la necesidad de hacer investigación y producir colecciones zoológicas en la UNAM, por parte del personal académico de carrera y de los estudiantes de licenciatura y posgrado (CONACULTA, 2015).

Dentro de este museo se crearon las colecciones digitales en bases de datos con el fin de dar acceso libre a quien solicite información de la investigación en la rama de ornitología. Los datos de estas bases son supervisados con el fin de ser confiables

para todo aquel que requiera información de ornitología para continuar con la investigación (FC, 1978).

Dichos datos son alimentados por los observadores de aves tanto propios del Instituto como algunos externos a la UNAM los cuales ayuden a mejorar el servicio (FC, 1978).

En el esquema 1, se puede apreciar cómo se realiza una observación. La identificación del ave se puede lograr gracias a una guía de aves o conocimiento propio. Acerca de la identificación del lugar de observación se logrará gracias a un sistema de GPS o en caso de estar en la ciudad, por calles y colonias propias de la ciudad. Acerca de la anotación propiamente es tener almacenado el dato del nombre de la especie y del lugar. Acerca del ingreso del ave al sistema, este paso lo hace únicamente personal del museo y podrá depurar información que no corresponda con los registros típicos del lugar de observación. Para el futuro análisis el MZFC tendrá todos los datos almacenados, corroborados y podrá generar nuevo conocimiento a partir de éstos.



Esquema 1 Proceso de identificación de un ave. Fuente: propia.

Sin embargo el objetivo de este sistema es reducir tiempos y facilitar la observación de aves de modo que ,de acuerdo con el esquema 1, se pretende simplificar el paso 3, 4 y 5, los cuales lo podrá realizar automáticamente el sistema y el usuario podrá ingresar el ave al sistema directamente para que los encargados del MZFC pueden hacer una depuración de datos para un futuro análisis con mayor facilidad agregando también la exactitud de los datos contenidos, ya que recordando la justificación de este sistema, es para reducir el porcentaje de errores que se tiene actualmente.

Retomando el diagrama 1 de Proceso de identificación de un ave y recordando el objetivo de la tesis, tenemos que tomar en cuenta que el primer paso después de identificar el ave será colocar en algún lugar el dato. Como se requiere reducir tiempos, costos y errores, generaremos una aplicación móvil para localizar vía GPS la localización exacta del lugar del registro.

Con base a esto se tiene como idea generar un sistema compuesto en dos partes, una aplicación móvil y un servicio web para la administración de las bases de datos al cual sólo podrá tener acceso el encargado del MZFC.

Una aplicación móvil es un programa de computadora diseñado para correr en teléfonos inteligentes. Estas aplicaciones pueden utilizar algunos componentes de los teléfonos inteligentes, tal es el caso del GPS, WiFi y el mismo almacenamiento de memoria interna del celular que serán componentes fundamentales en este sistema. El GPS se refiere al sistema de posicionamiento global (Global Positioning System, GPS por sus siglas en inglés) es un sistema de localización basado en posicionamiento de navegación vía satélite. Este servicio es completamente independiente del servicio de telefonía con el que este contratado el teléfono, ya que éste servirá también en caso de emergencia para localizar al individuo. Otra señal que también es fundamental para el uso de esta App es el WiFi. Red de área local inalámbrica (Local Area Wireless Computer Networking, WiFi por sus siglas en inglés) es una tecnología de transferencia de datos que permite conectar el teléfono móvil con internet e incluso generar conexión entre varios celulares. Con estas tecnologías que se encuentran dentro de cada teléfono inteligente se podrá llevar el sistema perfectamente.

En la actualidad se ha comprobado que este tipo de sistemas sirven siempre y cuando se dé la veracidad de los datos. Un ejemplo de esto es la base de datos mundial de eBird.

EBird es una herramienta que consiste en una lista de verificación en tiempo real, la cual permite a cualquier observador de aves (científico o no) registrar las aves observadas en un lugar (eBird, 2013). Lanzado en 2002 por Cornell Lab of Ornithology y National Audubon Society provee una gran cantidad de información básica y especializada para cualquiera que así lo requiera. Basado en un sistema web de recepción de datos y muestreo de estos mismos, EBird es la herramienta más usada por ornitólogos alrededor del mundo por su facilidad, confiabilidad e integración de datos (Navarro, Ortiz y Townsend, 2008).

Sin embargo a pesar de que esta herramienta es muy útil, no se tiene un control exclusivo para México. Es por esto que el sistema está muy enfocado para México, sin embargo este mismo sistema puede ser escalado a todo el mundo dependiendo de la base de aves que se tenga.

Actualmente el listado más confiable basado en los comentarios dados por los encargados del MZFC, que se tiene de aves, exclusivamente de su nombre, es el Listado de Aves Mundial del Comité Internacional de Ornitología (International Ornithological Council, IOC por sus siglas en inglés). El objetivo de esta lista es facilitar la comunicación mundial en ornitología, estandarizar los nombres de las especies y conservación de un historial de nombres de aves por año (Gill y Wright, 2006).

Dicho listado ha permitido la generación de aplicaciones móviles para celulares para la observación de aves. Sin embargo como se verá en la tabla 1, las aplicaciones móviles no cumplen con los requerimientos que solicita el museo, ya que el control no lo lleva un Instituto o Universidad que avale dicha información. Algunas de ellas se basan en información de eBird. Como vimos anteriormente en ese sistema pueden ingresar datos cualquier tipo de observador, el problema recae en que no siempre será una información confiable. Otro detalle acerca de estas aplicaciones móviles es que sólo 1 contiene el GPS activado para la observación de aves. En su mayoría sólo son de registro de aves, más no de localización de las mismas.

Aplicación	Audubon Birds - A Field Guide	iBird Pro	Sibley Birds of North America
Costo	52.8	130.77	263.44
Listado Confiable	No	Si	Si
Localización GPS	No	Si	Si
CARACTERÍSTICAS GENERALES	<p>Guía de consulta</p> <p>Imágenes alta definición</p> <p>Requiere internet forzoso para descarga de imágenes</p> <p>Audio de las aves</p> <p>Contiene libro de notas</p>	<p>Guía para observar aves</p> <p>Basados en eBird</p> <p>No permite agregar anotaciones</p> <p>No muestra observaciones propias para análisis</p> <p>Agrega localización de ave</p>	<p>Fotos de aves vistas en vuelo y en posición estática</p> <p>Mapas detallados de la ubicación de aves</p> <p>Comparación de datos e imágenes de diferentes especies en la misma pantalla</p> <p>Localización de aves cercanas a la zona</p> <p>Utilización del gps para saber que aves se tienen cerca</p> <p>Basados en bibliografía, no en bases</p> <p>Más como una guía de consulta</p>

Tabla 1. Aplicaciones móviles más utilizadas en Android basado en los datos obtenidos de Playstore de Google (Mayo,2015)

El GPS nos permite una localización precisa o muy aproximada del lugar donde se encuentre el observador, sin embargo como cualquier aparato electrónico puede presentar un porcentaje de error. Dicho porcentaje varía de entre 0 metros a 7.6metros, todo dependerá del celular que se tenga. Sin embargo, hablando en coordenadas el porcentaje de error es de 0.5%, lo cual en el campo de la ornitología, se considera como aceptable, ya que tendremos a partir de la coordenada un área aproximada para hacer estudios (FAA, 2006).

Gracias a todo lo anterior podemos saber que un dato será generado con los datos de localización exactos, sin embargo, es importante verificar que los datos mandados por el usuario sean los correctos o que correspondan a la tendencia de apariciones con respecto a la especie registrada. Es por eso que se creará un servicio web el cual nos permita administrar sin problema los datos mandados por el usuario.

Un servicio web (en inglés, Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones como es el caso de este sistema (Sheth, 2008). Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier

plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet (Sheth, 2008). La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Pero todo el servicio web y la aplicación misma se deben desarrollar conforme a una arquitectura de diseño, ya que si no se tuviera ésta podría tener como resultado un desorden al momento de la creación de estas dos. Es por eso que se toma de base el Modelo Vista Controlador (MVC). El MVC es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones (Alicante, 2013). Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo (Bascón, 2004). Está compuesto, como su nombre lo indica, de tres componentes básicos, el modelo, la vista y el controlador.

El modelo es aquel que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia (Alicante, 2013). Es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento que en este caso serán las bases de datos. También define las reglas de negocio (la funcionalidad del sistema) (Bascón, 2004). Un ejemplo de regla puede ser: "Si no se tiene las coordenadas solicitadas al GPS, validar que éste esté prendido". Así mismo lleva un registro de las vistas y controladores del sistema (Alicante, 2013). Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

La vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste (Alicante, 2013). Tiene como tareas recibir datos del modelo y los muestra al usuario (Bascón, 2004). Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

El controlador es aquel que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (Alicante, 2013). Recibe los eventos de entrada y contiene las reglas de gestión de eventos.

Acorde con Bascón generalmente el flujo es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, seleccionado una opción del menú principal).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón observador para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aún así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Como tal el MVC presenta varias ventajas. La aplicación está implementada modularmente. Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación. Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, sólo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos. Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.

Con todo lo anterior se comprenden el problema, la idea de lo que se pretende desarrollar y cómo se desarrollará conforme a que bases, sin embargo, un punto clave antes de comenzar a desarrollar en su totalidad, es entender cada uno de los conceptos que se usarán y por qué están siendo usadas en esta tesis, es por eso que el siguiente capítulo se comentará a fondo que conceptos esenciales del desarrollo de software, así como de teoría de cada herramienta que se usará.

Gracias a lo anterior se entiende el problema, se sabe hacia dónde está dirigido el sistema y la finalidad del mismo. Pero para entender más a fondo cómo será desarrollado y con qué bases se tienen para el desarrollo del mismo, se tiene que tomar en cuenta un marco de referencia que para este sistema, se tendrá que hablar los conocimientos adquiridos en la FI para poder desarrollar este sistema.

VII. Marco de referencia

Como ya se comentó anteriormente, el objetivo de esta tesis será crear un sistema para facilitar la localización de aves al momento de la observación. Haciendo hincapié en lo que sería el proceso del desarrollo del sistema tenemos que recordar el proceso completo.

El sistema compuesto de 2 partes, una parte móvil y una parte administrativa dirigida por el MZFC quá los datos tanto del ave como de la localización que serán envidadas a través de red para ser ingresados en una nueva base datos que estará en su totalidad controlada por el MZFC tanto de permisos de ingresos como permisos de salida. Como se ve todo esto se almacena en bases de datos, pero requerimos saber, de que van las bases de datos.

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos (Korth, 2002). La colección de datos, normalmente denominada base de datos, contiene información relevante para un proyecto o como en este caso un sistema. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente (Ullman, 1997).

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información (Ullman, 1997). La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información (Korth, 2002). Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos (Ullman, 1997).

Uno de los propósitos principales de un SGBD es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos (Ullman, 1997).

Para que el sistema sea útil debe recuperar los datos eficientemente. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos (Korth, 2002). Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadores, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema. De acuerdo con Ullman, así como con Korth se tienen los siguientes niveles de abstracción:

- Nivel físico: El nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.
- Nivel lógico: El siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
- Nivel de vistas: El nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios del sistema de base de datos no necesitan toda esta información. En su lugar, tales usuarios necesitan acceder sólo a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la

abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos

Bajo la estructura de la base de datos se encuentra el modelo de datos, una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia (Ullman, 1997). Para ilustrar el concepto de un modelo de datos, describimos dos modelos de datos: el modelo entidad-relación y el modelo relacional (Ullman, 1997). Los diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes acorde con Ullman y Korth: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos.

El modelo de datos entidad-relación (MER) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una cosa u objeto en el mundo real que es distinguible de otros objetos (Korth, 2002). Por ejemplo, cada ave es una entidad, y las localizaciones pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades (Korth, 2002). El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones (Korth, 2002).

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único (Ullman, 1997). El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos (Ullman, 1997).

Las columnas de la tabla corresponden a los atributos del tipo de registro. El modelo de datos relacional es el modelo de datos más ampliamente usado (Ullman, 1997), y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional (Korth, 2002), que es el que usamos actualmente en este sistema. El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos MER. Los diseños de bases de datos a menudo se realizan en el modelo MER, y después se traducen al modelo relacional.

El modelo orientado a objetos se puede observar como una extensión del modelo MER con las nociones de encapsulación, métodos (funciones) e identidad de objeto. El modelo de datos relacional orientado a objetos combina las características del modelo de datos orientado a objetos y el modelo de datos relacional (Korth, 2002). Los modelos de datos semi-estructurados permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos (Korth, 2002). Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos (Korth, 2002). El lenguaje de marcas extensible (eXtensible Markup Language, por sus siglas en inglés XML) se usa ampliamente para representar datos semi-estructurados como es el caso de este sistema.

Este tipo de datos almacenados en XML lo usamos para la transferencia de datos entre el sistema móvil y el servidor donde se almacenaran los datos en el MZFC ya que la facilidad y compresibilidad que presentan nos permite mandar y enviar una cantidad de datos mayor con una cantidad menor de espacio y la transferencias de archivos entre uno y otro se realicen con mayor velocidad ahorrando tiempo y espacio en el disco duro del teléfono inteligente.

Un sistema de bases de datos proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones (Korth, 2002). En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes

separados; en su lugar simplemente forman partes de un único lenguaje de bases de datos, tal como SQL que es el usado en este sistema por su facilidad de comprender y su conocimiento que se tiene del mismo.

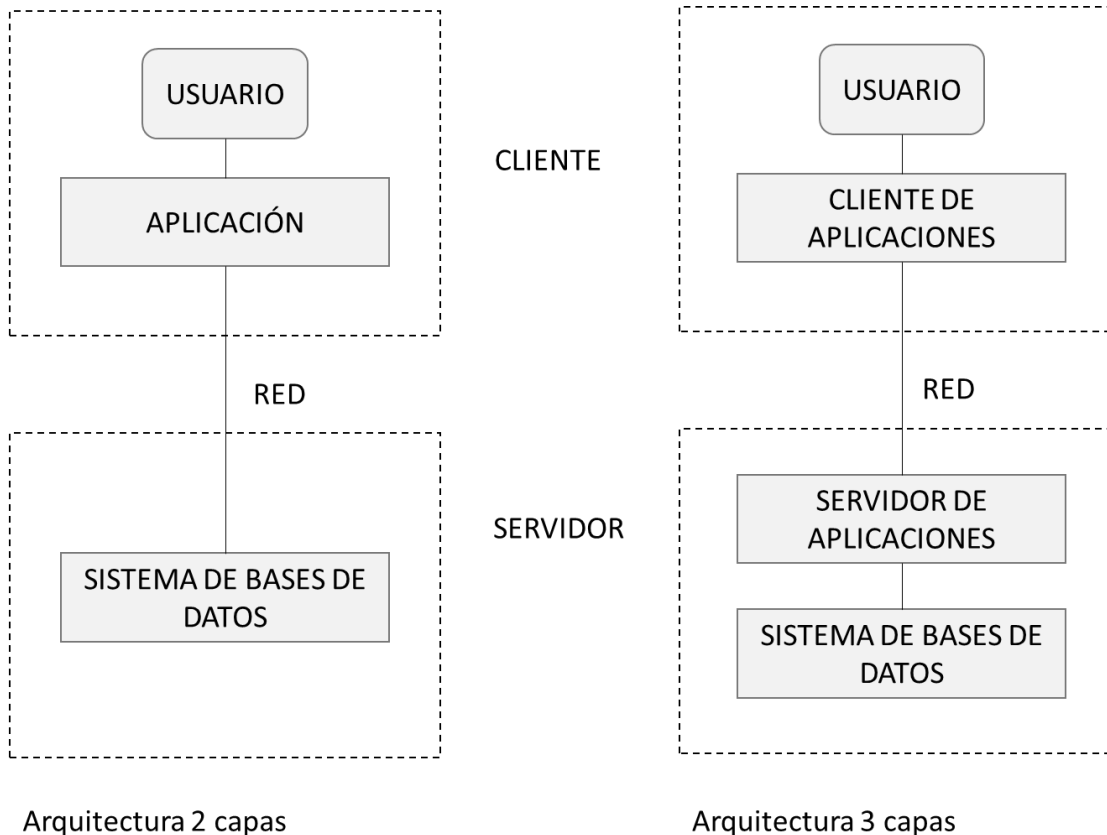
Un lenguaje de definición de datos (LDD) es un lenguaje el cual permite a los usuarios generar, almacenar y modificar datos dentro de las bases de datos. Un lenguaje de manipulación de datos (LMD) es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado (Korth, 2002).

Con lo anterior tenemos que SQL permite insertar, manipular y acceder a la información y debido a esto es por lo que usamos SQL ampliamente en el sistema.

Retomando la parte del sistema que se compone de bases de datos, es fundamental saber la arquitectura del mismo. Las aplicaciones de bases de datos se dividen usualmente en dos o tres partes. En una arquitectura de dos capas, la aplicación se divide en un componente que reside en la máquina cliente, que llama a la funcionalidad del sistema de bases de datos en la máquina servidor mediante instrucciones del lenguaje de consultas.

Los estándares de interfaces de programas de aplicación como ODBC y JDBC se usan para la interacción entre el cliente y el servidor. En cambio, en una arquitectura de tres capas, la máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar, el cliente se comunica con un servidor de aplicaciones, usualmente mediante una interfaz de formularios.

El servidor de aplicaciones, a su vez, se comunicó con el sistema de bases de datos para acceder a los datos. La lógica de negocio de la aplicación, que establece las acciones a realizar bajo determinadas condiciones, se incorpora en el servidor de aplicaciones, en lugar de ser distribuida a múltiples clientes. Para nuestro sistema se usará una arquitectura de tres capas.



Esquema 2. Arquitectura de aplicaciones.

Como se ve en el esquema ya estamos hablando de aplicaciones, pero para saber de aplicaciones debemos de irnos un poco atrás, como se programa la aplicación. En este caso la móvil.

La programación Orientada a Objetos (POO) es una metodología que basa la estructura de los programas en torno a los objetos. Los lenguajes de POO ofrecen medios y herramientas para describir los objetos manipulados por un programa (Carballo, 2005). Más que describir cada objeto individualmente, estos lenguajes proveen una construcción (Clase) que describe a un conjunto de objetos que poseen las mismas propiedades. Un objeto es una entidad caracterizada por sus atributos propios y cuyo comportamiento está determinado por las acciones o funciones que pueden modificarlo, así como también las acciones que requiere de otros objetos. Un objeto tiene identidad e inteligencia y constituye una unidad que oculta tanto datos

como la descripción de su manipulación (Carballo, 2005). Puede ser definido como una encapsulación y una abstracción: una encapsulación de atributos y servicios, y una abstracción del mundo real. Para el contexto del Enfoque Orientado a Objetos (EOO) un objeto es una entidad que encapsula datos (atributos) y acciones o funciones que los manejan (métodos). También para el EOO un objeto se define como una instancia o particularización de una clase. Los objetos de interés durante el desarrollo de software no sólo son tomados de la vida real (objetos visibles o tangibles), también pueden ser abstractos (Carballo, 2005). En general son entidades que juegan un rol bien definido en el dominio del problema. Dentro de un objeto, los datos que llamamos se titulan atributos mientras que las acciones o funciones que realiza el objeto se llaman métodos.

La clase es la unidad de modularidad en el EOO (Carballo, 2005). La tendencia natural del individuo es la de clasificar los objetos según sus características comunes (clase). La clase puede definirse como la agrupación o colección de objetos que comparten una estructura común y un comportamiento común (Carballo, 2005). Consta de atributos y métodos que resumen las características y el comportamiento comunes de un conjunto de objetos. Todo objeto (también llamado instancia de una clase), pertenece a alguna clase. Mientras un objeto es una entidad concreta que existe en el tiempo y en el espacio, una clase representa sólo una abstracción. Todos aquellos objetos que pertenecen a la misma clase son descritos o comparten el mismo conjunto de atributos y métodos. Todos los objetos de una clase tienen el mismo formato y comportamiento, son diferentes únicamente en los valores que contienen sus atributos.

Algorítmicamente, las clases son descripciones netamente estáticas o plantillas que describen objetos. Su rol es definir nuevos tipos conformados por atributos y operaciones (Carballo, 2005). Por el contrario, los objetos son instancias particulares de una clase. Las clases son una especie de molde de fábrica, en base al cual son construidos los objetos. Durante la ejecución de un programa sólo existen los

objetos, no las clases (Carballo, 2005). La declaración de una variable de una clase no crea el objeto.

Teniendo esto como base para programar también se requiere hablar de los patrones de diseño.

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software (Damián, 2009). Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema, la cual puede ser utilizada en diversas situaciones. Los patrones de diseño reflejan todo el rediseño y remodificación que los desarrolladores han ido haciendo a medida que intentaban conseguir mayor reutilización y flexibilidad en su software. Los patrones documentan y explican problemas de diseño, y luego discuten una buena solución a dicho problema. Con el tiempo, los patrones comienzan a incorporarse al conocimiento y experiencia colectiva de la industria del software, lo que demuestra que el origen de los mismos radica en la práctica misma más que en la teoría.

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable (Damián, 2009). El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema concreto, describiendo cuándo aplicarlo y si tiene sentido hacerlo teniendo en cuenta otras restricciones de diseño, así como las consecuencias, ventajas e inconvenientes de su uso.

Para que una solución sea considerada un patrón debe poseer ciertos componentes fundamentales acorde con Damián:

- Nombre del patrón. Permite describir en pocas palabras un problema de diseño junto con sus soluciones y consecuencias.

- Problema (o fuerzas no balanceadas). Indica cuándo aplicar el patrón. En algunas oportunidades el problema incluye una serie de condiciones que deben darse para aplicar el patrón. Muestra la verdadera esencia del problema. Este enunciado se completa con un conjunto de fuerzas, término que se utiliza para indicar cualquier aspecto del problema que deba ser considerado a la hora de resolverlo, entre ellos:

-Requerimientos a cumplir por la solución.

- Restricciones a considerar.

-Propiedades deseables que la solución debe tener.

- Solución. No describe una solución o implementación en concreto, sino que un patrón es más bien como una plantilla que puede aplicarse en diversas situaciones diferentes. El patrón brinda una descripción abstracta de un problema de diseño y cómo lo resuelve una determinada disposición de objetos. Que la solución sea aplicable a diversas situaciones denota el carácter “recurrente” de los patrones.

- Consecuencias. Resultados en términos de ventajas e inconvenientes.

Los patrones de diseño refinan subsistemas o componentes de software orientado a objetos. Éstos describen una estructura genérica sobre el problema a resolver, de manera tal de poder contextualizarla con el problema a resolver (Damián, 2009).

Los patrones de diseño están agrupados en tres grupos: Patrones Creacionales, Patrones Estructurales y Patrones de Comportamiento.

Los patrones creacionales abstraen el proceso de instanciación de objetos, ayudando a que el sistema sea independiente de cómo se crean, componen y representan sus objetos (Damián, 2009). Estos patrones encapsulan el conocimiento sobre las clases concretas que utiliza el sistema. En otras palabras, estos patrones brindan soporte a una de las tareas más comunes dentro de la programación orientada a objetos: la instanciación. Estos patrones brindan las siguientes características:

- Instanciación genérica: permite que los objetos sean creados dentro del sistema sin especificar clases concretas en el código.

- Simplicidad: algunos patrones facilitan la creación de objetos, evitando que el cliente deba tener código complejo sobre como instanciar un determinado objeto.
- Restricciones creacionales: algunos patrones ayudan a establecer restricciones sobre la creación de objetos, tales como qué objeto crear, cuándo, cómo, etc.

Por lo general, son alternativas de diseño bajo estrategias de herencia o delegación que encapsulan el mecanismo de creación, independizando los tipos de objetos “producto” que se manejan.

Los patrones creacionales son los siguientes:

1. Singleton: asegura que una determinada clase sea instanciada una y sólo una vez, proporcionando un único punto de acceso global a ella.
2. Abstract Factory: provee una interfaz para crear familias de objetos productos relacionados o que dependen entre sí, sin especificar sus clases concretas.
3. Factory Method: define una interfaz para crear un objeto delegando la decisión de qué clase crear en las subclases. Este enfoque también puede ser llamado constructor “virtual”.
4. Builder: separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones. Simplifica la construcción de objetos con estructura interna compleja y permite la construcción de objetos paso a paso.
5. Prototype: facilita la creación dinámica de objetos mediante la definición de clases cuyos objetos pueden crear duplicados de sí mismos. Estos objetos son llamados prototipos.

Los patrones estructurales se encargan de cómo se combinan clases y objetos para formar estructuras más grandes (Damián, 2009). Los patrones estructurales de clases utilizan la herencia para componer interfaces o implementaciones. En lugar de combinar interfaces o implementaciones, los patrones estructurales de objetos describen formas de componer objetos para obtener nuevas funcionalidades. La flexibilidad añadida mediante la composición de objetos viene dada por la capacidad

de cambiar la composición en tiempo de ejecución, que es imposible con la composición de clases.

Ejemplos típicos son cómo comunicar dos clases incompatibles o cómo añadir funcionalidad a objetos.

Los patrones estructurales son los siguientes:

1. Adapter: funciona de intermediario entre dos clases cuyas interfaces son incompatibles de manera tal que puedan ser utilizadas en conjunto.
2. Bridge: disocia un componente complejo en dos jerarquías de clases: una abstracción funcional y la implementación interna, para que ambas puedan variar independientemente.
3. Composite: compone objetos en estructuras de árboles para representar jerarquías parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los complejos.
4. Decorator: agrega o limita responsabilidades adicionales a un objeto de forma dinámica, proporcionando una alternativa flexible a la herencia para extender funcionalidad.
5. Facade: proporciona una interfaz simplificada para un conjunto de interfaces de subsistemas. Define una interfaz de alto nivel que hace que un subsistema sea más fácil de usar.
6. Flyweight: permite el uso de un gran número de objetos de grano fino de forma eficiente mediante compartimiento.
7. Proxy: Provee un sustituto o representante de un objeto para controlar el acceso a éste. Este patrón posee las siguientes variantes:
 - Proxy remoto: se encarga de representar un objeto remoto como si estuviese localmente.
 - Proxy virtual: se encarga de crear objetos de gran tamaño bajo demanda.
 - Proxy de protección: se encarga de controlar el acceso al objeto representado.

Los patrones de comportamiento tienen que ver con algoritmos y asignación de responsabilidades. Estos patrones se focalizan en el flujo de control dentro de un

sistema (Damián, 2009). Ciertas formas de organizar los controles dentro del sistema pueden llevar a grandes beneficios en cuanto a mantenibilidad y eficiencia. Algunos ejemplos de estos patrones incluyen la definición de abstracciones de algoritmos, las colaboraciones entre objetos para realizar tareas complejas reduciendo las dependencias o asociar comportamiento a objetos e invocar su ejecución. Los patrones de comportamiento basados en clases acorde con Damián utilizan la herencia para distribuir el comportamiento entre clases, ellos son: Template Method e Interpreter. Mientras que los basados en objetos utilizan la composición.

Los patrones de comportamiento acorde con Damián son los siguientes:

1. Chain of responsibility: establece una cadena de mensajes dentro del sistema de manera tal que dicho mensaje sea manejado en el mismo nivel donde fue emitido, o redirigido a un objeto capaz de manejarlo. Evita acoplar el emisor del mensaje con un receptor, dando a más de un objeto la posibilidad de responder al mensaje.
2. Command: representa una solicitud con un objeto, de manera tal de poder parametrizar a los clientes con distintas solicitudes, encolarlas o llevar un registro de las mismas, y poder deshacer las operaciones. Estas solicitudes, al ser representadas como un objeto también pueden pasarse como parámetro o devolverse como resultados.
3. Interpreter: en un contexto donde se repite una determinada clase de problemas y el dominio es bien conocido, se pueden caracterizar estos problemas como un lenguaje y, a su vez, estos problemas pueden ser tratados por un “motor” de interpretación. Este patrón busca definir un intérprete para dicho lenguaje, para el cual define una gramática y un intérprete de la misma para poder resolver los problemas.
4. Iterator: provee un modo de acceder secuencialmente a los elementos de un objeto agregado (una colección) sin exponer su representación interna. El iterador está altamente acoplado al objeto agregado.
5. Mediator: simplifica la comunicación entre objetos dentro del sistema mediante la introducción de un objeto mediador que administra la distribución de mensajes entre objetos. Promueve bajo acoplamiento al evitar que los objetos se referencien unos a

otros explícitamente, permitiendo variar la interacción entre ellos independientemente.

6. Memento: preserva una “fotografía instantánea” del estado de un objeto con el fin de permitirle volver a su estado original, sin revelar su contenido al mundo exterior.

7. Observer: brinda un mecanismo que permite a un componente transmitir de forma flexible mensajes a aquellos objetos que hayan expresado interés en él. Estos mensajes se disparan cuando el objeto ha sido actualizado, y la idea es que quienes hayan expresado interés reaccionen ante este evento.

8. State: permite que un objeto modifique su comportamiento cada vez que cambie su estado interno. El objeto parecerá que cambió de clase.

9. Strategy: define una jerarquía de clases que representan algoritmos, los cuales son intercambiables. Estos algoritmos pueden ser intercambiados por la aplicación en tiempo de ejecución.

10. Template Method: define en una operación el esqueleto de un algoritmo, delegando en las subclasses algunos de sus pasos. Permite que las subclasses redefinan ciertos pasos del algoritmo sin cambiar su estructura.

11. Visitor: representa una operación sobre elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera. Brinda una forma sencilla y mantenible de realizar acciones sobre una familia de clases.

Una forma de conseguir que dos programas se transmitan datos, basada en el protocolo TCP/IP, es la programación de sockets. Un socket es un canal de comunicación entre dos programas que corren sobre ordenadores distintos o en el mismo (Márquez, 2015).

Desde el punto de vista de programación, un socket no es más que un fichero que se abre de una manera especial. Una vez abierto se pueden escribir y leer datos de él.

Acorde con Márquez, existen básicamente dos tipos de sockets:

- Orientados a conexión: ambos programas deben conectarse entre ellos con un socket y hasta que no esté establecida la conexión, ninguno puede transmitir datos. Se trata de TCP y garantiza que todos los datos van a llegar correctamente.

- No orientados a conexión: no es necesario que los programas se conecten. Cualquiera puede transmitir datos en cualquier momento, independientemente de que el otro esté escuchando o no. Es el protocolo UDP, y garantiza que los datos sean correctos, pero no que lleguen todos.

A la hora de comunicar dos programas, existen varias posibilidades para establecer la conexión. Uno de los programas debe estar arrancado y en espera de que otro quiera conectarse al, servidor. El otro programa es el que da el primer paso. En el momento de arrancarlo o cuando lo necesite, intenta conectarse al servidor. Este programa se denomina cliente (Márquez, 2014).

En nuestro caso, la aplicación móvil sería el programa cliente y el servidor sería el que está ubicado en el MZFC esperando siempre datos.

Para poder realizar la conexión entre ambos programas son necesarias varias cosas:

- Dirección IP del servidor: cada ordenador de una red tiene asignado un número único, que sirve para identificarle y distinguirlo de los demás, de forma que cuando un ordenador quiere hablar con otro, manda la información a dicho número.

- Servicio que queremos crear/utilizar: en un mismo ordenador pueden estar corriendo varios programas servidores dando servicios distintos. Cuando un cliente desea conectarse, debe indicar qué servicio quiere. Por ello, cada servicio dentro del ordenador debe tener un número único que lo identifique.

VIII. Desarrollo

Tomando en cuenta los requisitos presentados en la sección de antecedentes y tomando como base los estándares de desarrollo de software de IEEE aprendidos en la Facultad de Ingeniería, se procederá a desarrollar el software.

1. Introducción

1.1. Propósito

El sistema AppVe permitirá a los usuarios ingresar observaciones desde un dispositivo móvil para llevar un registro personal y poder compartir la información con el MZFC.

1.2. Convenciones

Usuario: Definido como cualquier individuo que tenga alguna relación con el MZFC el cual podrá ingresar avistamientos.

Administrador: Encargado de las bases de datos del MZFC el cual validará, depurará y controlará la información que se ingrese al sistema con el fin de hacer confiable la base de datos.

1.3. Audiencia dirigida

Los usuarios serán asignados por el personal del MZFC, sin embargo este sistema puede ser dirigido para cualquier usuario observador de aves.

1.4. Alcance

El sistema será distribuido de manera gratuita por el MZFC, por lo que el alcance esperado será a nivel nacional, siempre y cuando el MZFC apruebe al usuario.

2. Descripción general

Como se presentó en el capítulo VI, para realizar observaciones de aves es necesario seguir una serie de pasos. Con el fin de facilitar la tarea, mejorar los tiempos y reducir el error de localización, esta aplicación tiene como objetivo crear una aplicación móvil para celulares usando la tecnología GPS integrada dentro del celular para tener coordenadas más exactas o aproximadas de donde se realizó la observación del ave. Tras realizar la observación el usuario podrá enviar los datos al MZFC para un análisis a detalle.

2.1. Perspectiva del producto

El sistema completo será validado en un último paso por el MZFC, por lo que todos los datos ingresados serán correctos al final del proceso y podrán ayudar para la investigación científica en el área de la ornitología.

2.2. Funciones del producto

El sistema permitirá a los usuarios:

- Buscar dentro de un catálogo validado los nombres de las especies de aves.
- Ingresar avistamientos en tiempo real con sus coordenadas exactas o muy aproximadas.
- Validar información previa a ser enviada al MZFC.
- Envió de información de observaciones al MZFC
- Llevar un control personal de aves observadas, si fueron enviadas o no, fecha y lugar.

El sistema permitirá al administrador:

- Aprobar o rechazar datos enviados por usuarios.
- Desplegar toda la información obtenida.

2.3. Características del usuario

El usuario debe tener como mínimo conocimientos acerca de identificación de aves, tanto de su nombre como de sus características físicas para poder distinguir entre aves.

2.4. Restricciones

El sistema sólo se podrá correr bajo plataformas Android y es fundamental que el equipo disponga de GPS integrado. Para envío de datos al MZFC es necesario además tener conexión a internet.

2.5. Suposiciones y dependencias

Para el funcionamiento correcto del sistema se requiere:

Aplicación móvil:

- GPS encendido.
- Sistema operativo Android.
- Conexión a Internet (solo en caso de envío de datos).

Panel administrativo:

- Navegador de Internet.
- Adobe Flash instalado.
- Conexión a Internet.

2.6. Requisitos futuros

No aplica.

2.7. Entorno operativo

Aplicación móvil:

Solamente correrá en el sistema operativo Android, cualquier versión.

Panel administrativo:

Al ser dependiente de sólo el navegador de internet, el panel podrá ser consultado en cualquier sistema operativo.

3. Requerimientos de interfaz externa

3.1. Interfaz de usuario

El usuario deberá de contar con GPS encendido, de lo contrario no podrá registrar ninguna avistamiento. Al ingresar al sistema se presentaran 3 opciones: Ingreso de avistamiento, Envió de datos y verificación de datos.

3.1.1. Ingreso de avistamiento

Dentro de esta opción el usuario podrá ingresar un avistamiento de ave seleccionando la especie, en caso de existir algún comentario adicional, se podrá insertar en un campo de texto. En caso de no tener el GPS encendido, la aplicación mandará a la configuración del celular para poder encenderlo, se cerrara la aplicación y se dará por finalizada la sesión hasta abrir de nuevo la aplicación.

3.1.2. Envió de datos

Dentro de esta opción sólo se mostrará aquellos registros que falten de ser enviados al MZFC y permitirá mandar datos uno a uno o todos en conjunto. En caso de que el administrador haya considerado que es un usuario de confianza, antes de enviar un dato se mostrará una nueva pantalla, el cual permitirá al usuario ingresar su nombre para su fácil reconocimiento por el administrador.

3.1.3. Verificación de datos

Dentro de esta opción el usuario podrá ver sus avistamientos, filtrarlos por fecha de registro y podrán ser exportados a un archivo Excel para su análisis. Dicho archivo Excel se localizará en la carpeta raíz de la memoria.

3.2. Interfaz de administrador

El usuario administrador tendrá un panel general el cual se dividirá en 5 pestañas que serán descritas a continuación.

3.2.1. Registros nuevos

Mostrará en una tabla los registros nuevos que se han agregado y que falta por validar. Se podrán validar uno a uno o todos en conjunto. Se mostrara también un mapa para visualizar geográficamente donde fue realizada esa observación. En caso de tener un comentario, se mostrará al administrador.

3.2.2. Registros por especie

Mostrará una tabla con todas las especies registradas, ya sea que hayan sido validadas o no, diferenciando claramente entre aquellas validadas y aquellas que no. Se mostrará un mapa el cual visualizará todos los registros hechos de esa especie. En caso de existir un comentario acerca de un avistamiento en particular, se mostrará al administrador.

3.2.3. Registros por usuario

Mostrará una tabla con todas los registros hechos por usuario, ya sea que hayan sido validadas o no, diferenciando claramente entre aquellas validadas y aquellas que no. Se mostrará un mapa el cual visualizará todos los registros hechos por ese usuario. En caso de existir un comentario acerca de un avistamiento en particular, se mostrará al administrador.

3.2.4. Administrar usuarios

Mostrará el nombre actual del usuario y el nombre válido hecho por el usuario en caso de que el administrador haya considerado a este usuario de confianza. Así mismo el administrador podrá nombrar usuarios de confianza para que su nombre aparezca en la base de datos y se reconozca más fácilmente.

3.2.5. Estadísticas

Mostrará estadísticas de los registros verificados y permitirá exportar los datos a un archivo Excel con diferentes parámetros.

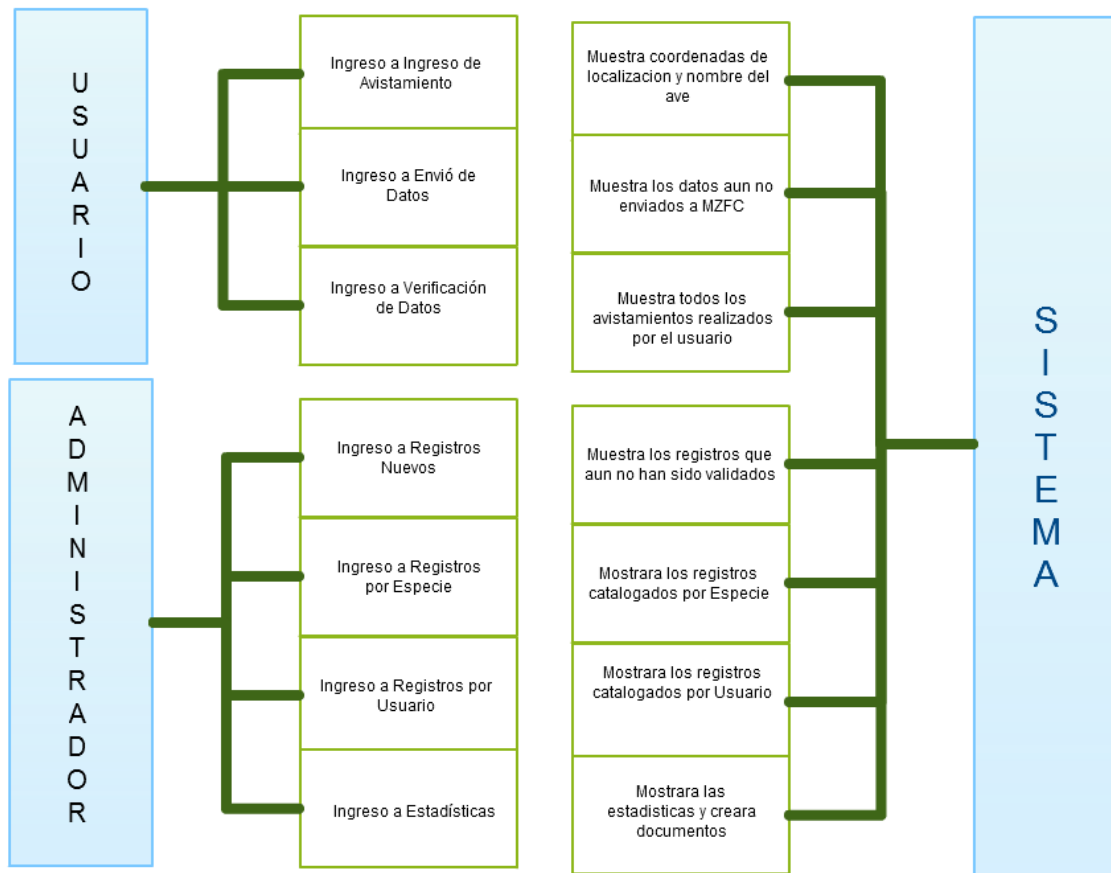


Diagrama 1. Diagrama de interfaces de usuario. Fuente: propia

3.3. Interfaz de hardware

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del usuario:

- Celular con GPS

De parte del Administrador

- Equipo con conexión a internet

3.4. Interfaz de software

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del usuario:

- Celular con sistema operativo Android
- 50 Mb de espacio disponible en memoria

De parte del administrador

- Equipo con navegador web con Adobe Flash instalado

4. Características del sistema

El sistema permitirá al usuario ingresar avistamientos con precisión con ayuda del GPS, validar dicha información, envió de datos al MZFC, ver histórico de registros en mapa y generación de documento acerca de sus propias observaciones. En cuanto al administrador del sistema podrá observar datos nuevos, validar información, controlar el ingreso de datos por usuario y recopilar toda la información en una sola base.

4.1. Funciones de usuario

4.1.1. Ingreso de avistamiento

El sistema le proporcionará al usuario una pantalla mediante la cual el podrá seleccionar del catálogo de aves alguna, así mismo, en caso de tener alguna observación adicional al avistamiento, se podrá realizar anotaciones sobre la misma.

4.1.2. Envió de datos

El sistema mostrará aquellos datos que aún no han sido enviados y permitirá la opción de enviar datos uno a uno o todos. Cuando el administrador da de alta al usuario como usuario confiable, éste recibirá una pantalla nueva que le permitirá ingresar su nombre de usuario

4.1.3. Verificación de datos

El sistema mostrará en forma de lista aquellos datos que han sido enviados y aquellos que no, en conjunto con la fecha y su nombre, adicional a esto, se tendrá la opción de un mapa, el cual nos mostrara los avistamientos del usuario. También se tendrá la opción de almacenar todos los datos en un archivo Excel para su análisis futuro.

4.2. Funciones de administrador

4.2.1. Registros nuevos

El sistema mostrará una pantalla en la que estarán los avistamientos sin registrar. Dichos avistamientos se podrán validar uno a uno o todos. En caso de que se tenga un comentario sobre ese avistamiento, el sistema mostrará el comentario de ese avistamiento. También se incluye un mapa el cual mostrará la localización del avistamiento.

4.2.2. Registros por especie

El sistema mostrará una pantalla con los avistamientos clasificados por especie. Dichos avistamientos pueden estar o no validados, pero se diferenciaron entre ellos. Así mismo se podrá ver los comentarios de cada uno de los avistamientos registrados. Finalmente se agrega un mapa para que por especie se muestren todos los avistamientos en el mapa

4.2.2. Registros por usuario

El sistema mostrará una pantalla con los avistamientos clasificados por usuario. Dichos avistamientos pueden estar o no validados, pero se diferenciarán entre ellos. Así mismo se podrá ver los comentarios de cada uno de los avistamientos registrados. Finalmente se agrega un mapa para que por usuario se muestren todos los avistamientos en el mapa.

4.2.3. Administración de usuarios

El sistema mostrará una pantalla donde se desplegarán las opciones del control de usuario comenzando por aprobar a usuarios confiables, los cuales podrán agregar su nombre para que el administrador pueda diferenciarlos de los otros usuarios. Así mismo se tiene la opción de bloqueo de usuarios, con el fin de que estos usuarios ya no puedan ingresar más datos basura.

4.2.4. Estadísticas

El sistema mostrará distintas gráficas acorde a datos verificados. Dentro de esta misma opción tenemos la exportación de datos en forma de Excel para un análisis futuro.



Diagrama 2. Componentes MVC. Fuente: propia.

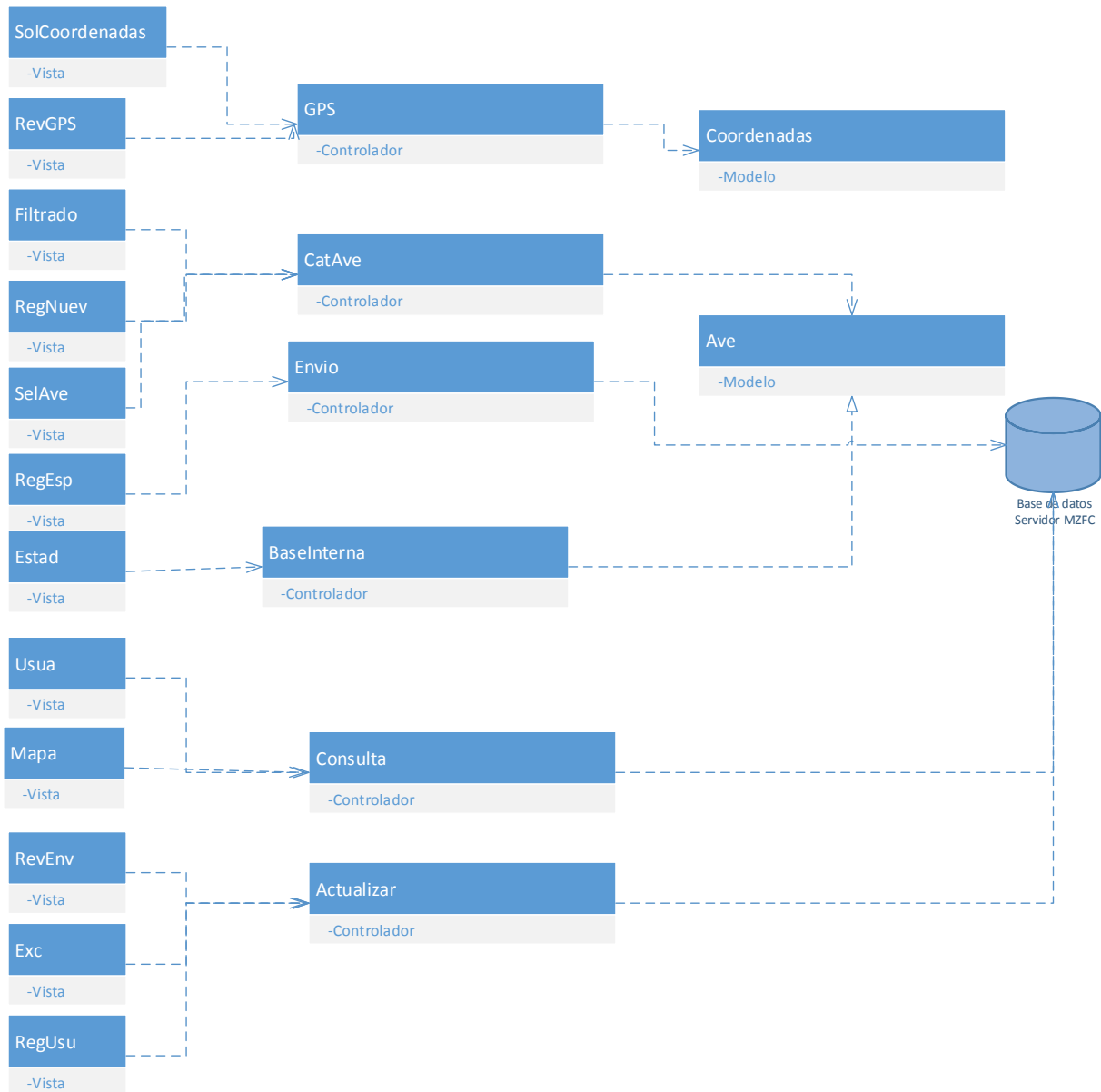


Diagrama 3. Diagrama de componentes MVC. Fuente: propia.

5. Otros requerimientos no funcionales

5.1. Requerimientos de desempeño (Performance Requirement)

El sistema se logrará hacer lo suficientemente ligero como para que cualquier celular con sistema operativo Android pueda correrlo.

5.2. Requerimientos de seguridad (Física)

El servidor del sistema debe contar con instalaciones adecuadas: sin polvo, sin cambios bruscos de voltaje y con nula o poca humedad.

5.3. Requerimientos de seguridad

El sistema no maneja datos de información sensible para el usuario, sin embargo todos los avistamientos al tener que ser verídicos, se pide una seguridad al menos media o alta para resguardar todos estos datos, así como generación de respaldos periódicos.

5.4. Atributos de calidad

El sistema cumple con los siguientes atributos de calidad.

- **Confiabilidad**
El sistema estará siempre funcionando si se sigue correctamente los requerimientos básicos de hardware.
- **Robustez**
Al estar compuesto por componentes de uso amplio (Flex y MySQL) se garantiza este punto.
- **Disponibilidad**
Si el servidor se mantiene en línea todo el tiempo, este sistema se podrá consultar en cualquier momento. Si por alguna situación el servidor se apaga, el sistema funcionará independiente en la aplicación móvil para el usuario.
- **Mantenibilidad**
La actualización del sistema y el cuidado de las bases de datos, así como de la aplicación correrán a cargo del administrador del MZFC, una vez entregado el sistema.
- **Usabilidad**
El sistema será lo suficientemente intuitivo y simple para que cualquier tipo de usuario pueda usarlo.

- Portabilidad

Gracias a que el sistema administrativo no depende del sistema operativo, cumplimos con este atributo.

6. Diseño de interfaz de usuario

6.1. Interfaz de usuario

La interfaz de usuario consiste en una serie de pantallas con las que el usuario puede interactuar con el sistema. Cada una de estas, está diseñada para hacer una tarea específica.

6.2 Descripción de la interfaz de usuario

Cada pantalla estará integrada por varios componentes: botones, barras y etiquetas dispuestos para que el usuario realice las acciones eficientemente.

6.3 Pantallas

El sistema está compuesto de las siguientes pantallas.

6.3.1 Pantalla de carga

Es la primera pantalla que observará el usuario. Esta pantalla no presenta alguna funcionalidad importante, sin embargo se muestra para describir la App.

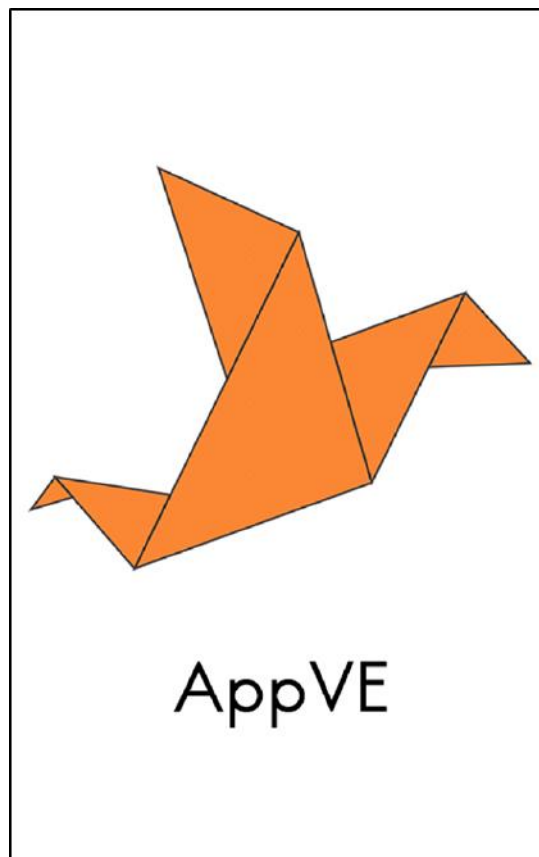


Imagen 1. Pantalla de carga. Fuente: propia.

6.3.2 Pantalla principal

Es la principal pantalla de la aplicación la cual nos permitirá ir a las siguientes pantallas de la aplicación sin problema alguno.

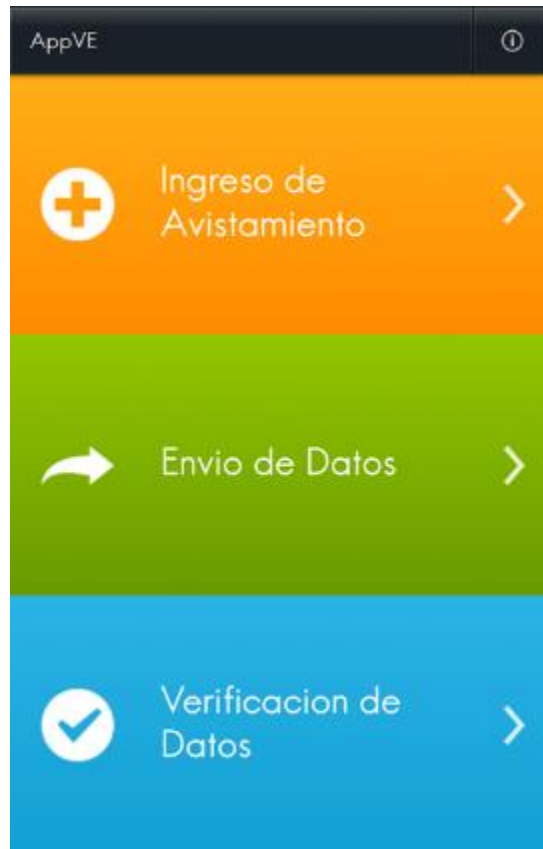


Imagen 2. Pantalla principal. Fuente: propia.

6.3.3 Pantalla ingreso de avistamiento

Es la primera opción que se encuentra en la pantalla principal, después de dar clic en Ingreso de avistamiento. En esta pantalla nos mostrará la lista de aves, la ubicación geográfica y el campo de observaciones.

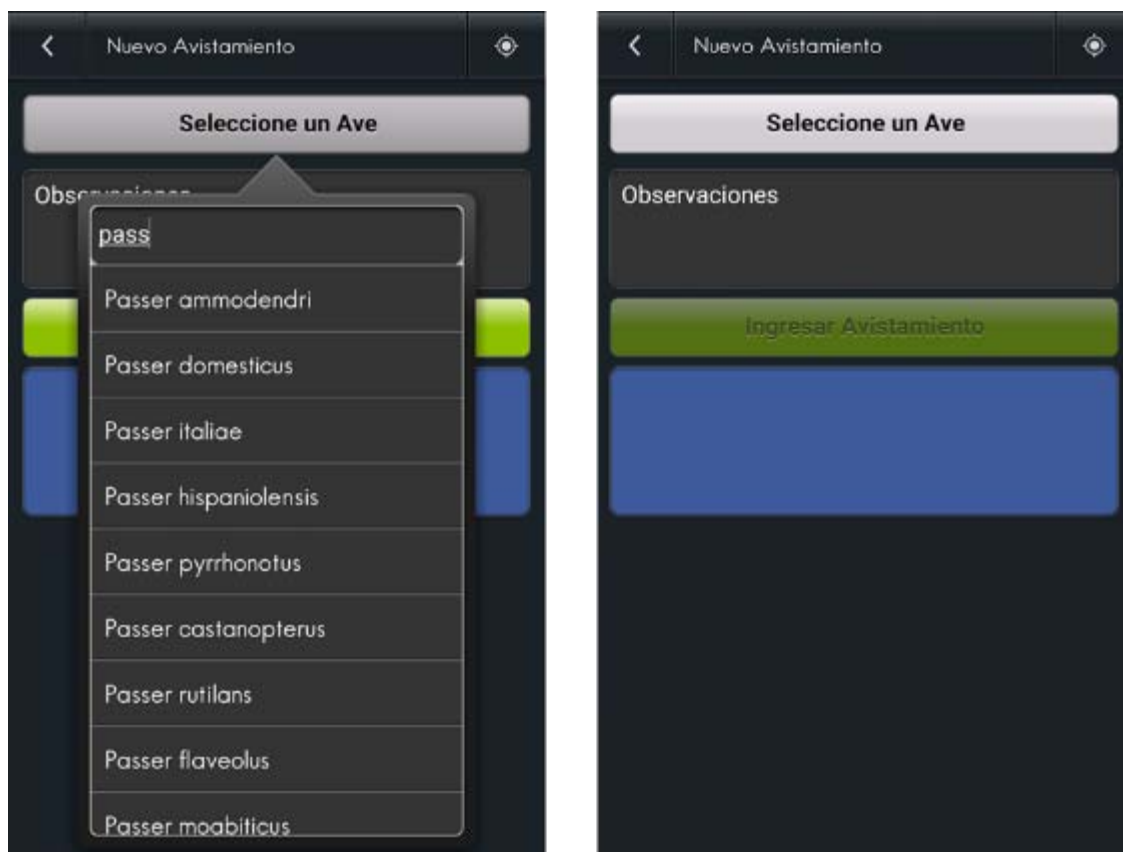


Imagen 3. Pantalla de ingreso de avistamiento. Fuente: propia.

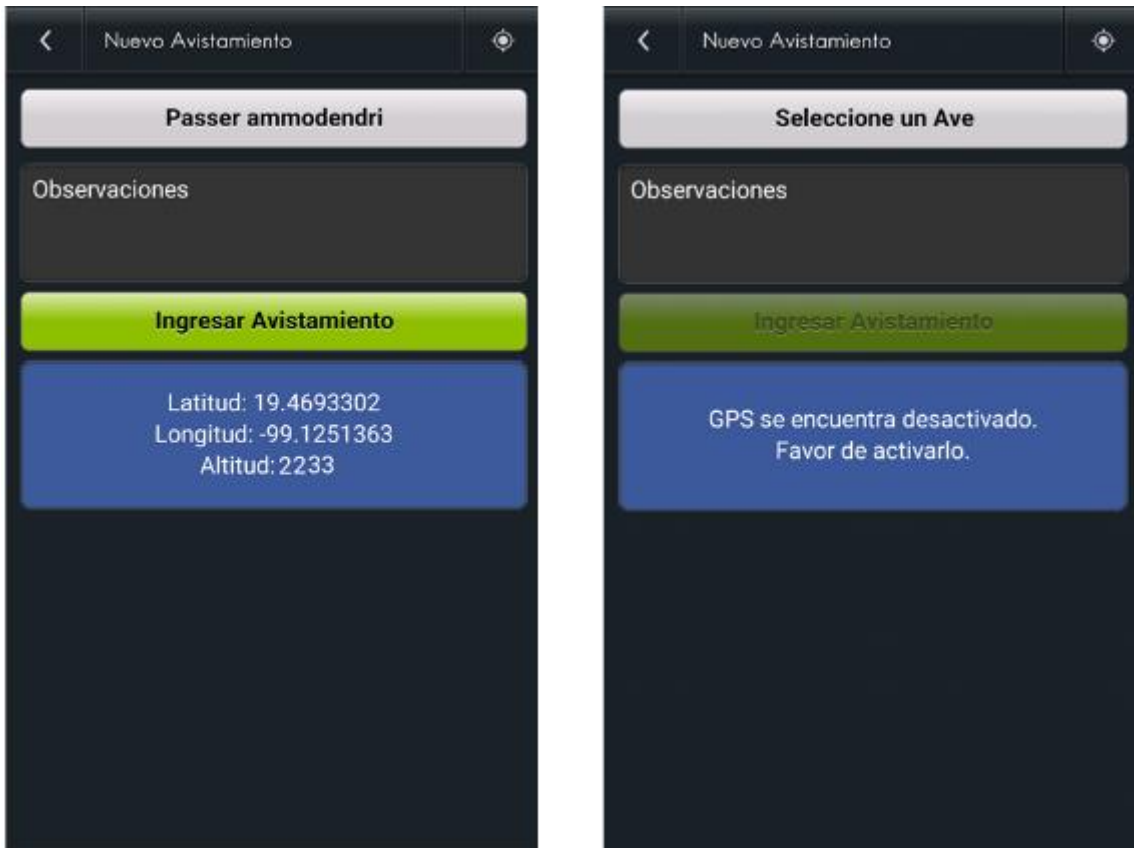


Imagen 4. Pantalla de ingreso de avistamiento. Fuente: propia.

6.3.4 Pantalla envió de datos

Es la segunda opción que se encuentra en la pantalla principal, después de dar clic en envió de datos. En esta pantalla nos mostrará los avistamientos que faltan por enviarse al MUZFC.

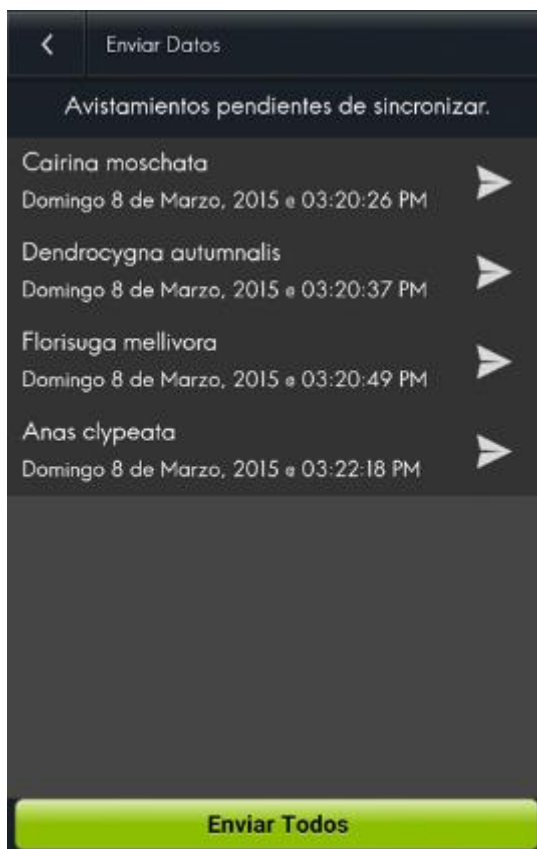


Imagen 5. Pantalla envió de datos. Fuente: propia.

6.3.5 Pantalla verificación de datos

Es la tercera opción que se encuentra en la pantalla principal, después de dar clic en verificación de datos. En esta pantalla nos mostrará los avistamientos que faltan por enviarse al MUZFC.



Imagen 6. Pantalla verificación de datos. Fuente: propia.

6.3.6 Pantalla mapa de datos

Ubicado en la esquina superior derecha dentro de la pantalla de verificación de datos, la cual nos permitirá ver en forma más geográfica los datos obtenidos en nuestros registros.

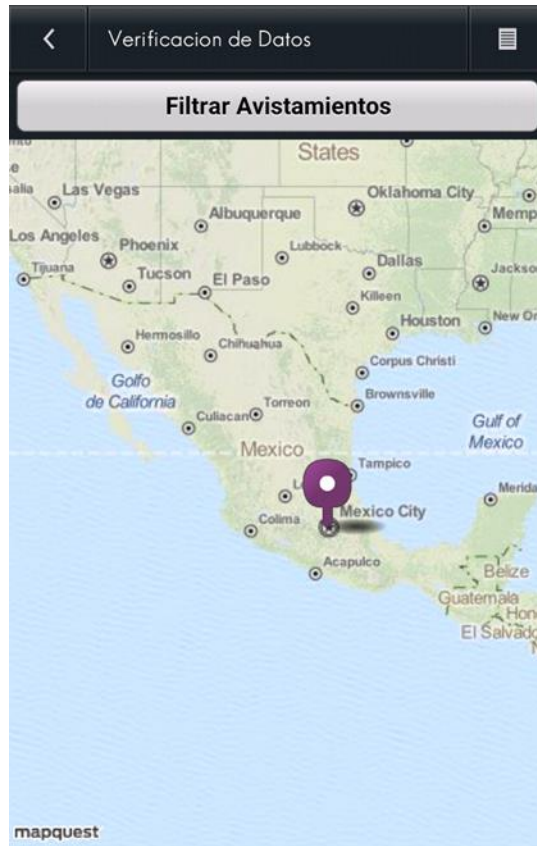


Imagen 7. Pantalla mapa de datos. Fuente: propia.

6.3.7 Pantalla Acerca de

Ubicado en la esquina superior derecha dentro de la pantalla Principal, nos muestra los créditos y el uso de esta aplicación.

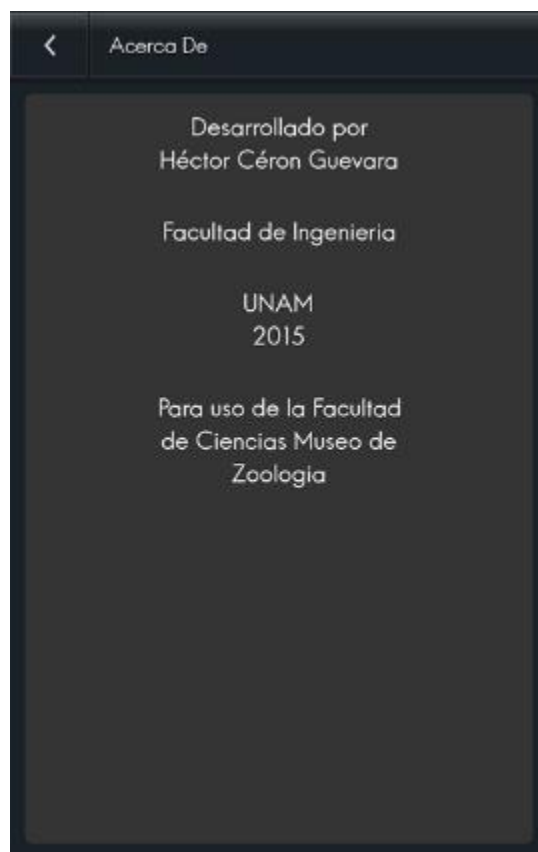


Imagen 8. Pantalla Acerca de. Fuente: propia.

6.3.8 Pantalla usuario verificado

Pantalla de acceso exclusivo luego de que el administrador haya aprobado al usuario como confiable, permitirá ingresar un nombre para su fácil identificación en el sistema del administrador.

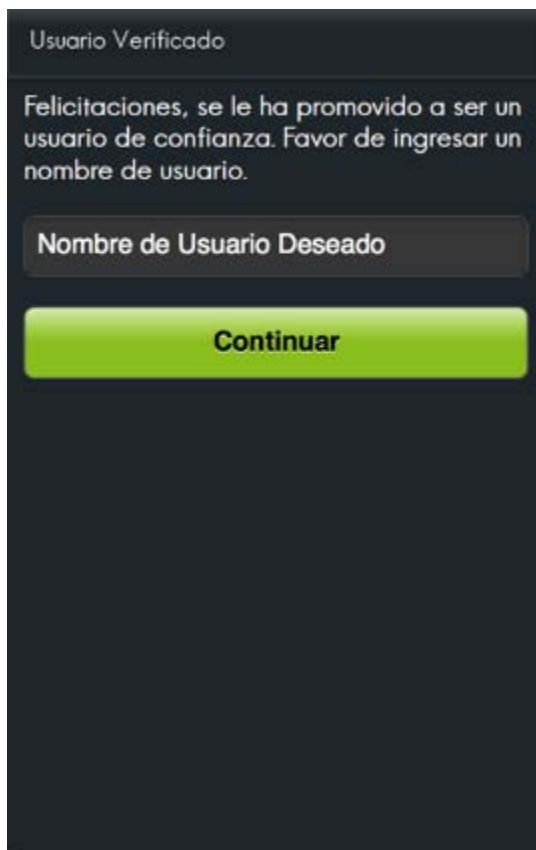


Imagen 9. Pantalla usuario verificado. Fuente: propia.

6.3.10 Pantalla registros por usuario

Pantalla en la que se muestran los registros por usuario. Se muestra también que aquellas que no están pintadas de verde son aquellas que no han sido aún verificadas.

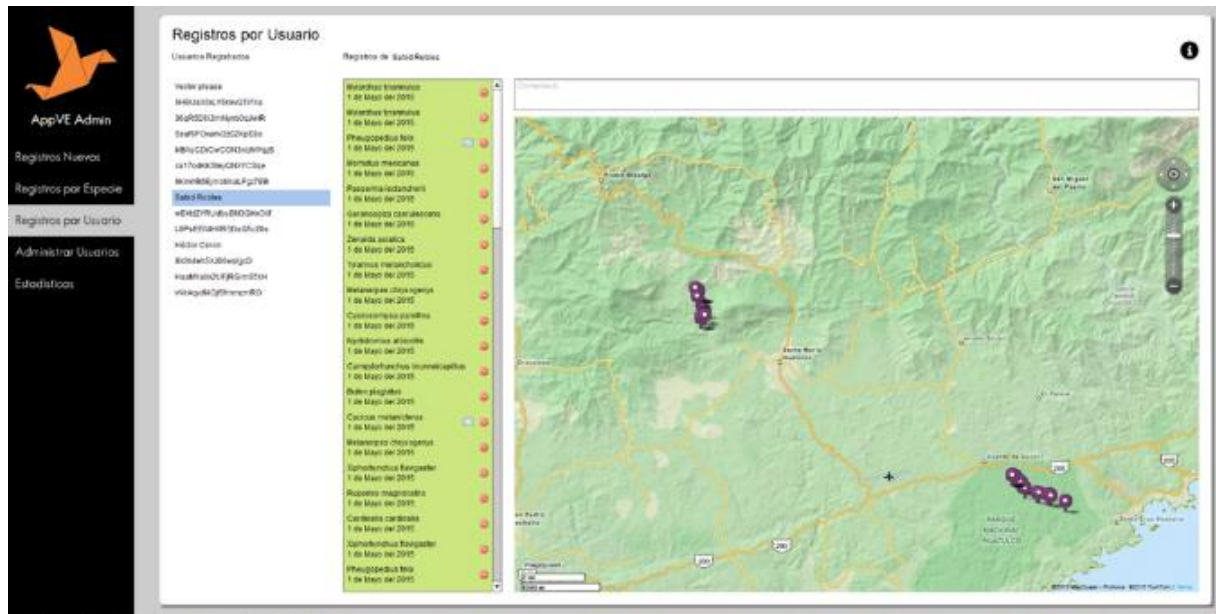


Imagen 12. Pantalla registros por usuario. Fuente: propia.

6.3.11 Pantalla administrar usuarios

Pantalla en la que se muestran los usuarios válidos y no válidos, así como las opciones para volverlos confiables o impedirles el acceso.

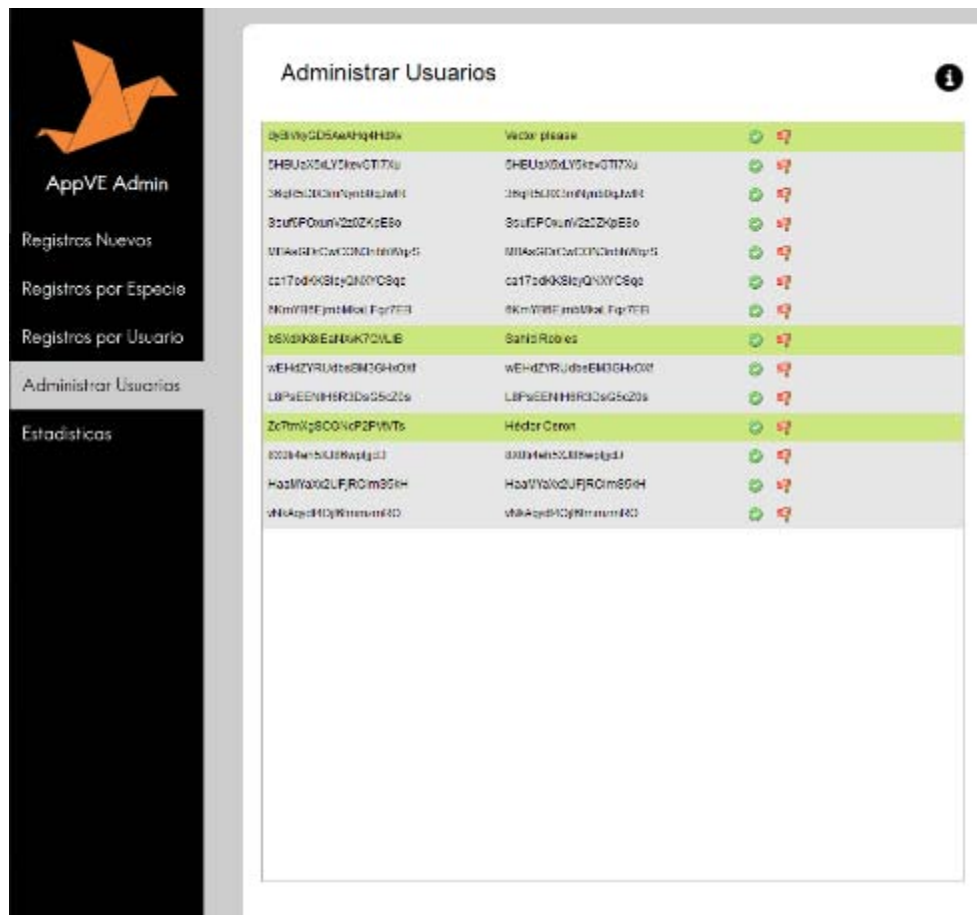


Imagen 13. Pantalla registros por usuario. Fuente: propia.

6.3.12 Pantalla estadísticas

Pantalla en la que se muestran las estadísticas generales de las bases de datos y que permite crear archivos de Excel para un análisis a futuro.

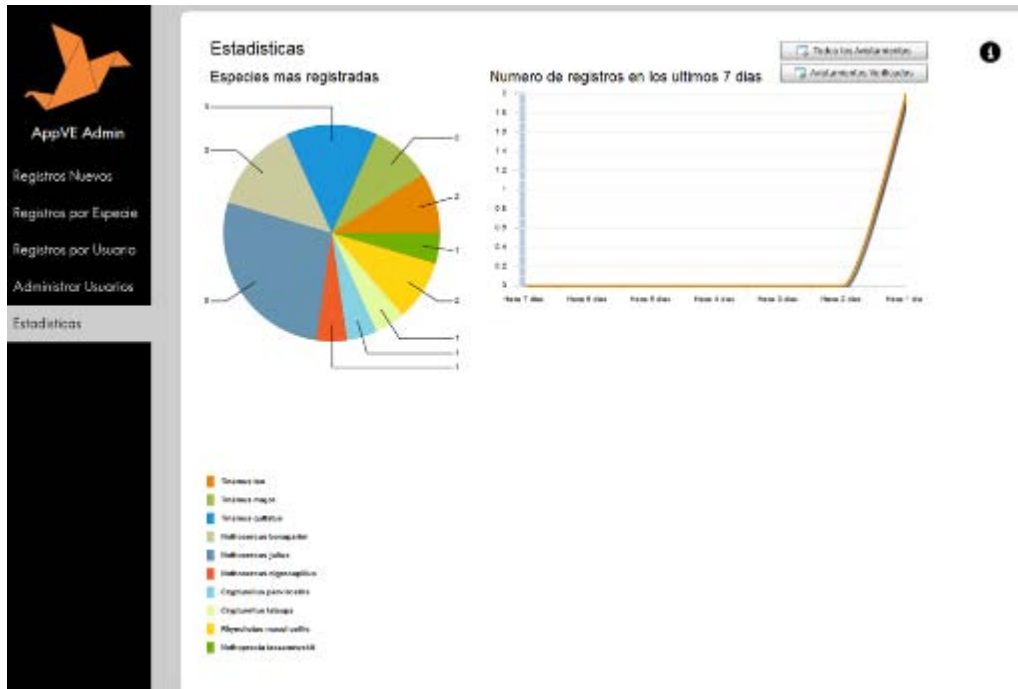


Imagen 14. Pantalla estadísticas. Fuente: propia.

6.4 Diagramas de interfaces de usuario

6.4.1 Pantalla de carga

Descripción:

El usuario aguardará a que cambie de pantalla en lo que la aplicación carga todos sus componentes.

6.4.2 Pantalla principal

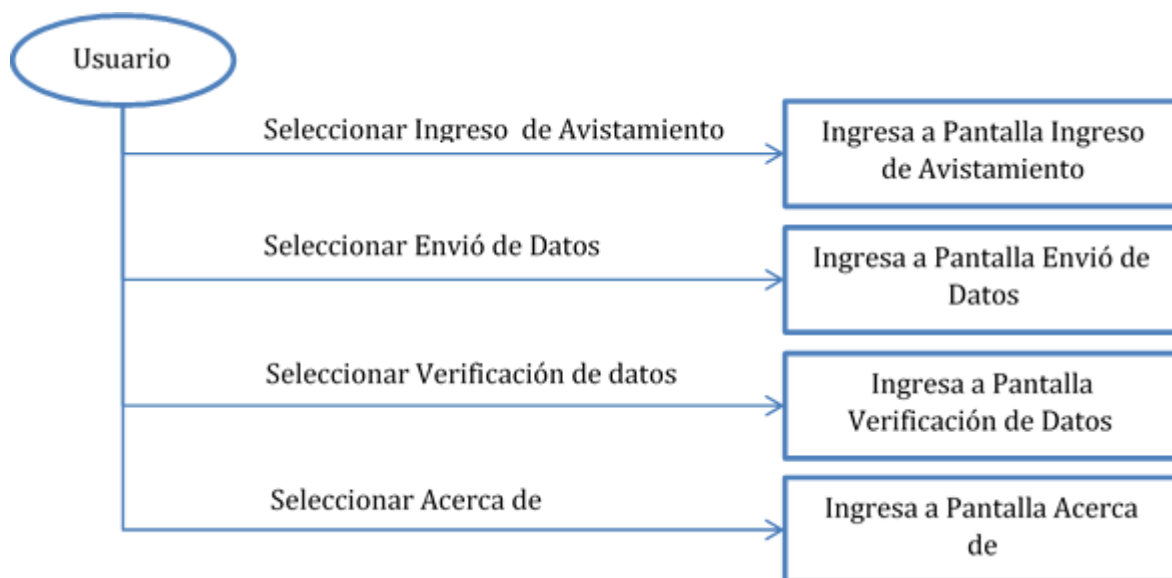


Diagrama 4. Diagrama de interfaces pantalla principal. Fuente: propia.

Descripción:

El usuario tendrá 4 opciones para ingresar a alguna pantalla de opciones de la aplicación.

Descripción paso a paso:

1. El usuario ingresara a la pantalla de Selección de Operaciones.

2. El usuario seleccionara una opción para ingresar a la siguiente pantalla.

6.4.3 Pantalla de ingreso de avistamiento

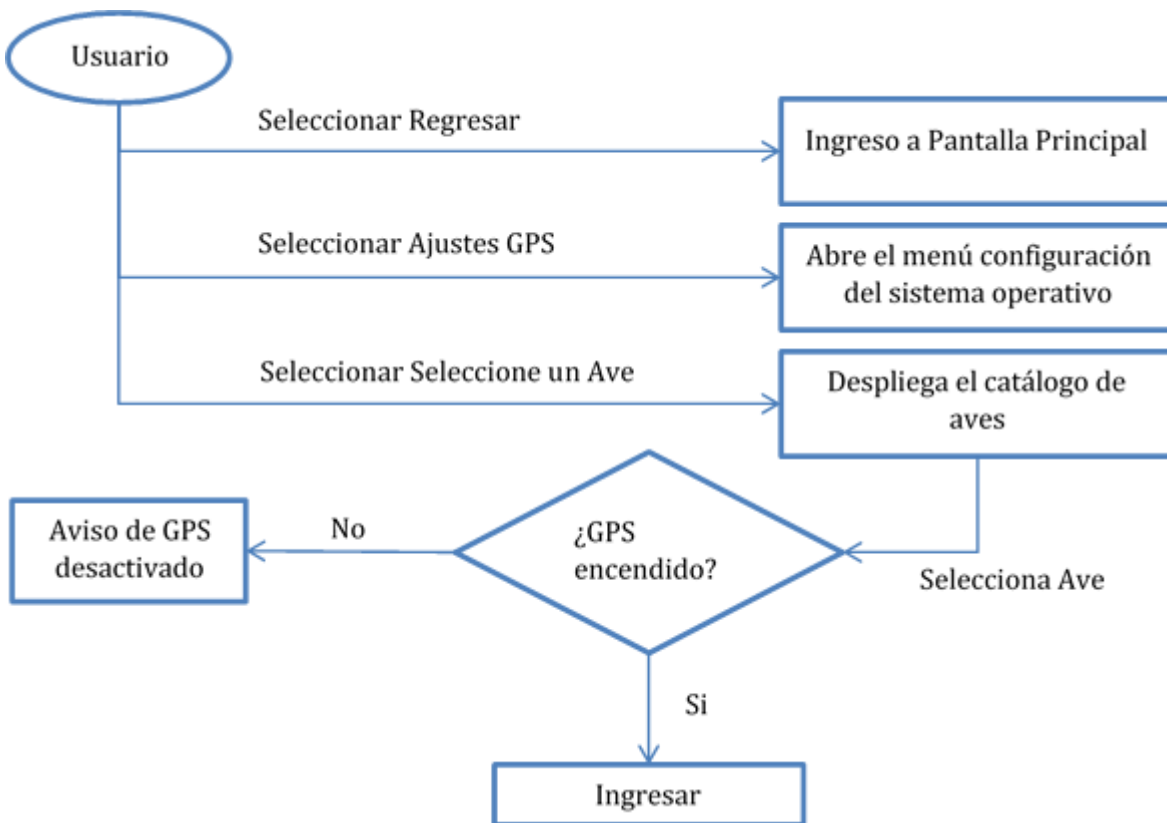


Diagrama 5. Diagrama de interfaces pantalla ingreso de avistamiento. Fuente: propia.

Descripción:

El usuario tendrá la opción de ingresar un ave o regresar al menú superior, sin embargo, en caso de que el GPS no esté encendido, esta opción no estará disponible.

Descripción paso a paso:

1. El usuario selecciona Selecciona un Ave.
2. El usuario selecciona de la lista desplegable una especie de ave.
3. El usuario ve la ubicación e ingresa avistamiento.

En caso de tener GPS apagado, mandara un aviso de GPS desactivado.

6.4.4 Pantalla de envío de datos

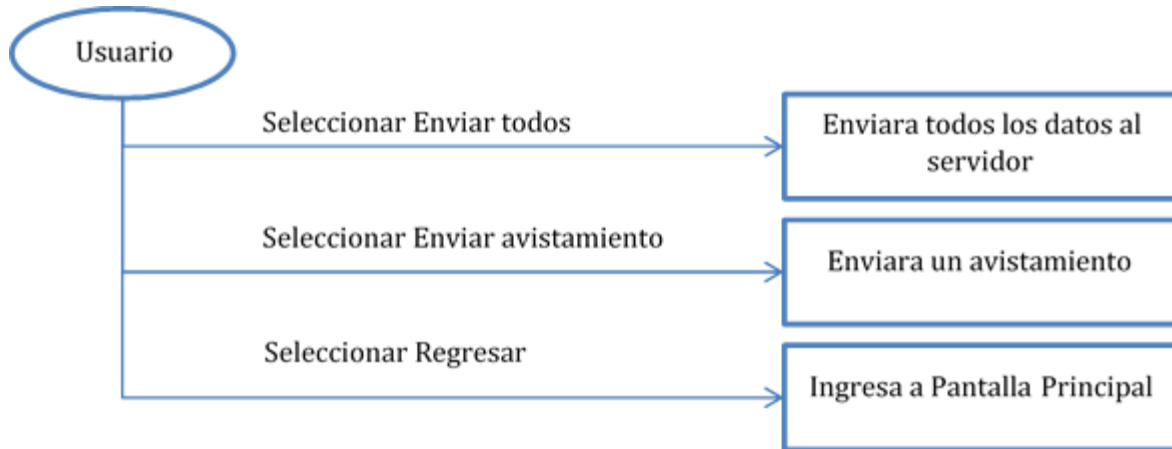


Diagrama 6. Diagrama de interfaces envío de datos. Fuente: propia.

Descripción:

El usuario podrá seleccionar un ave para enviar al servidor o mandar todos los avistamientos.

Descripción paso a paso:

1. El usuario dará clic en el botón enviar avistamiento.

En caso de que el usuario requiera mandar todos

1. El usuario dará clic en el botón Enviar Todos.

6.4.5 Pantalla verificación de datos

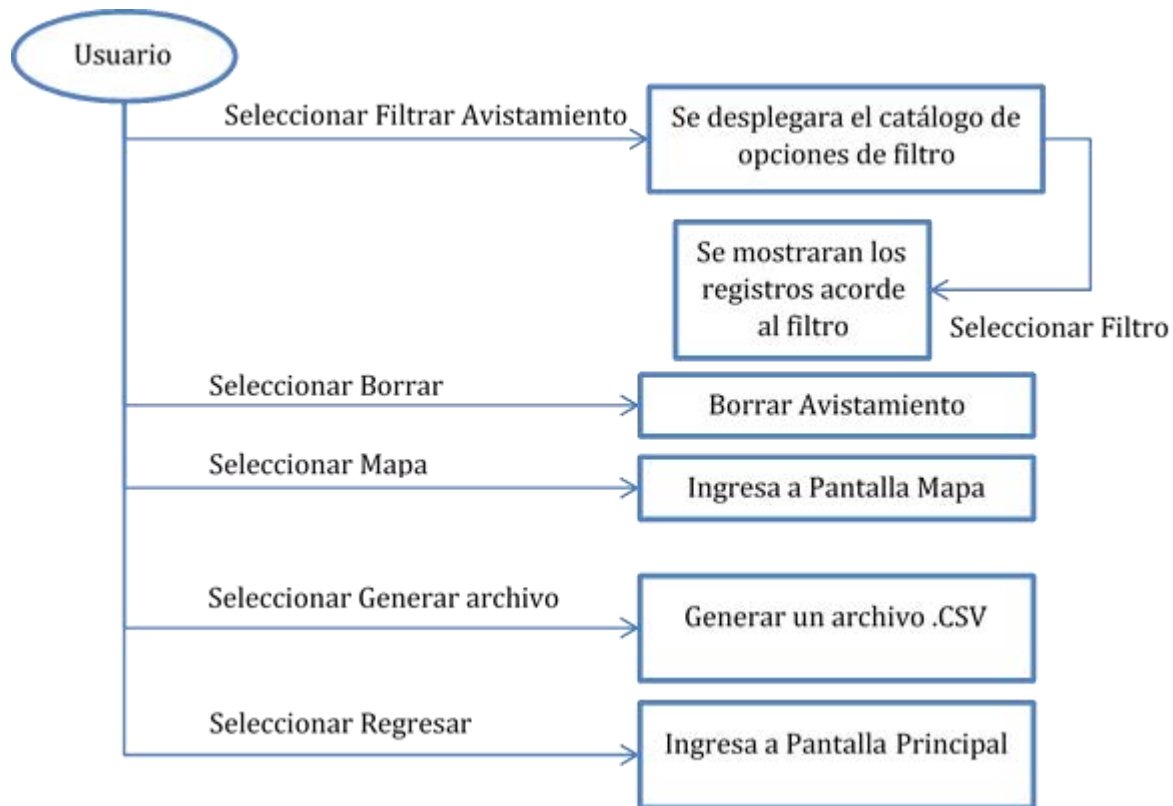


Diagrama 7. Diagrama de interfaces verificación de datos. Fuente: propia.

Descripción:

El usuario observará sus avistamientos y podrá almacenarlos en un archivo para su futuro uso.

Descripción paso a paso:

1. El usuario verá sus avistamientos en la pantalla.
2. El usuario presionara el botón generar archivo CSV. El archivo se generara en la raíz del sistema.

6.4.6 Pantalla mapa

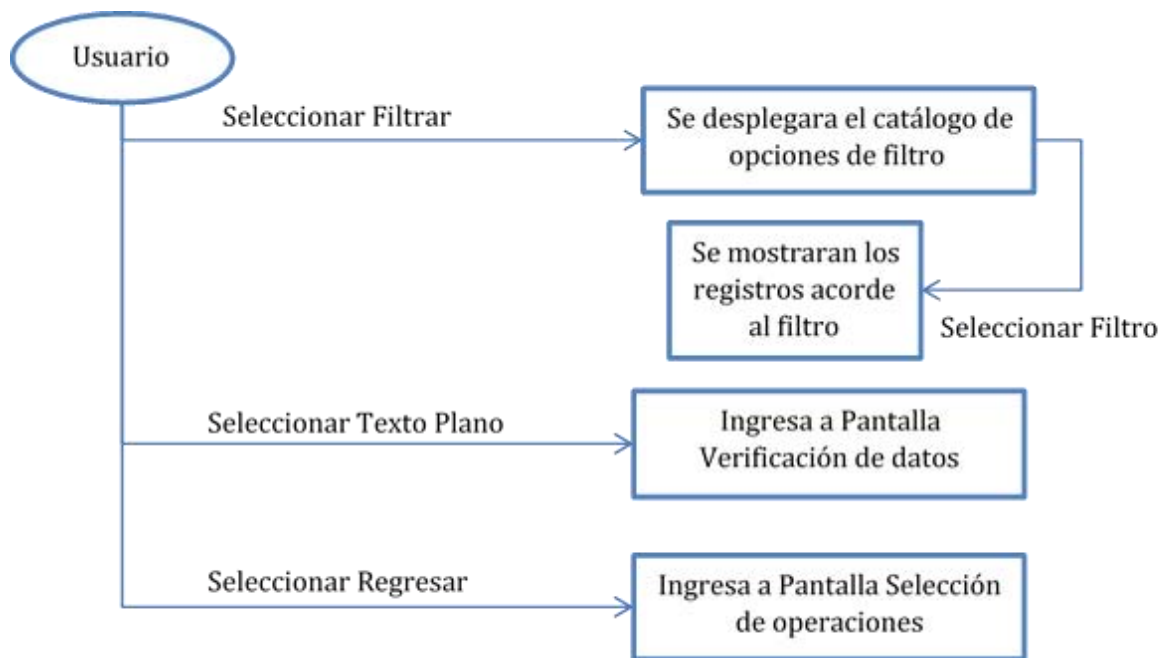


Diagrama 8. Diagrama de interfaces verificación de datos. Fuente: propia.

Descripción:

El usuario observará sus avistamientos en el mapa.

Descripción paso a paso:

1. El usuario verá sus avistamientos en el mapa.

6.4.7 Pantalla Acerca de

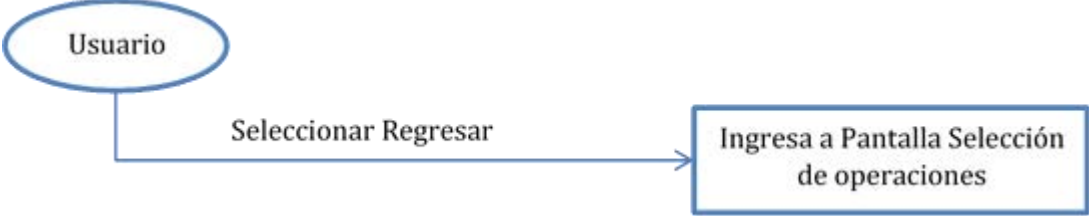


Diagrama 9. Diagrama de Acerca de. Fuente: propia.

Descripción:

Se desplegará en la pantalla los detalles acerca de la aplicación.

IX. Presentación de resultados

Acorde con la documentación y las necesidades del MZFC se tenía como meta una aplicación móvil en conjunto con un servicio web, el cual nos permitiría verificar los datos mandados por los biólogos. Originalmente se tenía la idea de desplegar esta aplicación a nivel mundial dando el servicio como una aplicación gratuita en la Playstore de Google. Sin embargo esta idea fue desechada, ya que el museo comentó que este sistema sería totalmente interno y para servicio exclusivo del museo en primera instancia para realizar las pruebas y luego, se podría colocar a nivel mundial. A la fecha de cierre de esta publicación el MZFC, aún no decide si lanzarlo a nivel mundial o ser de uso exclusivo del MZFC.

Así mismo se tenía pensado una aplicación móvil un poco más sencilla, respecto a datos contenidos en la misma. Acerca de esto, originalmente se consideró que sería solo registrarían longitud y latitud, así como el nombre del ave. Tras una primera junta de presentación del proyecto frente a los encargados y los usuarios principales de éste, se decidió agregar los elementos que se mostraron en desarrollo adicional (altitud, anotación de observaciones, consulta personal de datos, desplegado de datos en mapa), con el cual la aplicación se enriqueció bastante y los usuarios finales y administradores quedaron conformes con él.

Otro detalle que se tuvo que modificar también fue el panel de administrador, ya que originalmente se consideraba sólo entregar la base de datos a los que serían recibidos los datos sin ninguna interfaz gráfica, más que la del manejador de base de datos que ellos usaron. Gracias a la idea planteada por el MZFC esta sección se cambió por completo a un servicio web controlado totalmente por él, con el cual pueden aceptar o rechazar datos en el momento en que ellos así lo deseen, como bloquear y aceptar a algunos usuarios.

Originalmente el servicio web de este sistema se colocó en un servidor privado que el autor de esta tesis rentó. Sin embargo gracias a la buena disposición de la FC y

los resultados tan positivos que el MZFC obtuvo, la UNAM facilitó el uso de uno de sus servidores para la instalación. Actualmente este sistema se encuentra cargado en un servidor de la FC de la UNAM.

Gracias a las pruebas realizadas en campo logramos obtener como resultado que, en comparación con la forma original en la que se hacían observaciones se logró apreciar que los tiempos se redujeron significativamente. Por ejemplo en las pruebas realizadas en Oaxaca, un grupo control realizó las observaciones como normalmente lo hacían previo al sistema y otro realizó las pruebas con el sistema. Fue bastante convincente que el porcentaje de error que se tenía de las observaciones de la forma tradicional presentaba un porcentaje de error alto en comparación con los realizados por el sistema, así como el tiempo en el que tuvo los resultados de estas observaciones el MZFC. El sistema proporcionó la información en 2 días máximo desde el momento de la observación mientras que los realizados de forma tradicional tardaron los 15 días que tardaba toda la observación en campo, así como el regreso a la Ciudad de México. Otro punto importante es que con los datos ya ingresados en sistema ya pudieron ser manipulados al instante para un análisis. A diferencia de los registrados tradicionalmente los cuales además de llegar a la Ciudad de México, se tuvieron que volver a registrar dentro de un sistema que ellos tienen.

Con todo lo anterior se puede concluir que el sistema funciona bien, cumple sus objetivos que se tenían. Así mismo el museo actualmente comienza la transición del modelo tradicional de observaciones a este nuevo poco a poco.

Una de las cosas que logramos apreciar, es que los que más ocupan este sistema son los investigadores que irán a campo y son los que tienen en su mayoría teléfonos inteligentes con sistema Android instalado, sin embargo, aquellos que realizan análisis en el MZFC y que no salen a campo, en su mayoría tienen sistema IOS instalado en su celular.

X. Conclusiones

El objetivo de la presente tesis era reducir el porcentaje de error y ayudar al personal del MZFC, se solicitó desarrollar un sistema con el cual se pueda obtener la localización exacta o muy aproximada de los avistamientos de aves observados en el instante y poder almacenarlas para un futuro análisis de forma rápida y sencilla. Así mismo se lograra reducir los tiempos desde la observación hasta el análisis con el cual el MZFC podrá realizar este proceso más rápidamente y dar un alcance mayor en tema de exactitud avalado por tecnología de última generación.

Con esto de antecedentes se tuvo que seguir una serie de normas de documentación de toma de requerimientos para poder desarrollar un sistema estable sin problemas de desarrollo y que me permitirá llevar a cabo cada una de las necesidades que presentaba el MZFC. Este conocimiento fue adquirido dentro de las aulas de la Facultad de Ingeniería y aplicadas en forma práctica dentro de la tesis, lo cual brindo un gran apoyo para llevar a cabo esta tesis. Es forzoso entender que sin este conocimiento y sin las prácticas que se realizaron en la Facultad, es altamente probable que este proyecto no se hubiera realizado correctamente y los tiempos estimados posiblemente hubieran empeorado o complicado excesivamente el desarrollo.

Al momento del desarrollo, tras realizar la toma de requerimientos, se observó que los avistamientos originalmente eran tardados para registrar y que no se podía realizar tan rápido el análisis por los tiempos que tardaría el observador de ir del lugar donde realizó la anotación hacia donde poder registrar el dato y las locaciones donde se realizan. Existieron algunos problemas al momento del desarrollo ya que uno de los requerimientos fue la altitud, dato que el GPS en los teléfonos inteligentes no da un dato exacto, puesto que varía bastante. Fue por eso que, gracias al apoyo del personal del MZFC, se cambió este requerimiento por un campo de observaciones, para, en caso de que el avistamiento estuviera cerca de algún lugar

donde se pudiera dar cambios muy bruscos de altura (barrancos), se pudiera dar una anotación de donde se realizó el avistamiento.

Tras una serie de pruebas en campo real y anotaciones de parte del MZFC se logró crear un sistema el cual permite realizar avistamientos con alta exactitud y bajo costo y el análisis se podía realizar al instante de la observación o ese mismo día pero más tarde como se hace notar en la presentación de resultados.

Una de las hipótesis era que si se desarrolla un sistema informático para el manejo de información sobre aves y auxilie en el campo para localizar y almacenar en forma aproximada la ubicación, facilitará la administración científica de datos generados por el MZFC. Gracias a los resultados obtenidos en la fase de pruebas, se apreció que en efecto, el sistema generado ayudó en gran medida a localizar las aves de forma precisa y que además les reducía el trabajo en esta parte de inserción de datos, por lo que la hipótesis se comprueba.

La otra hipótesis que se planteo fue que el sistema informático reduciría costos de tiempo, así como facilitar el análisis futuro de estos datos con mayor exactitud a la que se tenía originalmente. Y en efecto, gracias a la tecnología del GPS se logró reducir costos de tiempo, ya que como se presentan en los resultados, en fase de pruebas, el sistema redujo considerablemente este efecto pasando de 2 a 3 semanas a solo un día con lo cual la hipótesis se comprueba dando como resultado que de las dos hipótesis planteadas en esta tesis, ambas son comprobadas.

Como recomendación al cliente se le dio la opción de dar mantenimiento de parte del desarrollador a esta aplicación para darle una evaluación continua más adelante y realizarle cambios si así lo requirieran. Además, se hizo la observación de, que para que el sistema siga funcionando en forma correcta, se debe de realizar un respaldo de información periódicamente, sin embargo, este periodo se dejara a consideración del MZFC y ellos decidirán cuándo realizar este respaldo.

Finalmente se tiene como siguiente paso ampliar esta aplicación a nivel nacional para que cualquier usuario con conocimientos de aves pueda insertar datos hacia las bases de datos que tendrán en su poder los encargados del MZFC, para poder trabajar con una mayor cantidad de datos. Gracias a que se tiene el filtro de usuarios confiables y usuarios no confiables, se podrá llevar un control total de los datos que se reciben y evitar inconsistencia de datos.

Una conclusión que se me hace de vital importancia es que la tesis es un elemento de difusión y divulgación que para ser un egresado de licenciatura debería promoverse más, ya que permite asimilar y tener una visión correcta de cómo difundir conocimientos aprendidos en la carrera, me permitió consolidar y sobre todo validar que son relevantes los conocimientos y que se pueden aplicar, pero también que son una base universal de conocimiento que nos da los elementos de poder aprender más y aplicarlo a nuestros problemas profesionales.

A forma de cierre, afirmo que si se realiza un sistema informático que cubra las necesidades del cliente en forma práctica, de manera legible y con la documentación necesaria, los problemas que se tienen pueden desaparecer o, como en esta tesis, reducir drásticamente el porcentaje de error, así como facilitar las actividades que realiza el cliente. De manera más específica para la tesis se logró el objetivo, y las hipótesis que se plantearon desde un principio fueron comprobadas, como se comentó antes, éste no es ni será la versión definitiva, ya que los tiempos van cambiando y las técnicas de observación irán haciendo lo mismo, por lo que se tendrá que mejorar esta aplicación o en su debido caso desecharla si ya no se ajusta a las épocas futuras, por lo cual se dejó el código fuente para la FC.

Afirmo también la importancia de desarrollar tesis, que en lo particular me permitió pensar más y poder difundir de manera escrita el conocimiento aprendido y aplicado a un problema real, que de otra manera quizás no sería tan eficiente, concluyo además que en ingeniería si no se aplica lo que se aprende, no se hace ingeniería.

XI. Referencias

Adams, P.; Barker, B.; Charles, C. (2004). *Software Requirements Specification*.
Obtenido en octubre de 2013, de la página <http://www.cse.msu.edu/~chengb/RF-491/Papers>

Alicante, U. (2013). *Modelo vista controlador (MVC)*. Obtenido en Julio de 2015, de la pagina <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Bascón, E. (2004). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. Acta Nova; Vol. 2, N° 4, 493-503.

Benslimane, D.; Dustdar, S.; Sheth, A. (2008). *Services Mashups: The New Generation of Web Applications*. IEEE Internet Computing, 10 (5), 13–15,

Bourque, P.; Fairley, R.E. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society.

Carballo, Y. (2005). *Programación Orientada a Objetos (POO)*. Venezuela. Universidad Central de Venezuela.

CONACULTA. (2015). *Museo de Zoología Alfonso L. Herrera*. Obtenido en mayo, 2015, de la página http://sic.conaculta.gob.mx/ficha.php?table=museo&table_id=469

Damián, G. (2009). *Patrones de Diseño, Refactorización y Anti patrones. Ventajas y Desventajas de su Utilización en el Software Orientado a Objetos*.

EBird (2012). *About eBird*. Obtenido en octubre de 2013 de la página <http://ebird.org/content/ebird/about/>

Escalante, P. (1929). *Colección Nacional de Aves*. Obtenido en septiembre, 2013, de la página <http://www.ibiologia.unam.mx/cnav/>

Facultad de Ciencias, Universidad Nacional Autónoma de México, Museo de Zoología "Alfonso L. Herrera". Departamento de Biología Evolutiva. (1978). *Inicio*. Obtenido en septiembre, 2013, de la página <http://osuno.fciencias.unam.mx/laboratorios/Mzoologia/Bienvenida.html>

Facultad de Ciencias, Universidad Nacional Autónoma de México, Museo de Zoología "Alfonso L. Herrera", Departamento de Biología Evolutiva. (1978). *Personal Académico y Curatorial*. Obtenido en septiembre, 2013, de la página http://osuno.fciencias.unam.mx/laboratorios/Mzoologia/AvesMam_MZFC_archivos/page0002.htm

Facultad de Ingeniería, Universidad Nacional Autónoma de México (2008) Plan de estudios de la carrera Ingeniería en Computación. Obtenido en septiembre, 2013, de la página http://www.ingenieria.unam.mx/paginas/Carreras/planes2010/ingComputo_Plan.htm

Federal Aviation Administration (FAA), *Press Release FAA Announces Major Milestone for Wide Area Augmentation System (WAAS)*. Marzo 24, 2006.

Gill, F. y Donsker D. (2015). IOC World Bird List (v 5.2). Obtenido en mayo, 2015, de la página <http://www.worldbirdnames.org/>

Gómez de Silva, H. y Alvarado, E. (2010). Breve historia de la observación de aves en México en el siglo XX y principios del siglo XXI. *HUITZIL*, 11(1), 9-20.

Guerrero, M. (2004). Guía básica para la observación de aves. *Boletín Científico Centro de Museos Museo de Historia Natural*, 8, 107-112.

Ibird. Obtenido en septiembre, 2013, de la página <http://ibird.com/>

Lind, J., Jakobsson, S. y Kullberg, C. (2010) *Current Ornithology*. Illinois, Estados Unidos. Springer.

Laboratorio de Ornitología de Cornell y Sociedad Nacional de Audubon. (2002). *About eBird*. Obtenido en septiembre, 2013, de la página <http://ebird.org/content/ebird/about/>

Laboratorio de Ornitología de Cornell y Sociedad Nacional de Audubon. (2002). *eBird*. Obtenido en septiembre, 2013, de la página <http://ebird.org/content/ebird/>

Márquez, M. (2014). *Programación de sockets en C de Unix/Linux*. Obtenido en Julio, 2015, de la página <http://teoriapa1112.blogspot.mx/2011/11/programacion-de-sockets-en-c-de.html>

National Audubon Society (2009). *History*. Obtenido en octubre de 2013 de la página <http://www.audubon.org/bird/cbc/history.html>

Navarro-Sigüenza, A.G., Ortiz-Pulido, R. y Peterson, A.T. (2008). Un panorama breve de la historia de la ornitología Mexicana. *Ornitología Neotropical*, 19, 367-380.

Pressman, Roger (2010). *Software Engineering: A Practitioner's Approach*. Boston, Estados Unidos.

Sociedad Nacional de Audubon. (2009). *Christmas Bird Count*. Obtenido en septiembre, 2013, de la página <http://birds.audubon.org/christmas-bird-count>.

Korth, H.; Silberschatz, A. y Sudarshan S. (2002). *Fundamentos de bases de datos* (4.^a ed.). México. Mc Graw Hill.

Taylor, S. (1987). *Introducción a los métodos cualitativos de investigación*.

Obtenido en mayo 2015 de la página

<http://201.147.150.252:8080/xmlui/handle/123456789/1216>

Ullman, J. (1997). Primer curso de sistemas de bases de datos, Nueva York, Estados Unidos

XII. Anexos

ANEXO 1 CÓDIGO FUENTE

```
<?xml version="1.0" encoding="utf-8"?>
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
xmlns:s="library://ns.adobe.com/flex/spark" title="AppVE" viewActivate="init(event)">

<fx:Script>
<![CDATA[
import mx.core.FlexGlobals;
import spark.events.ViewNavigatorEvent;

private var crearUsuarioStatement:SQLStatement;
private var cargarUsuarioStatement:SQLStatement;

private function generateRandomString(strlen:Number):String{
var chars:String = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
var num_chars:Number = chars.length - 1;
var randomChar:String = "";

for (var i:Number = 0; i < strlen; i++){
randomChar += chars.charAt(Math.floor(Math.random() * num_chars));
}
return randomChar;
}

protected function goAboutView(event:MouseEvent):void
{
// Ir al View de ACerca de...
navigator.pushView(AboutView);
}

protected function goNuevoAvistamiento(event:MouseEvent):void
{
// Ir al View de nuevo avistamiento
navigator.pushView(IngresarAvistamientoView);
}

protected function goVerificacionDatos(event:MouseEvent):void
{
// Ir al View de Verificacion de Datos
navigator.pushView(VerificacionDatosView);
}

protected function goEnvioDatos(event:MouseEvent):void
```

```

{
// Ir al View Envio de Datos
navigator.pushView(EnviarDatosView);
}

protected function init(event:ViewNavigatorEvent):void
{
// Al entrar al menu principal verificar usuario
verificarUsuario();
}

private function verificarUsuario():void{
cargarUsuarioStatement = new SQLStatement();
cargarUsuarioStatement.sqlConnection = FlexGlobals.topLevelApplication.conn;
cargarUsuarioStatement.addEventListener(SQLEvent.RESULT, cargarUsuarioResult);
cargarUsuarioStatement.addEventListener(SQLErrorEvent.ERROR, cargarUsuarioError);
var cargarUsuarioQuery:String = "SELECT * FROM usuario";

cargarUsuarioStatement.text = cargarUsuarioQuery;
cargarUsuarioStatement.execute();
}

private function cargarUsuarioResult(event:SQLEvent):void{
var result:SQLResult = cargarUsuarioStatement.getResult();

//Revisamos la informacion del primer usuario que se encuentra en el indice 0, si el usuario no existe lo creamos.
if(result.data != null){
//trace(result.data[0].user_id)
FlexGlobals.topLevelApplication.usuario = result.data[0].user_id;
FlexGlobals.topLevelApplication.usuarioAlias = result.data[0].alias;
FlexGlobals.topLevelApplication.usuarioStatus = result.data[0].status;
} else {
crearUsuario();
}
}

private function cargarUsuarioError(error:SQLErrorEvent):void{
trace(error.errorID);
}

private function crearUsuario():void{
crearUsuarioStatement = new SQLStatement();
crearUsuarioStatement.sqlConnection = FlexGlobals.topLevelApplication.conn;
crearUsuarioStatement.addEventListener(SQLEvent.RESULT, crearUsuarioResult);
crearUsuarioStatement.addEventListener(SQLErrorEvent.ERROR, crearUsuarioError);
var crearUsuarioQuery:String = "INSERT INTO usuario (user_id, alias, status) VALUES
("+generateRandomString(20)+"','"+generateRandomString(20)+"", 3)";

```

```

crearUsuarioStatement.text = crearUsuarioQuery;
crearUsuarioStatement.execute();
}

```

```

private function crearUsuarioResult(event:SQLEvent):void{
verificarUsuario();
}

```

```

private function crearUsuarioError(error:SQLErrorEvent):void{
trace(error.errorID);
}

```

```

]]>
</fx:Script>

```

```

<fx:Declarations>
<!-- Place non-visual elements (e.g., services, value objects) here -->
</fx:Declarations>

```

```

<s:actionContent>
<s:Button click="goAboutView(event)">
<s:icon>
<s:MultiDPIBitmapSource source160dpi="@Embed('assets/icons/160/2_action_about.png')"
source240dpi="@Embed('assets/icons/240/2_action_about.png')"
source320dpi="@Embed('assets/icons/320/2_action_about.png')"/>
</s:icon>
</s:Button>
</s:actionContent>

```

```

<s:VGroup id="mainGroup" width="100%" height="100%" gap="0" requestedMaxRowCount="3"
rowHeight="{mainGroup.height/3}">

```

```

<s:Group width="100%" height="33%" click="goNuevoAvistamiento(event)">
<s:Rect width="100%" height="100%">
<s:fill>
<s:LinearGradient rotation="90">
<s:entries>
<s:GradientEntry color="#FFAE18"/>
<s:GradientEntry color="#FF8A00"/>

```

```

</s:entries>
</s:LinearGradient>
</s:fill>
</s:Rect>
<s:BitmapImage source="@Embed('assets/icons/add.png')" left="20" verticalCenter="0" smooth="true" width="50" height="50"/>
<s:Label text="Ingreso de Avistamiento" styleName="myLabel" left="100" right="80" verticalCenter="0"/>
<s:BitmapImage source="@Embed('assets/rightarrow.png')" right="5" verticalCenter="0" smooth="true" width="40" height="40"/>
</s:Group>

```

```

<s:Group width="100%" height="33%" click="goEnvioDatos(event)">
<s:Rect width="100%" height="100%">
<s:fill>
<s:LinearGradient rotation="90">
<s:entries>
<s:GradientEntry color="#92C500"/>
<s:GradientEntry color="#669900"/>

</s:entries>
</s:LinearGradient>
</s:fill>
</s:Rect>
<s:BitmapImage source="@Embed('assets/icons/send.png')" left="20" verticalCenter="0" smooth="true" width="50"
height="50"/>
<s:Label text="Envio de Datos" styleName="myLabel" left="100" right="80" verticalCenter="0"/>
<s:BitmapImage source="@Embed('assets/rightarrow.png')" right="5" verticalCenter="0" smooth="true" width="40" height="40"/>
</s:Group>

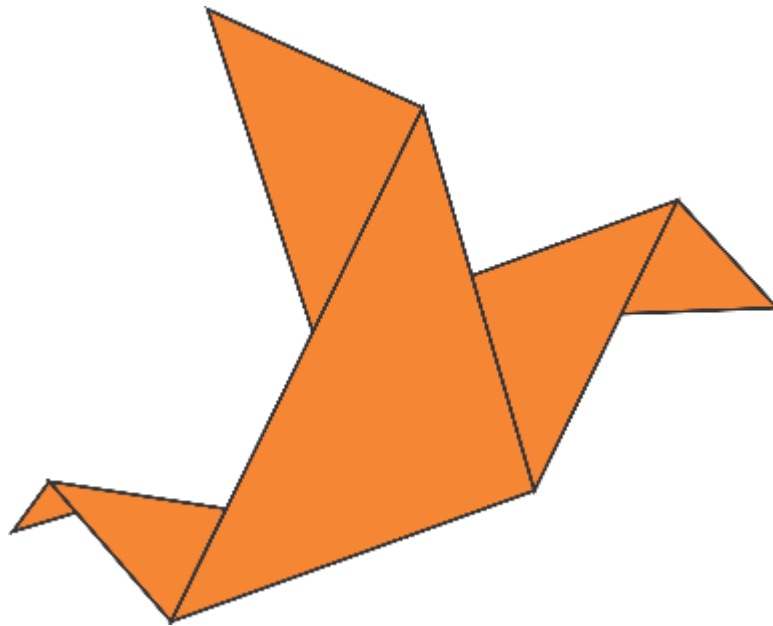
<s:Group click="goVerificacionDatos(event)" width="100%" height="33%">
<s:Rect width="100%" height="100%">
<s:fill>
<s:LinearGradient rotation="90">
<s:entries>
<s:GradientEntry color="#2CB1E1"/>
<s:GradientEntry color="#12A0CF"/>

</s:entries>
</s:LinearGradient>
</s:fill>
</s:Rect>
<s:BitmapImage source="@Embed('assets/icons/check.png')" left="20" verticalCenter="0" smooth="true" width="50"
height="50"/>
<s:Label text="Verificacion de Datos" styleName="myLabel" left="100" right="80" verticalCenter="0"/>
<s:BitmapImage source="@Embed('assets/rightarrow.png')" right="5" verticalCenter="0" smooth="true" width="40" height="40"/>
</s:Group>
</s:VGroup>
</s:View>

```

ANEXO 2 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

PARA EL SISTEMA



AppVe

1. Introducción

Las funciones y las características que debe cumplir el sistema AppVe tomando como base el estándar de la IEEE 830, se describen a continuación para tener un punto de referencia entre las características necesarias y las soluciones que en este caso es crear un sistema para capturar avistamientos y administrarlos bajo el control del MZFC.

1.1 Propósito

El propósito de este documento es dar una descripción de los requerimientos, limitaciones e interacciones del sistema AppVe en su primera versión. El sistema AppVe permitirá a los usuarios del MZFC ingresar avistamientos de aves en una aplicación móvil para después poder ser administrados por el MZFC para un análisis subsecuente.

1.2 Convenciones

Administrador: Usuario con funciones administrativas dentro del sistema.

Base de datos: Conjunto de toda la información que se maneja en el sistema.

GPS: Sistema de localización satelital

MZFC: Museo de Zoología Alfonso L. Herrera de la Facultad de Ciencias.

SRS: Especificación de requerimientos de software, Software Requirements Specification en inglés

UNAM: Universidad Nacional Autónoma de México.

Usuario: Aquella persona que interactúa con el sistema.

WiFi: Mecanismo de conexión inalámbrica para la transferencia de datos.

1.3 Audiencia dirigida y sugerencia de lectura.

El sistema será dirigido para el personal que labora en el MZFC, sin embargo el alcance del mismo se puede dar a cualquier individuo con conocimiento de reconocimiento de aves por vista.

1.4 Alcance del producto

AppVe es un sistema que permite la localización de aves de manera exacta basados en los conocimientos del que registra el avistamiento. Así mismo permite generar archivos internos dentro del sistema los cuales el usuario podrá consultar para un análisis propio. Adicionalmente se tiene un sistema de servicio web el cual permite al administrador validar la información enviada por cada uno de los usuarios y con el cual se generara una base de datos la cual podrá ser consultada al instante y en cualquier momento.

1.5 Referencias

[1] Institute of Electrical and Electronics Engineers (1998, Junio, 25) IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998. Nueva York, Estados Unidos: The institute of Electrical and Electronics Engineers, Inc.

2. Descripción general

En la actualidad el MZFC realiza avistamientos periódicos hechos por el personal que labora en este lugar, sin embargo se ha observado que muchas de sus observaciones han tenido una tendencia al error ya que no todo el personal que realiza avistamientos cuenta con un aparato para dar un cálculo aproximado o exacto del lugar. Así mismo al momento de tener estas observaciones, pasa mucho tiempo entre que se realiza la observación y en el momento en que por fin se da el análisis, ya que se tienen que trasladar desde el lugar de la observación hasta el MZFC para la captura de los datos. Es por eso que tomando como base esta problemática el sistema AppVe tiene como fin resolver cada uno de estos problemas usando una herramienta que es muy común, un celular inteligente mediante el cual se podrán hacer observaciones con coordenadas generadas por GPS, transferencia de archivos mediante WiFi y almacenamiento de datos remoto.

2.1 Perspectiva del producto

El sistema AppVe permitirá a sus usuarios registrar los avistamientos junto con sus coordenadas de forma automática basada en el criterio del usuario que realiza el avistamiento. Una vez que el usuario está conforme con este dato, mandara esta información al servidor localizado en el MZFC para que ahí el administrador de su visto bueno para comenzar con un análisis.

2.2 Funciones del Producto

El sistema AppVe permitirá a sus usuarios:

- Registrar avistamientos
- Ver los avistamientos realizados hasta el momento
- Generar un archivo para consultas de avistamientos realizados
- Enviar los datos de avistamientos al MZFC para la revisión del administrador.

El sistema AppVe permitirá a sus administradores:

- Consultas los avistamientos nuevos y antiguos del sistema
- Extraer toda la información para un análisis posterior
- Bloquear usuarios que no manden datos correctos
- Permitir a usuarios confiables darles la oportunidad de personalizar su nombre de usuario

2.3 Características de los usuarios

Los usuarios del sistema AppVe deben de contar con los conocimientos básicos de reconocimiento de aves, se debe de contar con un teléfono inteligente con sistema operativo Android instalado y sistema de GPS integrado y encendido al momento de abrir la aplicación.

2.4 Restricciones

El sistema en su versión portable debe de estar en un área abierto ya que el GPS no funciona en lugares muy cerrados como túneles y cuevas. Para el servidor donde se alojaran los datos se debe tener una conexión permanente a internet y el servidor debe de contar con Apache HTTP Server instalado y debidamente configurado.

2.5 Suposiciones y dependientes

Para un funcionamiento correcto del sistema AppVe, la parte móvil requiere las siguientes características:

- Android 2.3 o superior instalado
- Contar con GPS
- Tener permisos de instalación de fuentes desconocidas de aplicaciones sobre el mismo.

Para el funcionamiento del sistema AppVe, el servidor deberá contener los siguientes requisitos:

- Permanente conexión a internet
- Apache HTTP Server instalado

2.6 Requerimientos futuros

En el debido caso de querer liberar este sistema para todo el mundo, se necesitara una licencia para publicar aplicaciones móviles en la tienda Playstore de Google.

2.7 Entorno operativo

El sistema AppVe en la parte móvil, solo se podrá correr en aquellos teléfonos inteligentes con Android instalado. Para el panel administrativo no importa el sistema operativo ya que este solo dependerá enteramente del navegador web y la conexión a internet.

3. Requerimientos de interfaz externa

3.1. Interfaz de Usuario

El usuario deberá de contar con GPS encendido, de lo contrario no podrá registrar ninguna avistamiento. Al ingresar al sistema se presentaran 3 opciones: Ingreso de avistamiento, Envío de datos y verificación de datos. Dependiendo de la opción que se escoja es la acción que se procederá. Dichas acciones están explicadas en la sección 4 características del sistema.

3.1.1. Ingreso de avistamiento

Dentro de esta opción el usuario podrá ingresar un avistamiento de ave seleccionando la especie, en caso de existir algún comentario adicional, se podrá insertar en un campo de texto. En caso de no tener el GPS encendido, la aplicación mandara a la configuración del celular para poder encenderlo, se cerrara la aplicación y se dará por finalizada la sesión hasta abrir de nuevo la aplicación.

3.1.2. Envío de Datos

Dentro de esta opción solo se mostrara aquellos registros que falten de ser enviados a el MZFC y permitirá mandar datos uno a uno o todos en conjunto. En caso de que el administrador haya considerado que es un usuario de confianza, antes de enviar un dato se mostrara una nueva pantalla el cual permitirá al usuario ingresar su nombre para su fácil reconocimiento por el administrador.

3.1.3. Verificación de datos

Dentro de esta opción el usuario podrá ver sus avistamientos, filtrarlos por fecha de registro y podrán ser exportados a un archivo Excel para su análisis. Dicho archivo Excel se localizara en la carpeta raíz de la memoria.

3.2. Interfaz de Administrador

El usuario administrador tendrá un panel general el cual se dividirá en 5 pestañas que serán descritas a continuación.

3.2.1. Registros Nuevos

Mostrará en una tabla los registros nuevos que se han agregado y que falta por validar. Se podrán validar uno a uno o todos en conjunto. Se mostrara también un mapa para visualizar geográficamente donde fue realizada esa observación. En caso de tener un comentario, se mostrara al administrador.

3.2.2. Registros por Especie

Mostrará una tabla con todas las especies registradas, ya sea que hayan sido validadas o no, diferenciando claramente entre aquellas validadas y aquellas que no. Se mostrara un mapa el cual visualizara todos los registros hechos de esa especie.

En caso de existir un comentario acerca de un avistamiento en particular, se mostrara al administrador.

3.2.3. Registros por Usuario

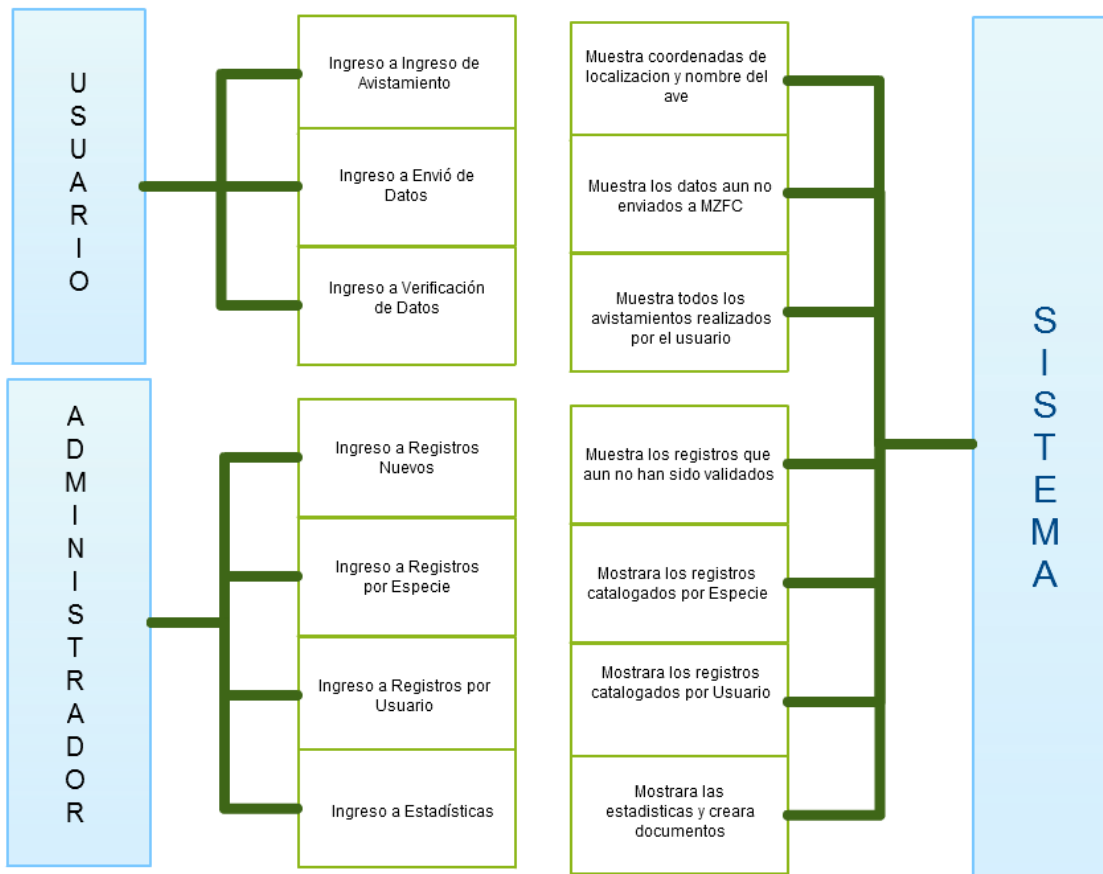
Mostrará una tabla con todas los registros hechos por usuario, ya sea que hayan sido validadas o no, diferenciando claramente entre aquellas validadas y aquellas que no. Se mostrara un mapa el cual visualizara todos los registros hechos por ese usuario. En caso de existir un comentario acerca de un avistamiento en particular, se mostrara al administrador.

3.2.4. Administrar Usuarios

Mostrará el nombre actual del usuario y el nombre valido hecho por el usuario en caso de que el administrador haya considerado a este usuario de confianza. Así mismo el administrador podrá nombrar usuarios de confianza para que su nombre aparezca en la base de datos y se reconozca más fácilmente.

3.2.5. Estadísticas

Mostrará estadísticas de los registros verificados y permitirá exportar los datos a un archivo Excel con diferentes parámetros.



3.3 Interfaz de Hardware

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del Usuario:

- Teléfono inteligente con GPS

De parte del Administrador

- Equipo con conexión a internet

3.4 Interfaz de Software

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del Usuario:

- Celular con sistema operativo Android
- 50 Mb de espacio disponible en memoria

De parte del Administrador

- Equipo con navegador web con Adobe Flash instalado

3.5 Interfaz de comunicación

Se utilizara el protocolo TCP/IP para conectar la aplicación móvil con el sistema en línea.

4. Características del sistema

El sistema permitirá al usuario ingresar avistamientos con precisión con ayuda del GPS, validar dicha información, envió de datos al MZFC, ver histórico de registros en mapa y generación de documento acerca de sus propias observaciones. En cuanto al administrador del sistema podrá observar datos nuevos, validar información, controlar el ingreso de datos por usuario y recopilar toda la información en una sola base.

4.1. Funciones de usuario

4.1.1. Ingreso de avistamiento

El sistema le proporcionara al usuario una pantalla mediante la cual el podrá seleccionar del catálogo de aves alguna, así mismo, en caso de tener alguna observación adicional al avistamiento, se podrá realizar anotaciones sobre la misma.

4.1.2. Envió de datos

El sistema mostrara aquellos datos que aún no han sido enviados y permitirá la opción de enviar datos uno a uno o todos. Cuando el administrador da de alta al usuario como usuario confiable, este recibirá una pantalla nueva que le permitirá ingresar su nombre de usuario

4.1.3. Verificación de datos

El sistema mostrara en forma de lista aquellos datos que han sido enviados y aquellos que no, en conjunto con la fecha y su nombre, adicional a esto, se tendrá la opción de un mapa el cual nos mostraran los avistamientos del usuario. También se tendrá la opción de almacenar todos los datos en un archivo Excel para su análisis futuro.

4.2. Funciones de administrador

4.2.1. Registros nuevos

El sistema mostrara una pantalla en la que estarán los avistamientos sin registrar. Dichos avistamientos se podrán validar uno a uno o todos. En caso de que se tenga un comentario sobre ese avistamiento, el sistema mostrara el comentario de ese avistamiento. También se incluye un mapa el cual mostrara la localización del avistamiento.

4.2.2. Registros por especie

El sistema mostrara una pantalla con los avistamientos clasificados por especie. Dichos avistamientos pueden estar o no validados, pero se diferenciarian entre ellos. Así mismo se podrá ver los comentarios de cada uno de los avistamientos registrados. Finalmente se agrega un mapa para que por especie se muestren todos los avistamientos en el mapa

4.2.2. Registros por usuario

El sistema mostrara una pantalla con los avistamientos clasificados por usuario. Dichos avistamientos pueden estar o no validados, pero se diferenciarian entre ellos. Así mismo se podrá ver los comentarios de cada uno de los avistamientos registrados. Finalmente se agrega un mapa para que por usuario se muestren todos los avistamientos en el mapa.

4.2.3. Administración de usuarios

El sistema mostrara una pantalla donde se desplegaran las opciones del control de usuario comenzando por aprobar a usuarios confiables, los cuales podrán agregar su nombre para que el administrador pueda diferenciarlos de los otros usuarios. Así mismo se tiene la opción de bloqueo de usuarios, con el fin de que estos usuarios ya no puedan ingresar más datos basura.

4.2.4. Estadísticas

El sistema mostrara distintas graficas acorde a datos verificados. Dentro de esta misma opción tenemos la exportación de datos en forma de Excel para un análisis futuro.

5. Otros requerimientos no funcionales

5.1. Requerimientos de Desempeño (Performance Requirement)

El sistema se lograra hacer lo suficientemente ligero como para que cualquier celular con sistema operativo Android pueda correrlo.

5.2. Requerimientos de seguridad (Física)

El servidor del sistema debe contar con instalaciones adecuadas: sin polvo, sin cambios bruscos de voltaje y con nula o poca humedad.

5.3. Requerimientos de seguridad

El sistema no maneja datos de información sensible para el usuario, sin embargo todos los avistamientos al tener que ser verídicos, se pide una seguridad al menos media o alta para resguardar todos estos datos así como generación de respaldos periódicos.

5.4. Atributos de Calidad

El sistema cumple con los siguientes atributos de calidad.

- **Confiabilidad**

El sistema estará siempre funcionando si se sigue correctamente los requerimientos básicos de hardware.

- **Robustez**

Al estar compuesto por componentes de uso amplio (Flex y MySQL) se garantiza este punto.

- **Disponibilidad**

Si el servidor se mantiene en línea todo el tiempo, este sistema se podrá consultar en cualquier momento. Si por alguna situación el servidor se apaga, el sistema funcionara independiente en la aplicación móvil para el usuario.

- **Mantenibilidad**

La actualización del sistema y el cuidado de las bases de datos así como de la aplicación correrán a cargo del administrador del MZFC una vez entregado el sistema.

- **Usabilidad**

El sistema será lo suficientemente intuitivo y simple para que cualquier tipo de usuario pueda usarlo.

- **Portabilidad**

Gracias a que el sistema administrativo no depende del sistema operativo, cumplimos con este atributo.

5.5 Esquema de Negocios (Business Rules)

El presente documento comprende la primera parte del sistema el cual se basa en los requerimientos tomados por el cliente (MZFC). El mismo se acatara a la norma de aceptar y respetar cada uno de los requerimientos sin cambio para poder realizar el sistema de manera correcta y sin imprevistos.

En caso de que el cliente presente algún cambio, será necesario presentar con tiempo un cambio de requerimientos el cual será evaluado y vuelto a considerar en tiempos y costos que estarán señalados en el SRS, documento que se anexa también.

A continuación se presenta el plan maestro de desarrollo del sistema.



Aprobación de documento

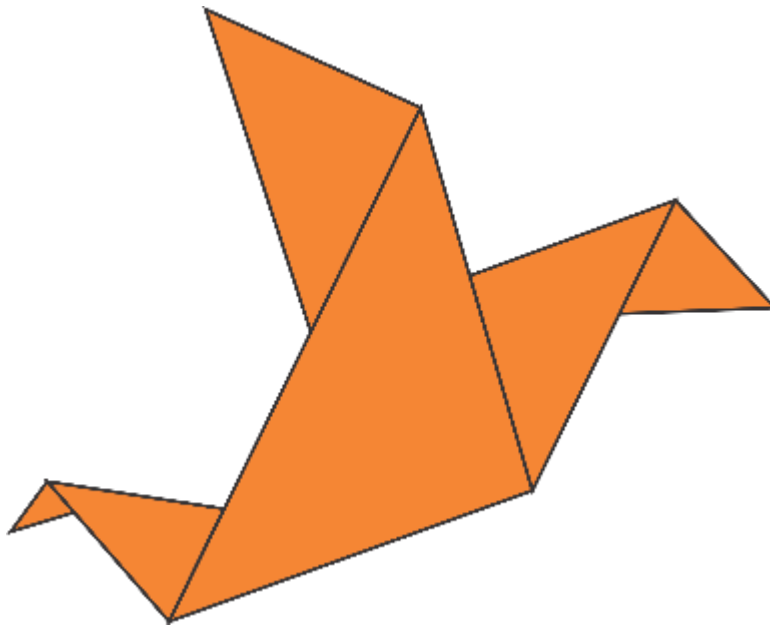
El siguiente documento y los requerimientos del sistema AppVe han sido aprobados por:

Alejandro Gordillo Martínez
Maestro en Ciencias
Administrador de datos de MZFC

Cerón Guevara Héctor
Desarrollador

ANEXO 3 DESCRIPCIÓN DEL DISEÑO DE SOFTWARE

PARA EL SISTEMA



AppVe

1. Introducción

Las funciones y las características que debe cumplir el sistema AppVe tomando como base el estándar de la IEEE 1016, se especifican a continuación para tener un punto de referencia de cómo se construirá el sistema y se obtenga el objetivo de realizar avistamientos de forma exacta con una fácil forma de almacenamiento para futuros análisis.

1.1 Propósito

El propósito de este documento es dar una descripción detallada del diseño total del sistema de forma tal que el desarrollo del software puede llevarse a cabo sabiendo todo lo que se necesita y se espera construir. El sistema AppVe permitirá a los usuarios registrar avistamientos de forma exacta o muy aproximada y al administrador controlar los datos para un futuro análisis.

1.2 Convenciones

Administrador: Usuario con funciones administrativas dentro del sistema.

Base de datos: Conjunto de toda la información que se maneja en el sistema.

GPS: Sistema de localización satelital

MZFC: Museo de Zoología Alfonso L. Herrera de la Facultad de Ciencias.

SDD: Descripción de diseño por software, Software Design Description en ingles

SRS: Especificación de requerimientos de software, Software Requirements Specification en ingles

UNAM: Universidad Nacional Autónoma de México.

Usuario: Aquella persona que interactúa con el sistema.

WiFi: Mecanismo de conexión inalámbrica para la transferencia de datos.

1.3 Audiencia dirigida y sugerencia de lectura.

El sistema será dirigido para el personal que labora en el MZFC, sin embargo el alcance del mismo se puede dar a cualquier individuo con conocimiento de reconocimiento de aves por vista.

1.4 Alcance del producto

AppVe es un sistema que permite la localización de aves de manera exacta basados en los conocimientos del que registra el avistamiento. Así mismo permite generar archivos internos dentro del sistema los cuales el usuario podrá consultar para un análisis propio. Adicionalmente se tiene un sistema de servicio web el cual permite al administrador validar la información enviada por cada uno de los usuarios y con el cual se generara una base de datos la cual podrá ser consultada al instante y en cualquier momento.

1.5 Referencias

[1] Institute of Electrical and Electronics Engineers (2009, Marzo, 19) *IEEE Standard for Information Technology –Systems Design- Software Design Descriptions*. IEEE Std 1016-2009. Nueva York, Estados Unidos: The institute of Electrical and Electronics Engineers, Inc.

2. Propósito general

2.1 Perspectiva del producto

El sistema AppVe permitirá a sus usuarios registrar los avistamientos junto con sus coordenadas de forma automática basada en el criterio del usuario que realiza el avistamiento. Una vez que el usuario está conforme con este dato, mandara esta información al servidor localizado en el MZFC para que ahí el administrador de su visto bueno para comenzar con un análisis.

2.2 Funciones del Producto

El sistema AppVe permitirá a sus usuarios:

- Registrar avistamientos
- Ver los avistamientos realizados hasta el momento
- Generar un archivo para consultas de avistamientos realizados
- Enviar los datos de avistamientos al MZFC para la revisión del administrador.

El sistema AppVe permitirá a sus administradores:

- Consultas los avistamientos nuevos y antiguos del sistema
- Extraer toda la información para un análisis posterior

- Bloquear usuarios que no manden datos correctos
- Permitir a usuarios confiables darles la oportunidad de personalizar su nombre de usuario

2.3 Características de los usuarios

Los usuarios del sistema AppVe deben de contar con los conocimientos básicos de reconocimiento de aves, se debe de contar con un teléfono inteligente con sistema operativo Android instalado y sistema de GPS integrado y encendido al momento de abrir la aplicación.

2.4 Restricciones

El sistema en su versión portable debe de estar en un área abierto ya que el GPS no funciona en lugares muy cerrados como túneles y cuevas. Para el servidor donde se alojaran los datos se debe tener una conexión permanente a internet y el servidor debe de contar con Apache HTTP server instalado y debidamente configurado.

2.5 Suposiciones y dependientes

Para un funcionamiento correcto del sistema AppVe, la parte móvil requiere las siguientes características:

- Android 2.3 o superior instalado
- Contar con GPS
- Tener permisos de instalación de fuentes desconocidas de aplicaciones sobre el mismo.

Para el funcionamiento del sistema AppVe, el servidor deberá contener los siguientes requisitos:

- Permanente conexión a internet
- Apache HTTP Server instalado

2.6 Requerimientos futuros

En el debido caso de querer liberar este sistema para todo el mundo, se necesitara una licencia para publicar aplicaciones móviles en la tienda Playstore de Google.

2.7 Entorno operativo

El sistema AppVe en la parte móvil, solo se podrá correr en aquellos teléfonos inteligentes con Android instalado. Para el panel administrativo no importa el sistema operativo ya que este solo dependerá enteramente del navegador web y la conexión a internet.

3. Requerimientos de interfaz externa

3.1 Interfaz de Usuario

El usuario deberá tener instalado en su teléfono inteligente con sistema Android la aplicación móvil para poder interactuar con el sistema. Al ingresar al sistema podrá realizar alguna de las siguientes opciones:

-Ingreso de Avistamiento: Al usuario se le mostrara un catálogo con el nombre de las aves. El usuario tendrá que seleccionar el nombre de esta y después dará clic en ingresar avistamiento para registrarlo. Las coordenadas y los registros quedaran grabados para poder ser envíos posteriormente.

-Envío de Datos: El usuario podrá enviar datos que ya han sido registrados previamente al servidor ubicado en el MZFC para su futuro análisis.

-Verificación de Datos: El usuario podrá validar sus datos en forma gráfica así como visualmente a través de un mapa. Adicionalmente a esto se podrán extraer los datos particulares de cada usuario a través de un archivo el cual se generara tras presionar en el botón generar.

Por otra parte, el administrador tendrá las mismas opciones que el usuario, solo que él, al tener acceso al panel de administrador, tendrá una opción adicional

-Administrar Usuarios: el administrador tendrá la opción de permitirle al usuario ingresar su nombre en caso de ser de confianza o denegarle el acceso en caso de así requerirlo.

Para implementar los requerimientos anteriores se usara el siguiente modelo.

SelAve
-Vista

RevGPS
-Vista

Filtrado
-Vista

RegNuev
-Vista

Estad
-Vista

Usua
-Vista

SolCoordenadas
-Vista

RevEnv
-Vista

Mapa
-Vista

RegEsp
-Vista

RegUsu
-Vista

Exc
-Vista

CatAve
-Controlador

GPS
-Controlador

BaseInterna
-Controlador

Envio
-Controlador

Ave
-Modelo

Coordenadas
-Modelo

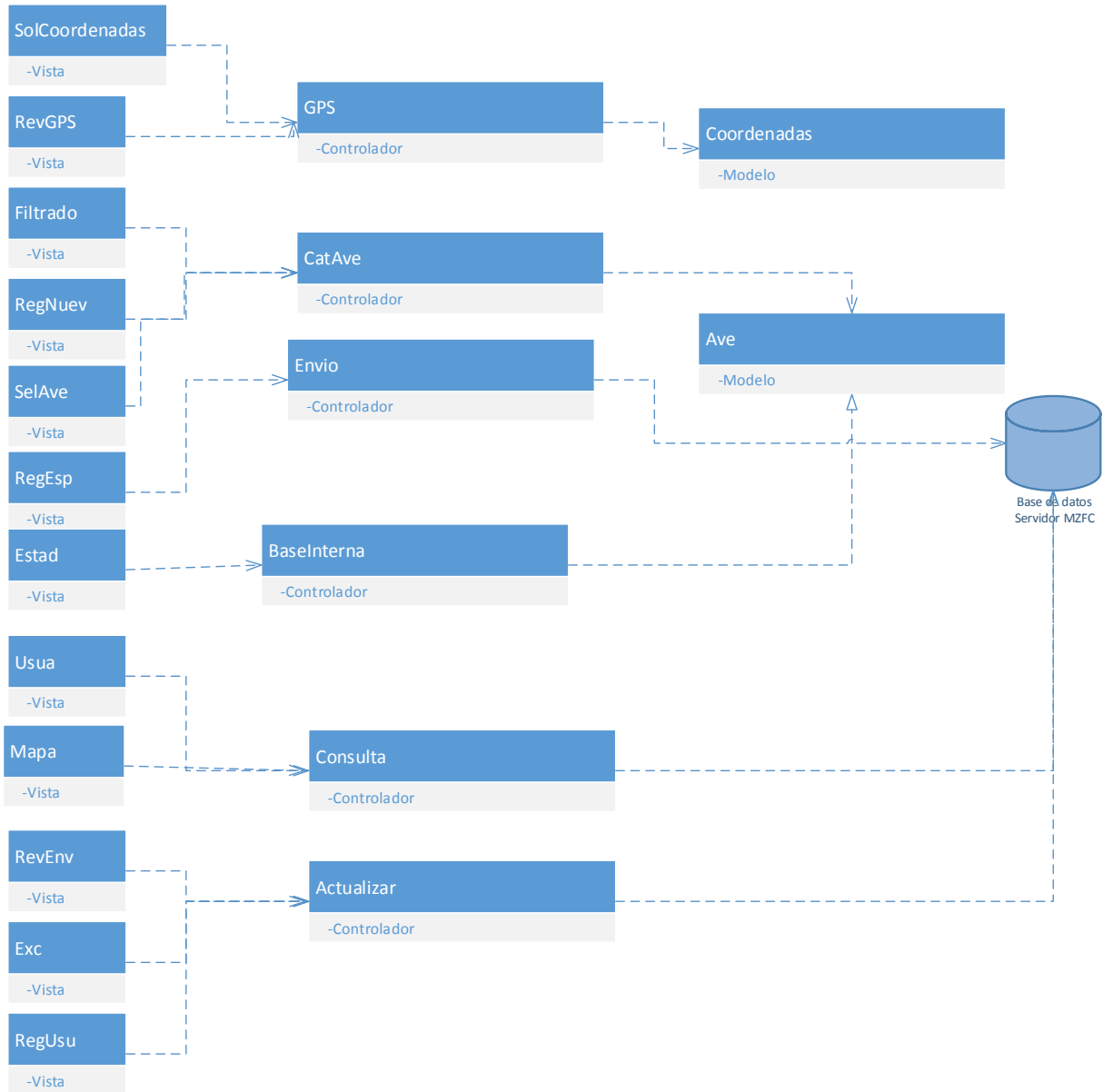


Diagrama de componentes.

3.2 Diseño de interfaz de Usuario

3.2.1 Interfaz de Usuario

La interfaz de usuario consiste en una serie de pantallas con las que el usuario podrá interactuar en su teléfono inteligente. Cada una de estas, está diseñada para realizar una tarea específica.

3.2.2 Descripción de la interfaz de usuario

Cada pantalla estará integrada por varios componentes: botones, barras, etiquetas dispuestas para que el usuario realice las acciones eficientemente.

3.2.3 Pantallas

El sistema estará compuesto de las siguientes:

3.2.3.1 Pantalla de carga

Es la primera pantalla que observara el usuario. Esta pantalla no presenta alguna funcionalidad importante, sin embargo se muestra para describir la App.

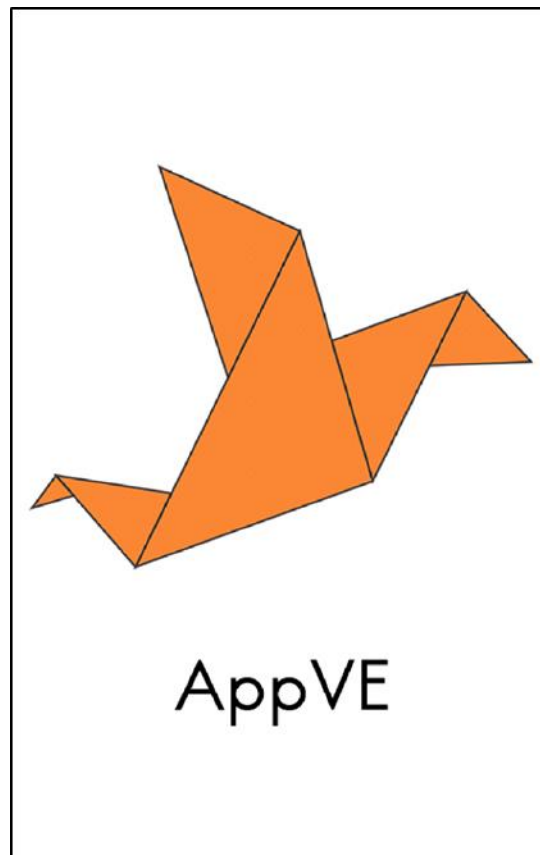


Imagen 15. Pantalla de carga. Fuente: propia.

3.2.3.2 Pantalla Principal

Es la principal pantalla de la aplicación la cual nos permitirá ir a las siguientes pantallas de la aplicación sin problema alguno.

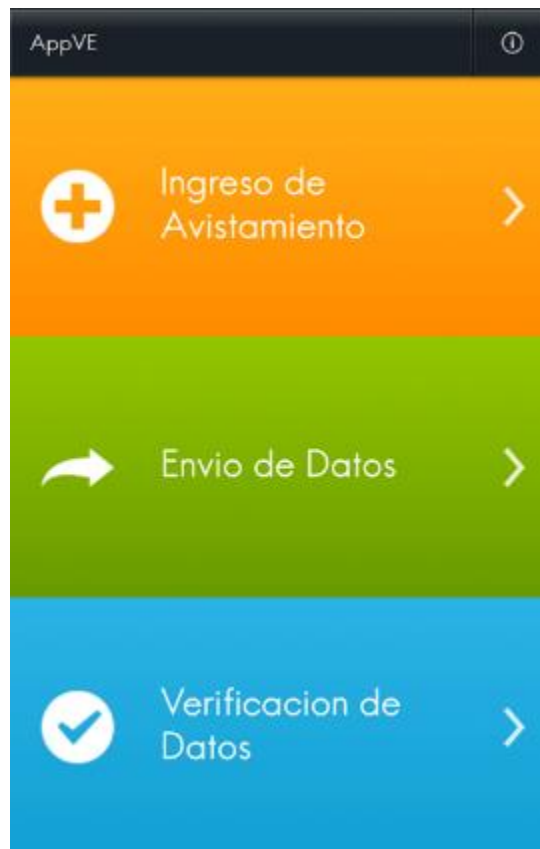


Imagen 16. Pantalla Principal. Fuente: propia.

3.2.3.3 Pantalla Ingreso de Avistamiento

Es la primera opción que se encuentra en la pantalla principal, después de dar clic en Ingreso de avistamiento. En esta pantalla nos mostrara la lista de aves, la ubicación geográfica y el campo de Observaciones.



Imagen 17. Pantalla de Ingreso de Avistamiento. Fuente: propia.

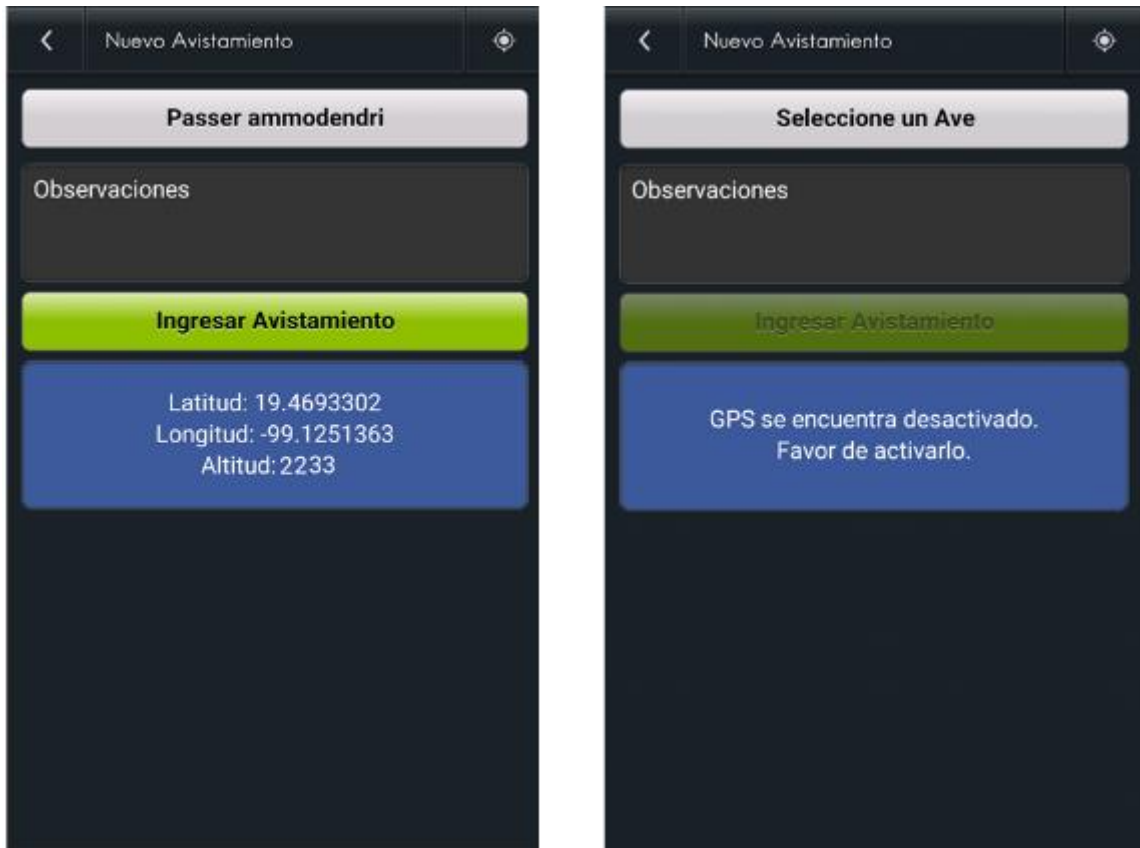


Imagen 18. Pantalla de Ingreso de Avistamiento. Fuente: propia.

3.2.3.4 Pantalla Envió de datos

Es la segunda opción que se encuentra en la pantalla principal, después de dar clic en Envió de datos. En esta pantalla nos mostrara los avistamientos que faltan por enviarse al MUZFC.

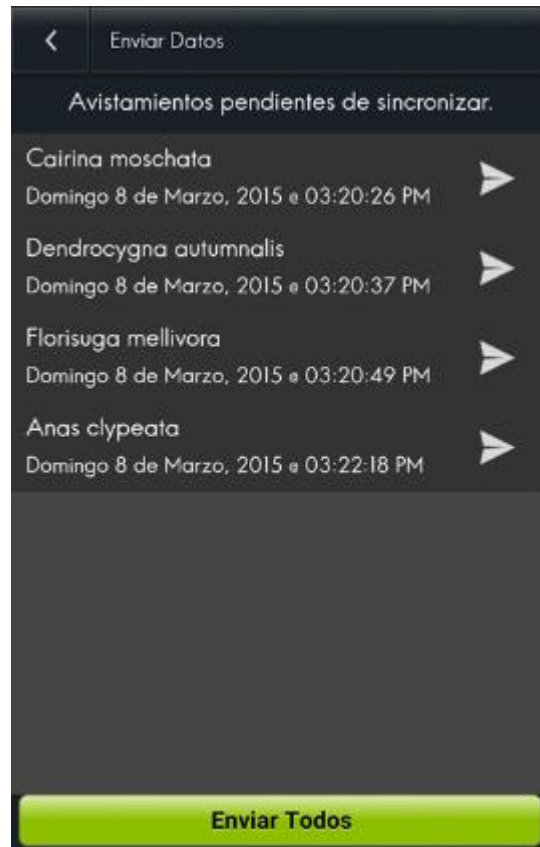


Imagen 19. Pantalla Envió de datos. Fuente: propia.

3.2.3.5 Pantalla Verificación de Datos

Es la tercera opción que se encuentra en la pantalla principal, después de dar clic en Verificación de datos. En esta pantalla nos mostrara los avistamientos que faltan por enviarse al MUZFC.

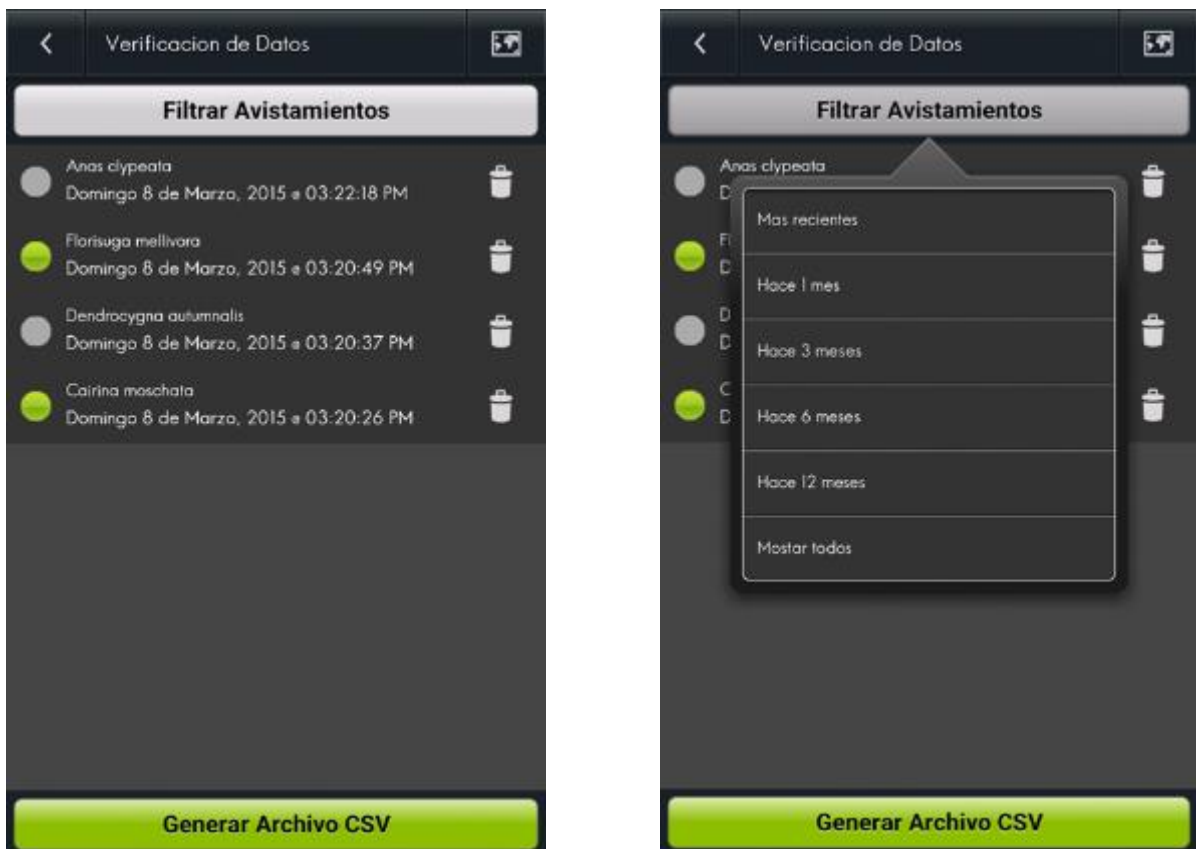


Imagen 20. Pantalla Verificación de Datos. Fuente: propia.

3.2.3.6 Pantalla Mapa de Datos

Ubicado en la esquina superior derecha dentro de la pantalla de Verificación de Datos, la cual nos permitirá ver en forma más geográfica los datos obtenidos en nuestros registros.

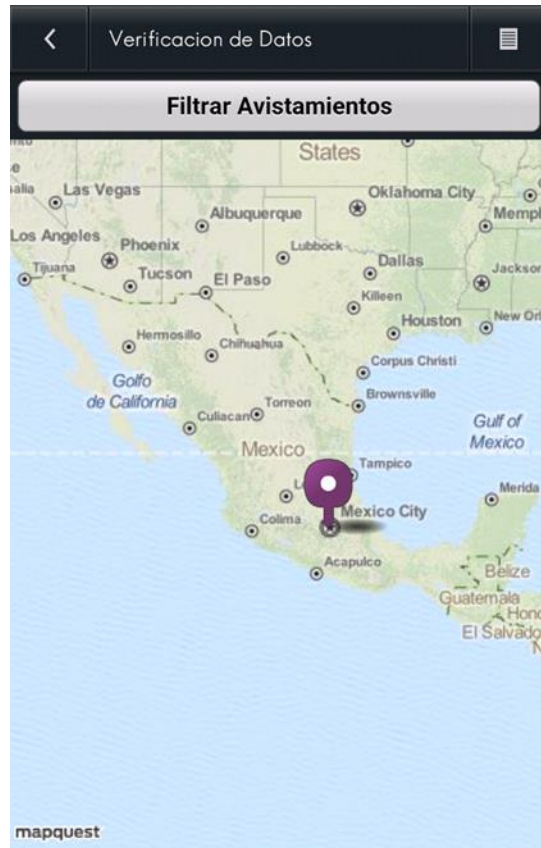


Imagen 21. Pantalla Mapa de Datos. Fuente: propia.

3.2.3.7 Pantalla Acerca de

Ubicado en la esquina superior derecha dentro de la pantalla Principal, nos muestra los créditos y el uso de esta aplicación.

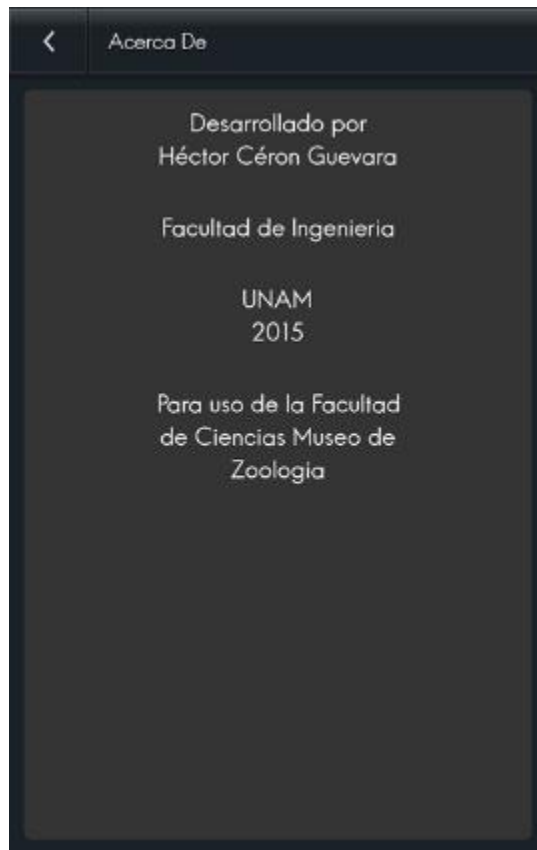


Imagen 22. Pantalla Acerca de. Fuente: propia.

3.2.3.8 Pantalla Usuario Verificado

Pantalla de acceso exclusivo luego de que el administrador haya aprobado al usuario como confiable, permitirá ingresar un nombre para su fácil identificación en el sistema del administrador.

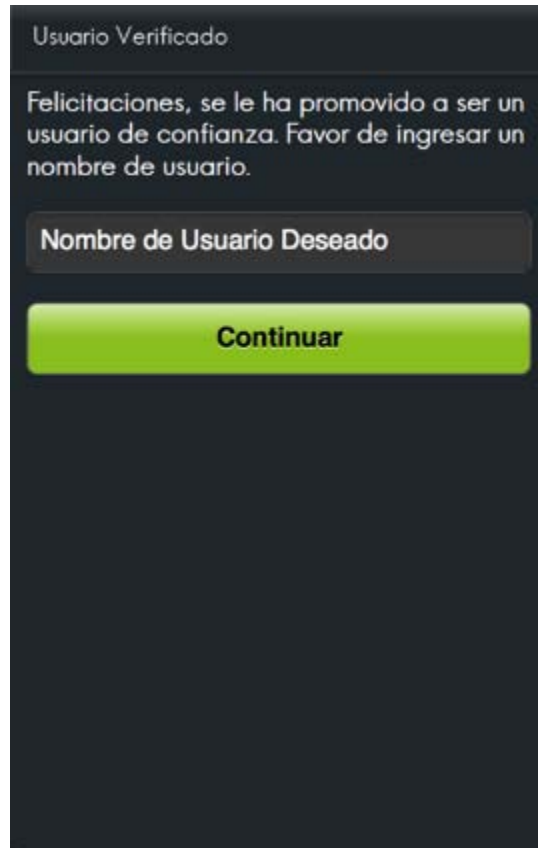


Imagen 23. Pantalla Usuario Verificado. Fuente: propia.

3.2.3.10 Pantalla por Especie

Pantalla en la que se muestran los registros por especie. Se muestra también que aquellas que no están pintadas de verde son aquellas que no han sido aún verificadas.

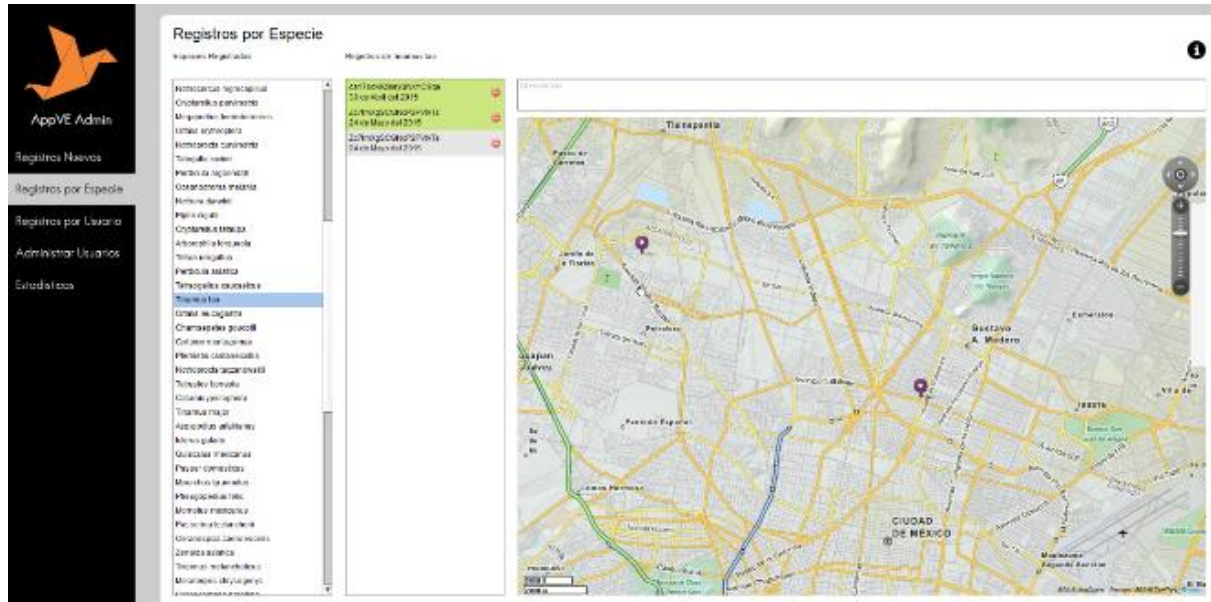


Imagen 25. Pantalla Registros por Especie. Fuente: propia.

3.2.3.11 Pantalla Registros por Usuario

Pantalla en la que se muestran los registros por Usuario. Se muestra también que aquellas que no están pintadas de verde son aquellas que no han sido aún verificadas.

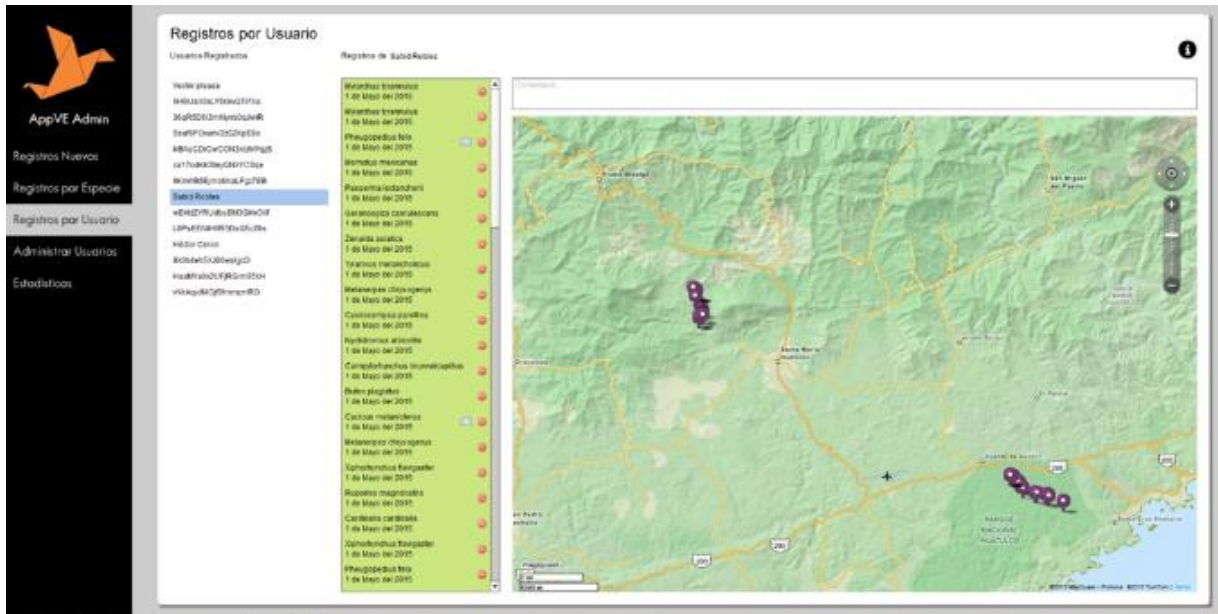


Imagen 26. Pantalla Registros por Usuario. Fuente: propia.

3.2.3.12 Pantalla Administrar Usuarios

Pantalla en la que se muestran los usuarios válidos y no válidos, así como las opciones para volverlos confiables o impedirles el acceso.

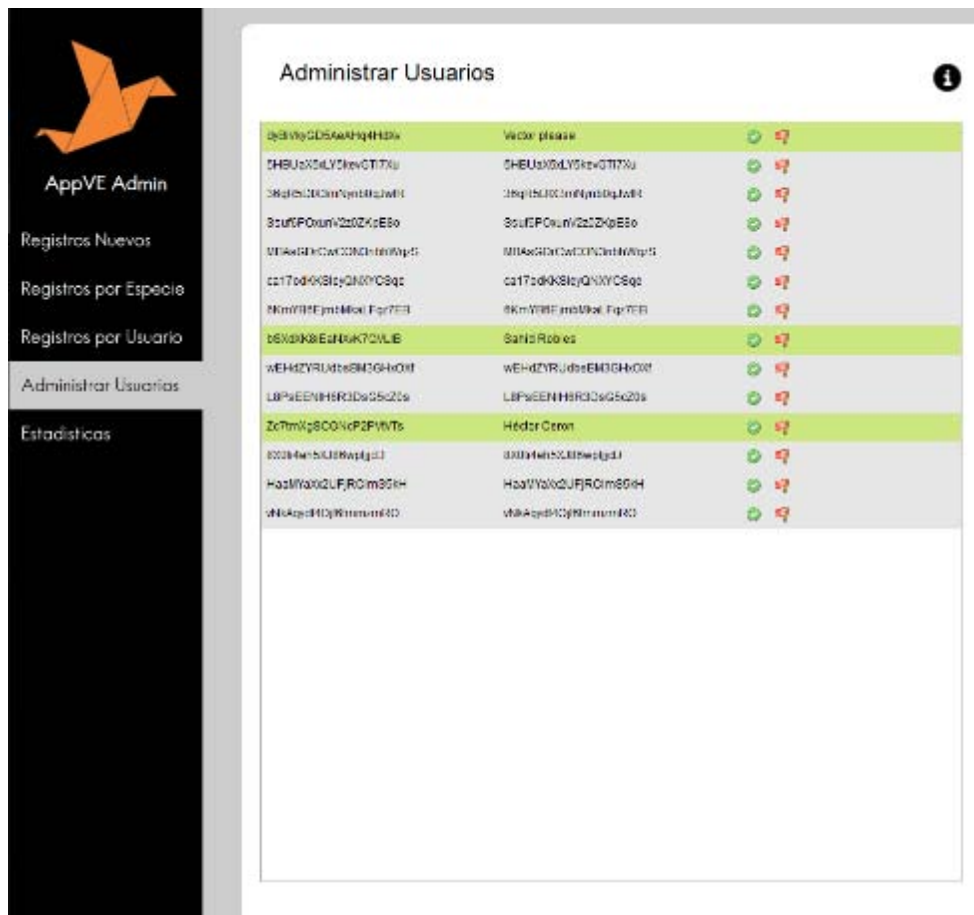


Imagen 27. Pantalla Registros por Usuario. Fuente: propia.

3.2.3.13 Pantalla Estadísticas

Pantalla en la que se muestran las estadísticas generales de las bases de datos y que permite crear archivos de Excel para un análisis a futuro.

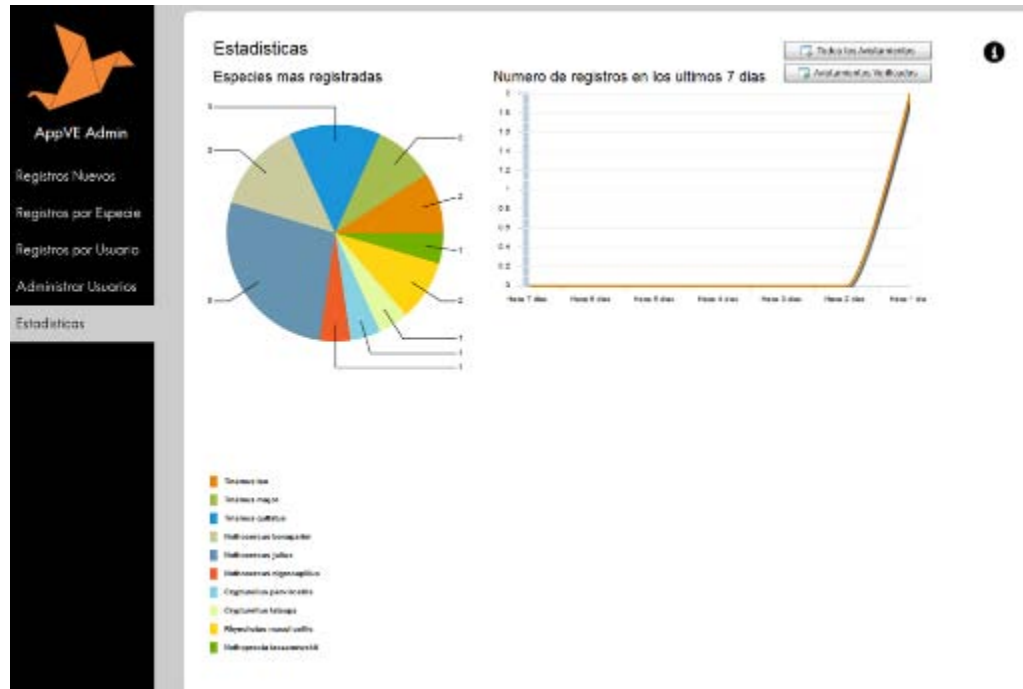


Imagen 28. Pantalla Estadísticas. Fuente: propia.

3.3 Interfaz de Hardware

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del Usuario:

- Teléfono inteligente con GPS

De parte del Administrador

- Equipo con conexión a internet

3.4 Interfaz de Software

Para un óptimo funcionamiento del sistema es necesario tener:

De parte del Usuario:

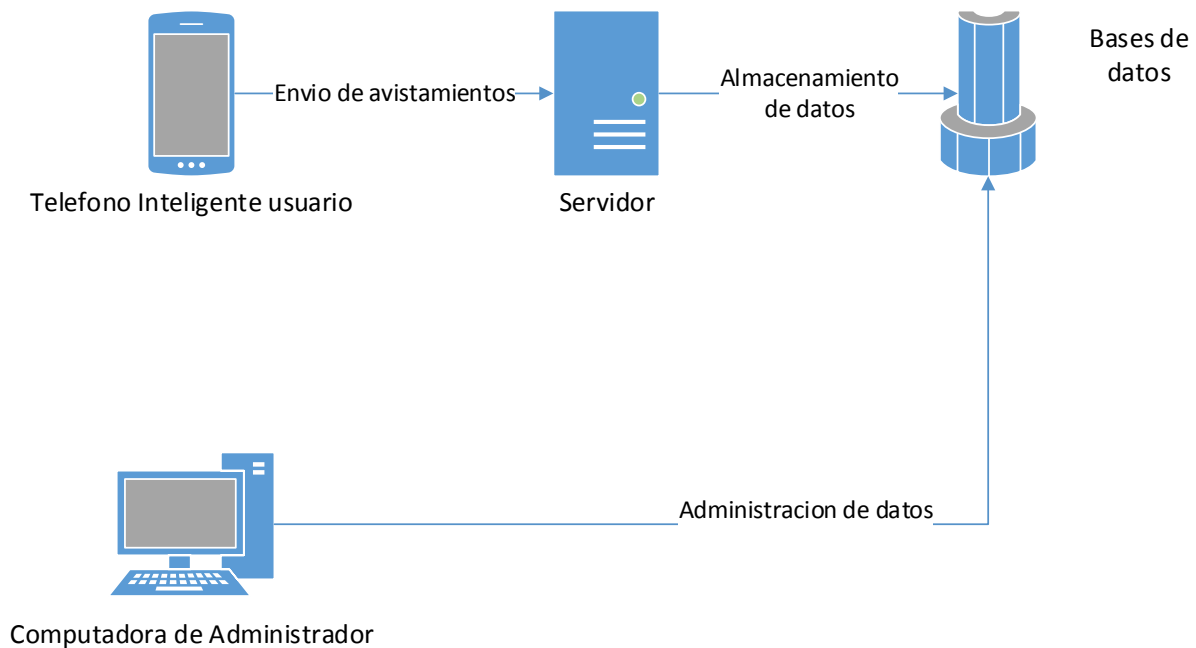
- Celular con sistema operativo Android
- 50 Mb de espacio disponible en memoria

De parte del Administrador

- Equipo con navegador web con Adobe Flash instalado

3.5 Interfaz de comunicación

Se utilizara el protocolo TCP/IP para conectar la aplicación móvil con el sistema en línea.



4. Características del sistema

El usuario podrá interactuar con el sistema mediante el uso de la aplicación móvil la cual permitirá enviar avistamientos y validar sus datos antes de ser enviados a un servidor donde el administrador validara toda la información que haya mandado el usuario.

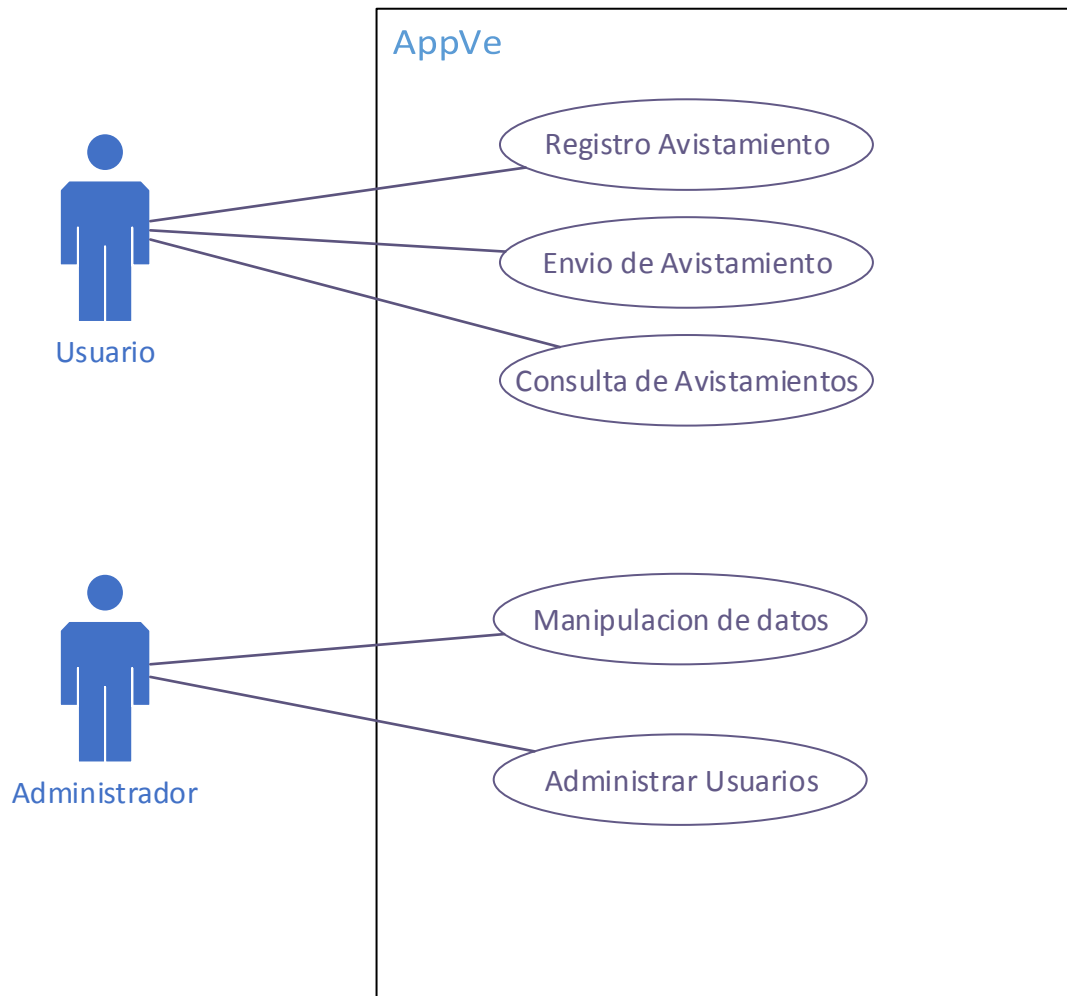


Diagrama de caso de Uso

4.1. Casos de uso

4.1.1 Ingreso de avistamiento

El usuario podrá ingresar un avistamiento a partir de una observación validada y será guardada siempre y cuando el teléfono inteligente tenga los datos de localización obtenidos por el GPS.

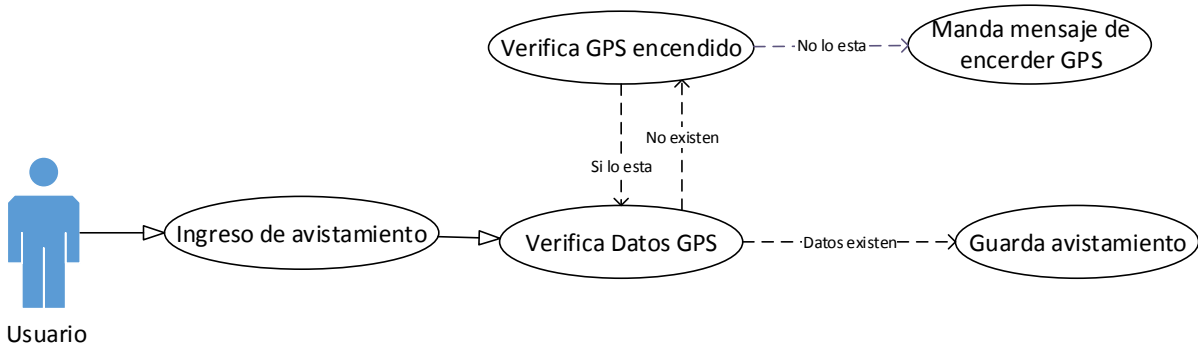


Diagrama de caso de uso de Ingreso de Avistamiento

Nombre del caso de uso: Ingreso de Avistamiento	Prioridad: Alta
Actor Principal: Usuario	
StakeHolders involucrados: Usuario	
Descripción: Poder ingresar un avistamiento que se haya realizado en campo.	
Objetivo: Ingresar un avistamiento realizado en campo con sus coordenadas geográficas.	
Medida de éxito: Se mostraran las coordenadas y al momento de guardas avistamiento, se regresara a la pantalla anterior tras realizar la observación.	
Precondición: Haber realizado una observación con éxito.	
Primer contacto: Hacer clic en el botón Ingreso de avistamiento.	
Secuencia: 1) Seleccionar en la pantalla de inicio el botón Ingreso de avistamiento. 2) Seleccionar en el catálogo el nombre del ave. 3) En caso de tener observaciones anotarlas en el cuadro correspondiente. 4) Esperar a las coordenadas. 5) Seleccionar el botón de Guardar Avistamiento.	
Suposiciones: El usuario debe de haber la identificación correctamente al ave.	
Entrada: Nombre del ave forzoso y Observación opcional.	
Salida: Se guardara en el teléfono inteligente la observación.	

4.1.2 Envió de datos

El usuario podrá enviar datos de avistamientos realizados por el al servidor del MZFC.

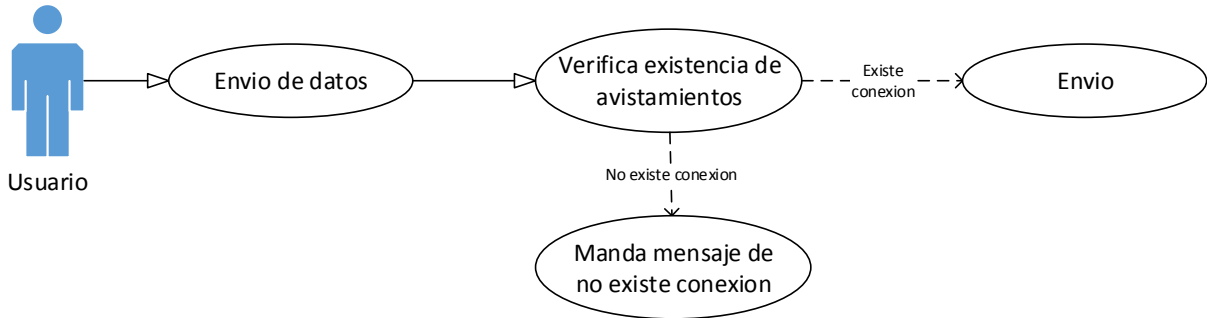


Diagrama de caso de uso Envió de datos

Nombre del caso de uso: Envió de datos	Prioridad: Alta
Actor Principal: Usuario	
StakeHolders involucrados: Usuario	
Descripción: Poder enviar los datos de observación realizadas por el usuario al MZFC para validación.	
Objetivo: Enviar datos al MZFC para su validación.	
Medida de éxito: Mensaje de datos enviados.	
Precondición: Tener al menos un dato de avistamiento realizado	
Primer contacto: Hacer clic en el botón Envió de datos	
Secuencia: 1) Seleccionar en la pantalla de inicio de Envió de datos. 2) Seleccionar uno por uno cada avistamiento o seleccionar el botón enviar todos. 3) Esperar a que se confirme el envío	
Suposiciones: El usuario tendrá al menos una observación pendiente para enviar	
Entrada: Avistamientos sin enviar.	
Salida: Se confirmara que han sido enviados con éxito, en caso contrario no se mandaran.	

4.1.3 Muestra avistamientos

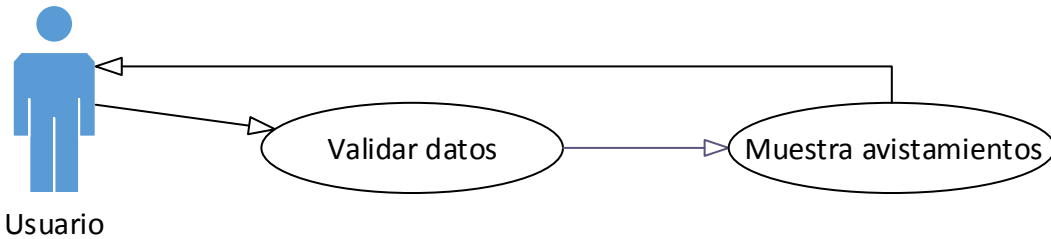


Diagrama de caso de uso Muestra avistamientos

Nombre del caso de uso: Muestra avistamientos	Prioridad: Alta
Actor Principal: Usuario	
Stakeholders involucrados: Usuario	
Descripción: Poder visualizar todos los avistamientos realizados por el usuario para su valuación personal.	
Objetivo: Ver los avistamientos realizados por el usuario.	
Medida de éxito: Los avistamientos serán mostrados en la pantalla.	
Precondición: Tener al menos realizado un avistamiento previamente	
Primer contacto: Dar clic en el botón Verificación de Datos.	
Secuencia: 1) Seleccionar en la pantalla de inicio Verificación de Datos. 2) Observar los datos	
Suposiciones: El usuario tendrá al menos un avistamiento realizado.	
Entrada: Avistamientos realizados.	
Salida: Se muestra los datos de los avistamientos.	

4.1.4 Validar información

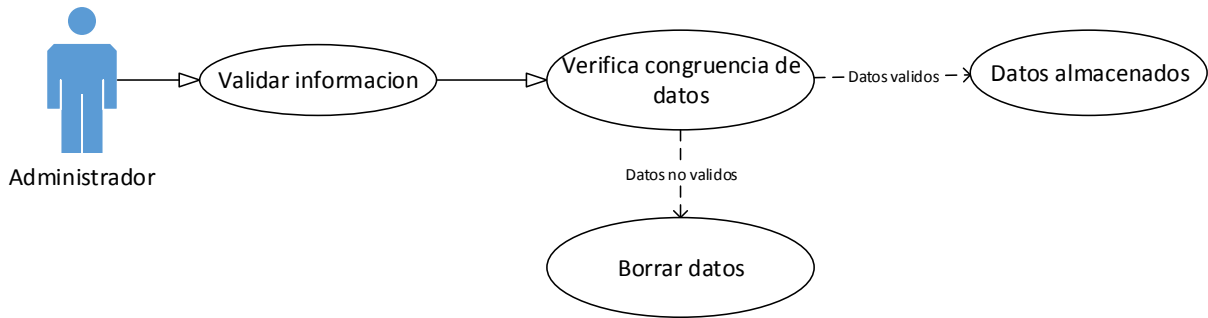


Diagrama de caso de uso Validar información

Nombre del caso de uso: Validar información	Prioridad: Alta
Actor Principal: Administrador	
Stakeholders involucrados: Administrador	
Descripción: El administrador podrá validar si la información enviada por los usuarios es válida o no, acorde con los conocimientos del administrador de especies nativas del lugar de registro.	
Objetivo: El administrador validara si la información es correcta o no.	
Medida de éxito: El administrador ingresara el valor de valido o no al sistema.	
Precondición: Tener al menos un dato pendiente por validar.	
Primer contacto: Al momento de ingresar al panel de administrador se mostraran todos los avistamientos a validar.	
Secuencia: 1) Ingresar en el panel de administrador. 2) Dar clic en aceptar o rechazar avistamiento.	
Suposiciones: El administrador debe al menos de tener uno o más avistamientos pendientes por revisar.	
Entrada: Avistamientos pendientes por enviar.	
Salida: El avistamiento desaparecerá de la lista de pendientes por revisar.	

4.1.5 Validar usuarios

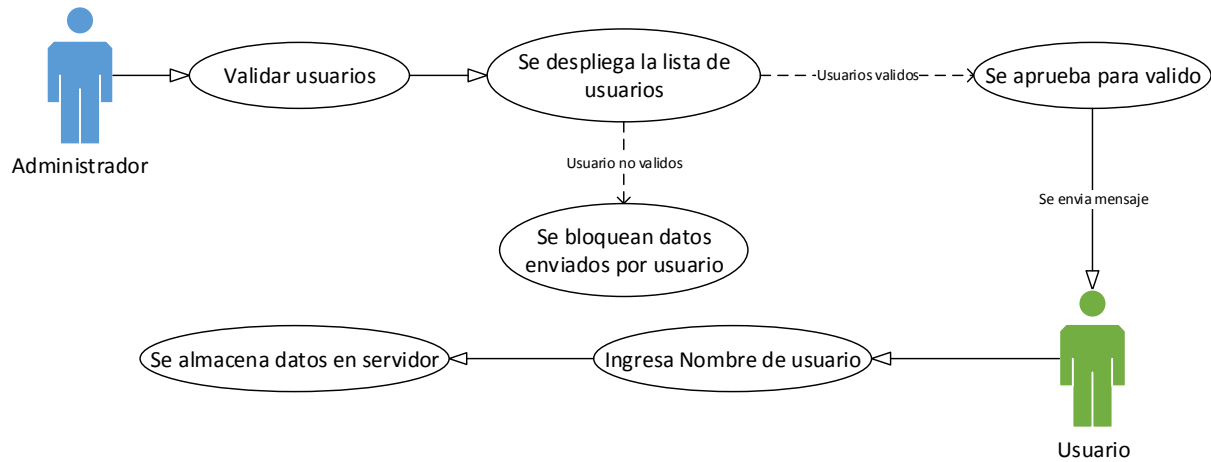


Diagrama de caso de uso Validar usuarios

Nombre del caso de uso: Validar usuarios	Prioridad: Alta
Actor Principal: Administrador	
StakeHolders involucrados: Administrador y usuario	
Descripción: El administrador permitirá al usuario ser de confianza para poder ingresar su nombre personalizado y tener un mejor control de quien envía o bloquearlo en caso de que este usuario mande varios registros incorrectos.	
Objetivo: El administrador aprobará a un usuario para ser válido o bloqueará aquel que mande datos incorrectos.	
Medida de éxito: El usuario puede ingresar su nombre dentro de la App	
Precondición: No estar registrado actualmente como un usuario de confianza.	
Primer contacto: El administrador verá en el panel todos los usuarios.	
Secuencia: 1) El administrador verá todos los usuarios con sus nombres generados al azar dentro del sistema 2) El administrador decidirá que usuario es de confianza y se aprobará para el usuario pueda ingresar su nombre en sistema. En caso de ser un usuario para bloquear, se dará clic en bloquear. 3) El usuario en caso de ser de confianza, se le mandará una pantalla al momento de enviar datos la cual le pedirá que ingrese su nombre para mandarlo al sistema.	
Suposiciones: Tener al menos un usuario que haya mandado un avistamiento.	
Entrada: El administrador autoriza un usuario.	
Salida: El administrador recibe el nombre del usuario.	

5. Otros requerimientos no funcionales

5.1. Requerimientos de Desempeño (Performance Requirement)

El sistema se lograra hacer lo suficientemente ligero como para que cualquier celular con sistema operativo Android pueda correrlo.

5.2. Requerimientos de seguridad (Física)

El servidor del sistema debe contar con instalaciones adecuadas: sin polvo, sin cambios bruscos de voltaje y con nula o poca humedad.

5.3. Requerimientos de seguridad

El sistema no maneja datos de información sensible para el usuario, sin embargo todos los avistamientos al tener que ser verídicos, se pide una seguridad al menos media o alta para resguardar todos estos datos así como generación de respaldos periódicos.

5.4. Atributos de Calidad

El sistema cumple con los siguientes atributos de calidad.

- **Confiabilidad**

El sistema estará siempre funcionando si se sigue correctamente los requerimientos básicos de hardware.

- **Robustez**

Al estar compuesto por componentes de uso amplio (Flex y MySQL) se garantiza este punto.

- **Disponibilidad**

Si el servidor se mantiene en línea todo el tiempo, este sistema se podrá consultar en cualquier momento. Si por alguna situación el servidor se apaga, el sistema funcionara independiente en la aplicación móvil para el usuario.

- **Mantenibilidad**

La actualización del sistema y el cuidado de las bases de datos así como de la aplicación correrán a cargo del administrador del MZFC una vez entregado el sistema.

- **Usabilidad**

El sistema será lo suficientemente intuitivo y simple para que cualquier tipo de usuario pueda usarlo.

- Portabilidad

Gracias a que el sistema administrativo no depende del sistema operativo, cumplimos con este atributo.

5.5 Esquema de Negocios (Business Rules)

El sistema tendrá como meta desarrollarse en un total de 2 meses basados en el siguiente calendario.

No.	Nombre	Duración	Principio	Fin	Predecesor
1	Generación de diseño de bases de datos	3 días	30-abr-15	02-may-15	
2	Creación de base con base al diseño	2 días	03-may-15	05-may-15	1
3	Documentación de base de datos	2 días	06-may-15	08-may-15	1;2
4	Integración de base de datos	1 día	09-may-15	10-may-15	3
5	Diseño de clases y vistas	5 días	11-may-15	16-may-15	4
6	Creación de vista base para App	5 días	17-may-15	22-may-15	5
7	Implementación de Ingresar Avistamiento	3 días	23-may-15	26-may-15	4
8	Implementación de Envió de datos	6 días	27-may-15	01-jun-15	4
9	Implementación de verificación de datos 1	3 días	02-jun-15	05-jun-16	4
10	Implementación de verificación de datos 2	5 días	07-jun-15	15-jun-15	4
11	Creación de vista base para sistema Web	6 días	16-jun-15	22-jun-15	7;8;9;10
12	Implementación ingresos nuevos	4 días	23-jun-15	27-jun-15	11
13	Implementación panel de administrador	8 días	28-jun-15	05-jul-15	11

Este mismo será con el fin de llevar el desarrollo a un fin completo sin retrasos ni cambios.

Una vez finalizado este desarrollo se contará con un periodo de pruebas de 15 días determinado por el cliente (MZFC) el cual podrá hacer observaciones del mismo y se podrán realizar cambios menores que no afecten el sistema funcional del sistema y que no sean de caso crítico. Estas pruebas tendrán como almacenamiento temporal los servidores de pruebas del desarrollador.

Una vez finalizado este periodo de pruebas se procederá a la instalación completa en las instalaciones que el cliente así señale y se entregará los manuales de usuario, administrador y de instalación.

Aprobación de documento

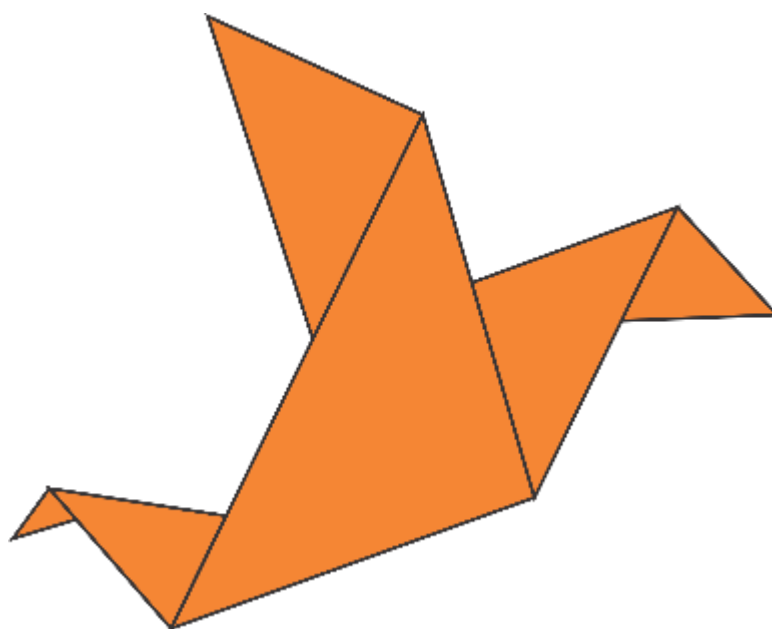
El siguiente documento y los requerimientos del sistema AppVe han sido aprobados por:

Alejandro Gordillo Martínez
Maestro en Ciencias
Administrador de datos de MZFC

Cerón Guevara Héctor
Desarrollador

ANEXO 4 ACUERDO DE TRABAJO

PARA EL SISTEMA



AppVe

1. Introducción

Las características que debe cumplir y los tiempos en que debe entregarse el sistema AppVe tomando como base una tabla SOW se describen a continuación, además de plasmar aquí mismo los factores que intervienen en el cálculo del costo que tendrá dicho sistema. Los acuerdos especificados en la forma y tiempo de pago se especifican a continuación en el documento, añadiendo además las horas, equipo de trabajo involucrado y costo de instalaciones donde se desarrolla el sistema AppVe y demás factores que intervienen en el costo que tendrá dicho sistema.

1.1 Propósito

El propósito de este documento es mencionar los acuerdos entre el cliente (MZFC) y el desarrollador en cuanto a fechas de entrega del sistema AppVe, así como dar el costo que tendrá el sistema poniendo los factores que intervienen en dicho cálculo. En este caso, se proporcionara una tabla con las tareas a ejercer con el tiempo de cada una de ellas, la fecha inicio y la fecha fin.

1.2 Audiencia dirigida y sugerencias de lectura

Los beneficios del sistema podrán ser percibidos en su totalidad por el personal del Museo de Zoología Alfonso L. Herrera de la Facultad de Ciencias de la Universidad Nacional Autónoma de México que a partir de aquí se referirá como el cliente

2. Alcance general

El siguiente documento describe el proceso necesario para desarrollar el sistema AppVe, así como un marco común entre el cliente y el desarrollador para medir el avance del proyecto. AppVe será un sistema que permita registrar los avistamientos realizados por el cliente y tendrá la posibilidad de administrarlos para un análisis futuro.

El proyecto será desarrollado de acuerdo a las siguientes etapas del proyecto:

- Toma de requerimientos
- Elaboración de SRS
- Elaboración de SDD

- Implementación

La implementación del sistema será hecha en Flash Builder usando ActionScript 3.0, Apache Flex y SQLite de para generar la App. Para el desarrollo del sistema en la sección del administrador se usaron las mismas herramientas anteriores pero en lugar de SQLite se usó MySQL y para la implementación se usó HTML.

Para que el sistema AppVe corra a la perfección, es necesario cumpla con los siguientes requerimientos:

Para la App, AppVe necesitara las siguientes características:

- Android 2.3 o superior instalado
- Contar con GPS
- Tener permisos de instalación de fuentes desconocidas de aplicaciones sobre el mismo.

Para el panel de administrador, AppVe se ejecutara en un servidor con las siguientes características mínimas:

- Memoria RAM de 4 gigabytes
- Tarjeta gráfica
- Puerto 8080 abierto (para Apache HTTP Server)
- Conexión a internet
- Apache HTTP Server: Versión 2.4.
- MySQL
- PHPMyAdmin

3. Actividades y responsabilidades

Tareas	Software
Documentación base de datos	SQLite, MySQL
Drenado de la base de datos	SQLite, MySQL
Controlador de la base de datos	SQLite, MySQL

Implementación de consulta en base de datos	SQLite, MySQL
Documentación Diseño de Clases	Apache Flex
Diseño de Vistas	Apache Flex
Documentación de vistas	Apache Flex
Ingresar Avistamiento	Apache Flex
Envío de datos	Apache Flex
Verificación de Datos: Texto Plano Mapa	Apache Flex
Panel de administrador Ingresos nuevos	Apache Flex
Panel de administrador	Apache Flex

4. Periodo de actividades

No.	Nombre	Duración	Principio	Fin	Predecesor
1	Generación de diseño de bases de datos	3 días	30-abr-15	02-may-15	
2	Creación de base con base al diseño	2 días	03-may-15	05-may-15	1
3	Documentación de base de datos	2 días	06-may-15	08-may-15	1;2
4	Integración de base de datos	1 día	09-may-15	10-may-15	3
5	Diseño de clases y vistas	5 días	11-may-15	16-may-15	4
6	Creación de vista base para App	5 días	17-may-15	22-may-15	5
7	Implementación de Ingresar Avistamiento	3 días	23-may-15	26-may-15	4
8	Implementación de Envío de datos	6 días	27-may-15	01-jun-15	4
9	Implementación de verificación de datos 1	3 días	02-jun-15	05-jun-16	4
10	Implementación de verificación de datos 2	5 días	07-jun-15	15-jun-15	4
11	Creación de vista base para sistema Web	6 días	16-jun-15	22-jun-15	7;8;9;10
12	Implementación ingresos nuevos	4 días	23-jun-15	27-jun-15	11
13	Implementación panel de administrador	8 días	28-jun-15	05-jul-15	11
14	Pruebas	15 días	06-jul-15	21-jul-15	13;12
15	Documentación final	3 días	22-jul-15	25-jul-15	14
16	Generación de manuales	3 días	26-jul-15	29-jul-15	15
17	Entrega de proyecto y documentación	5 días	30-jun-15	04-ago-15	16

5. Calendario de eventos

Entregable	Fecha
Preview proyecto funcional y toma de cambios	27-may-15
Preview final con cambios solicitados	04-jul-15
Entrega de sistema para pruebas	06-jul-15
Instalacion completa	05-ago-15

6. Estándares a ocupar

La elaboración del SRS se basó en el estándar Std 830-1998 del Institute of Electrical and Electronics Engineers.

La elaboración del SDD fue basada en el estándar Std 1016TM-2009 del Institute of Electrical and Electronics Engineers.

La elaboración de la documentación se basara en el estándar IEEE Std 1063-2001.

7. Requerimientos especiales

Los costos del desarrollo en Moneda Nacional (M.N.) de este sistema serán los siguientes.

Gastos Anuales	Precio unitario	Cantidad	Total
Equipo de computo	\$ 15,000.00	1	\$ 15,000.00
Teléfono Inteligente Android 2.2	\$ 500.00	1	\$ 500.00
Teléfono Inteligente Android 5.1	\$ 12,000.00	1	\$ 12,000.00
Consumibles alimenticios	\$ 300.00	12	\$ 3,600.00
Gastos de mantenimiento	\$ 5,000.00	12	\$ 60,000.00
Renta servidor para pruebas	\$ 900.00	12	\$ 10,800.00
Sueldo programador	\$ 18,000.00	12	\$ 216,000.00
			\$ 317,900.00

El costo anual por el desarrollo de una aplicación con estos requerimientos será el proyectado en la tabla anterior, sin embargo este desarrollo quedo acotado a un periodo de 2 meses.

Haciendo constar esto el costo por desarrollo al mes será de \$26,491.67 (Veinte y Seis Mil Cuatrocientos Noventa Uno Con 67/100 M.N.)

El costo total del proyecto al final del desarrollo será de \$ \$52,983.33 (Cuarenta y Seis Mil Quinientos Ochenta y Tres Con 33/100 M.N.)

Estos costos no incluyen impuestos.

8. Criterio y forma de pago

Este sistema es entregado a la UNAM de mi parte, por lo que el costo será nulo, es decir gratis ya que la Universidad me apoyo en todo lo posible para mi educación y me brindó la oportunidad de realizar un proyecto para resolver un problema de la vida real.

Aprobación de documento

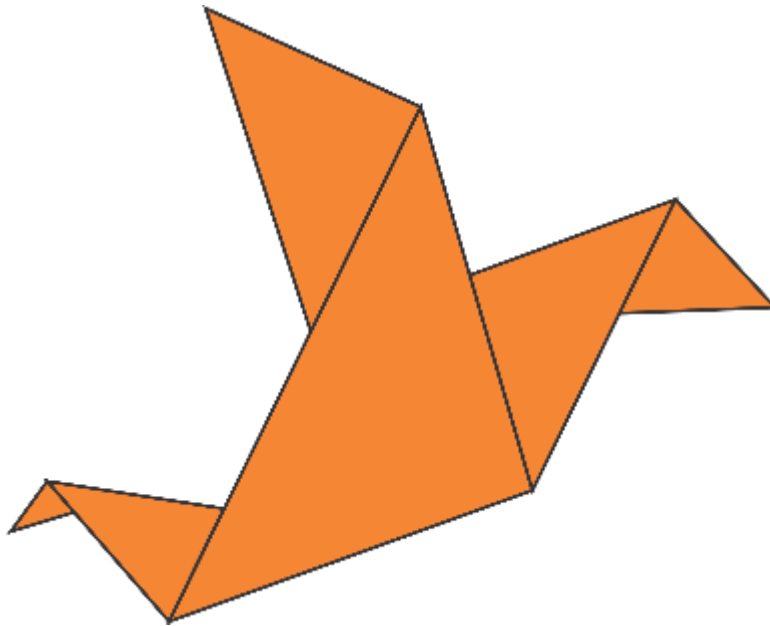
El siguiente documento y los requerimientos del sistema AppVe han sido aprobados por:

Alejandro Gordillo Martínez
Maestro en Ciencias
Administrador de datos de MZFC

Cerón Guevara Héctor
Desarrollador

ANEXO 5 MANUAL DE USUARIO

PARA EL SISTEMA



AppVe

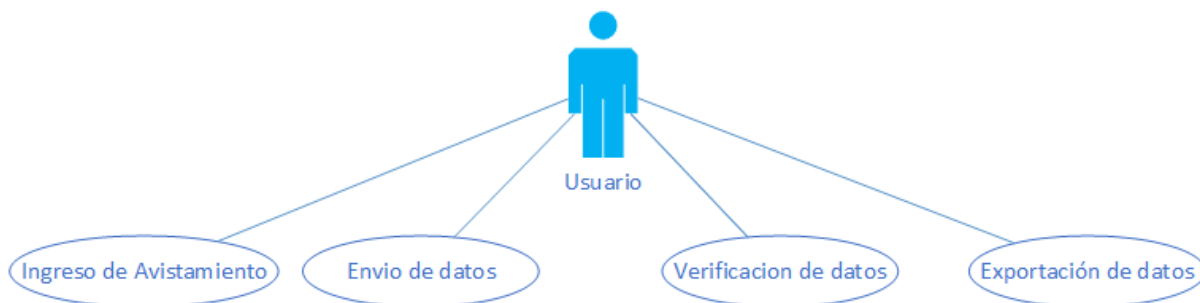
1. Introducción

Este documento contiene toda la información necesaria para que los usuarios puedan hacer uso del sistema AppVe. La finalidad del documento es ser una guía para conocer el funcionamiento de todas las características del sistema.

La observación de aves en México ha aumentado considerablemente desde que se inició sus actividades en la Nueva España. Así como el tiempo ha cambiado, la forma de almacenar datos y analizarlos ha cambiado también, es porque cubriendo una necesidad de tener los datos exactos o muy aproximados, el presente sistema pretende cubrir esa problemática.

AppVe es una aplicación móvil para teléfonos inteligentes con sistema operativo Android instalado el cual permitirá hacer registros para uso personal o para un análisis más profundo. Así mismo AppVe pretende de ser de gran ayuda para la Universidad Nacional Autónoma de México, ya que se tendrá dentro de sus funciones mandar datos a la misma para su validación y poder así ayudar a la investigación científica.

El siguiente diagrama describe las funciones que podrá realizar el usuario al momento de usar AppVe.



2. Conceptos importantes

2.1 Acceso a la aplicación

La aplicación está compuesta de varias secciones, pero existe una que es fundamental para el uso correcto de esta, la pantalla principal.



Esta será la primera pantalla que se mostrara al usuario al momento de ingresar a la aplicación. A partir de esta se darán todas las instrucciones que para cada uno de las funciones se irán mostrando.

3. Guía de uso

3.1 Ingreso de Avistamiento

En la pantalla principal seleccionar la opción Ingreso de Avistamiento.



Una vez seleccionada se desplegara la siguiente pantalla.



Para realizar un nuevo avistamiento se seleccionara la opción Seleccione un Ave para desplegar un catálogo como el siguiente.



Se seleccionara un ave del catálogo y se seleccionara en Ingresar avistamiento una vez que se haya localizado las coordenadas.



En caso de estar deshabilitado el GPS del teléfono inteligente se desplegara un anuncio de que está apagado mostrara un mensaje de que se debe encender en las opciones de GPS seleccionando el siguiente botón.

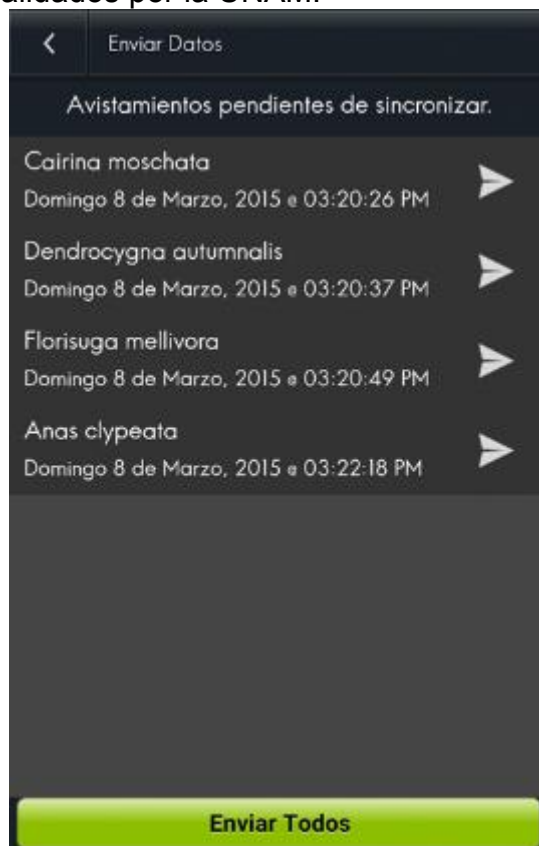


3.2 Envió de datos

En la pantalla principal seleccionar la opción Envió de Datos.



Una vez seleccionada se desplegara la siguiente pantalla con los avistamientos aun sin enviar a que sean validados por la UNAM.



Para realizar envíos de datos es fundamental que se tengan datos pendientes por enviar, ya que si no existe ningún dato pendiente, la pantalla estará vacía.

Para enviar un dato uno a uno se seleccionara la opción de flecha hacia la derecha.



En caso de que se requieran enviar todos se seleccionara la opción de enviar todos.

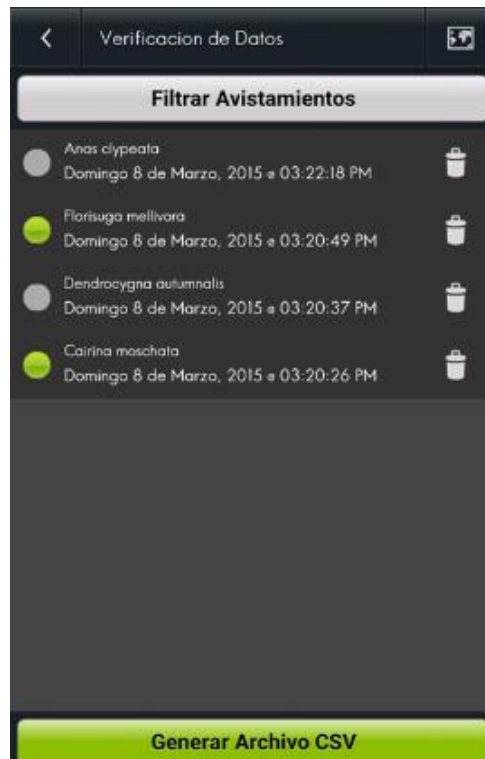


3.3 Verificación de datos

En la pantalla principal seleccionar la opción Verificación de datos.



Una vez seleccionada se desplegara la siguiente pantalla con los avistamientos realizados.



Para poder visualizar correctamente esta pantalla es fundamental tener al menos un avistamiento.

Dentro de verificación de datos existen dos modos, el modo texto Plano y el modo Mapa.

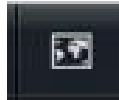
3.3.1 Verificación de datos Texto Plano

Corresponde a la opción por defecto que se muestra al seleccionar de la pantalla principal verificación de datos.

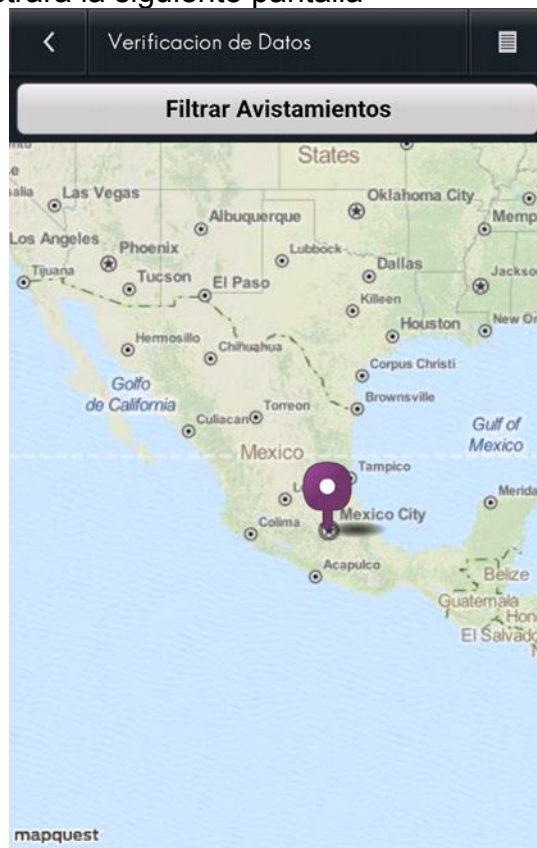
Se podrán visualizar aquí que datos ya han sido enviados por el color del botón al lado izquierdo del dato. Si se encuentra verde es que ya ha sido enviado, si se presenta de color gris, aún no ha sido enviado.

3.3.2 Verificación de datos Mapa

Se siguen los mismos pasos que la verificación anterior con la diferencia de que después de la pantalla de verificación se seleccionara la siguiente opción.



Al seleccionarla se mostrara la siguiente pantalla

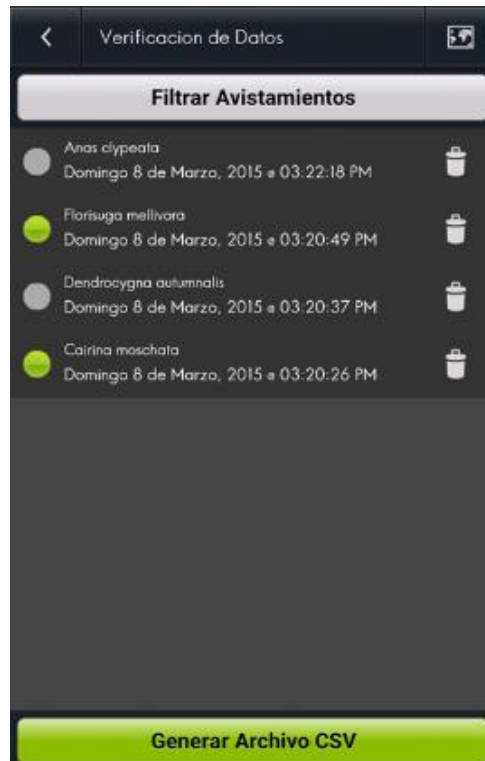


3.4 Exportación de datos

En la pantalla principal seleccionar la opción Verificación de datos.



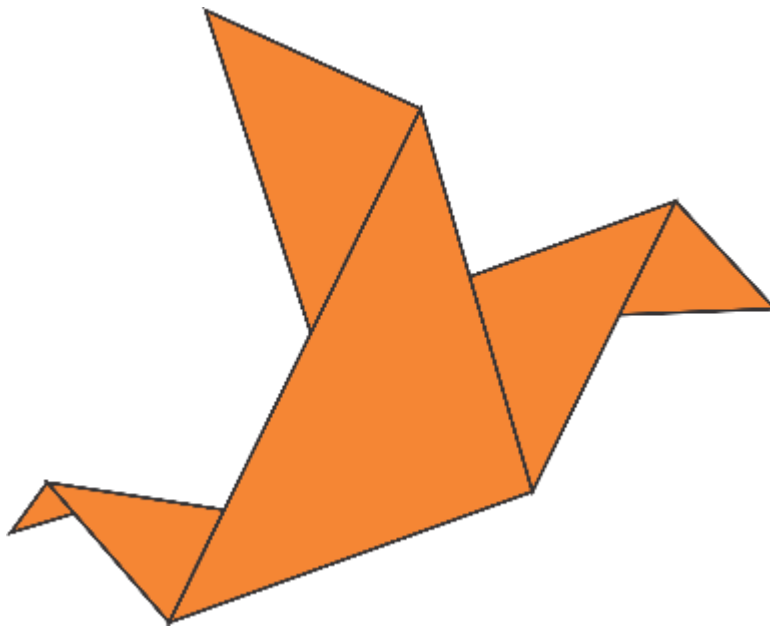
Una vez seleccionada se desplegará la siguiente pantalla con los avistamientos realizados.



Para poder exportar datos se seleccionará la opción Generar Archivo CSV y se creará un archivo en la memoria del teléfono inteligente con todos los avistamientos realizados.

ANEXO 6 MANUAL DE ADMINISTRADOR

PARA EL SISTEMA



AppVe

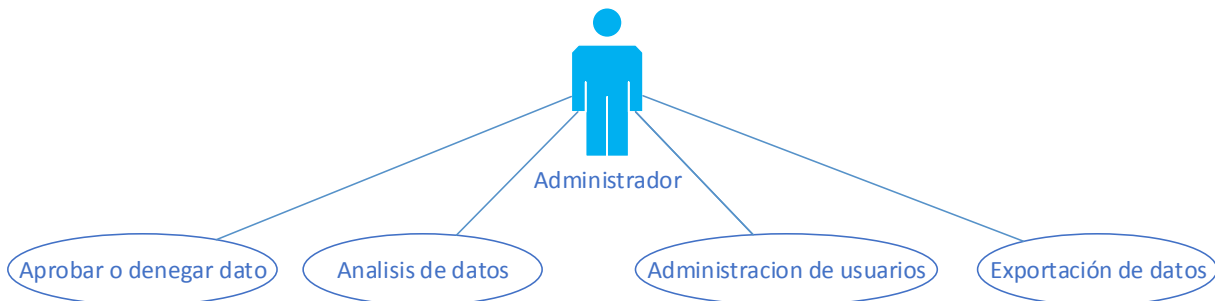
1. Introducción

Este documento contiene toda la información necesaria para que el usuario Administrador pueda hacer uso del sistema AppVe. La finalidad del documento es ser una guía para la administración tanto de los usuarios como de los datos generados por los usuarios.

AppVe es un sistema compuesto por 2 partes. Una versión móvil para teléfonos inteligentes y un panel administrador en un panel de administrador.

En el manual de usuario se explica a detalle el funcionamiento de la parte de la versión móvil que en su totalidad está dirigida a los usuarios.

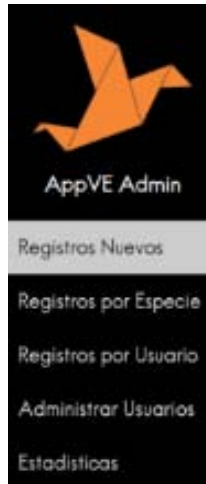
Para entender cada una de las funciones asignadas al administrador se desplegaran en el siguiente diagrama.



2. Conceptos importantes

2.1 Panel lateral

La pantalla del administrador estará siempre regida por un panel de lado izquierdo como el siguiente







Este panel nos permitirá ingresar a cada uno de las distintas pantallas del panel de administrador.

3. Guías de uso

3.1 Aprobar o denegar dato





Al ingresar al panel de administrador, se mostrara una pantalla como la siguiente.

 A screenshot of the 'Registros Nuevos' (New Records) interface. On the left is a sidebar with the 'AppVE Admin' logo and navigation options: Registros Nuevos, Registros por Especie, Registros por Usuario, Administrar Usuarios, and Estadísticas. The main area shows a table with the following data:

Especie	Usuario	Fecha	Acción
Halluxa cinerea	Zc7hmjg5CGAaP3PVN7x	12 de Mayo del 2015	 
Taraxacum	Zc7hmjg5CGAaP3PVN7x	24 de Mayo del 2015	 

 To the right of the table is a map of Mexico with a search bar at the top containing 'En caso'. The map shows major cities like Ciudad de México, Toluca de Lerdo, and Tlalpala.

En ella se mostrara en la parte central aquellos datos que son nuevos y que aún no se autentifica que sean o no correctos. Para permitir un dato se dará clic en la paloma verde o, en caso de ser incorrecto, dar clic en tache.

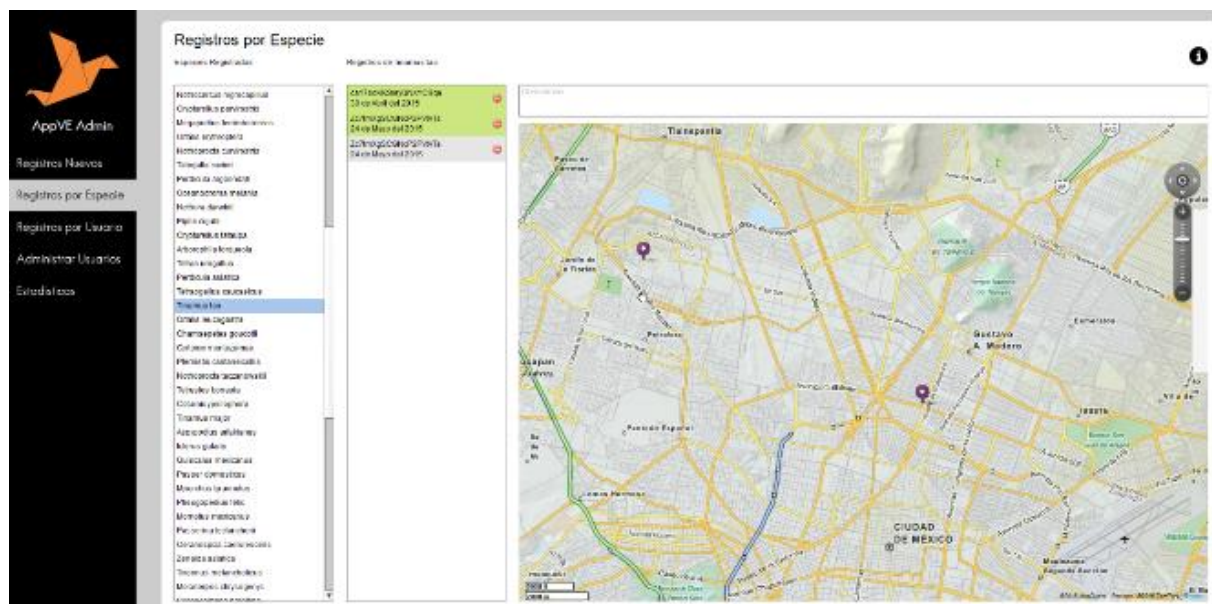
Especie	Usuario	Fecha	Accion
Nothura darwini	Zc7tmXgSCGNcP2PVVTs	12 de Mayo del 2015	 
Tinamus tao	Zc7tmXgSCGNcP2PVVTs	24 de Mayo del 2015	 

Estos datos de lado derecho se mostraran la ubicación geográfica de cada dato para una mejor visualización del dato.

En la parte superior del mapa también tenemos un panel de texto el cual contendrá comentarios en caso de así tenerlo la observación.

3.2 Análisis de datos

Para el análisis se podrá dar clic en el panel lateral en Registro por Usuario y Registro por Especie. Se mostrara una pantalla como la siguiente:

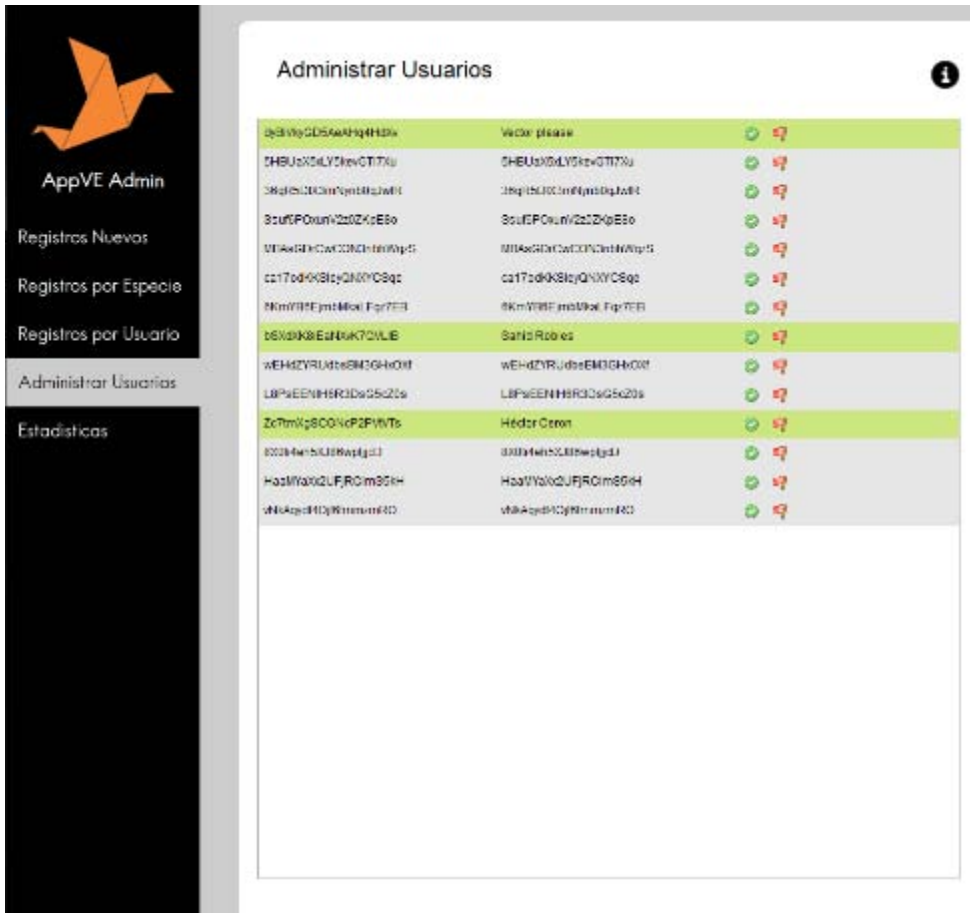


Catalogando por columnas para el caso de especies, la primera columna será el nombre de la especie y la segunda el usuario. Para el caso de usuarios, la primera columna será usuarios y la segunda especies. Al dar clic sobre la primera columna para ambos casos, se mostrara el general de todos los registros con ese parámetro y su ubicación en el mapa. Si se requiere un dato en particular se dará clic en el dato

particular a apreciar. Así mismo en caso de que el avistamiento tuviera una observación se mostrara en el cuadro de texto arriba del mapa.

3.3 Administración de usuarios

Para ingresar a la administración de usuarios, en el panel lateral se deberá de dar clic en Administración de Usuarios y mostrara una pantalla como la siguiente.



En esta pantalla el administrador decidirá que usuario o cual no serán de confianza así como bloquear los datos de algún usuario.

Para que un usuario sea de confianza debe de darse clic en el botón con una paloma verde. En caso de querer bloquear se dará clic sobre la bandera roja.

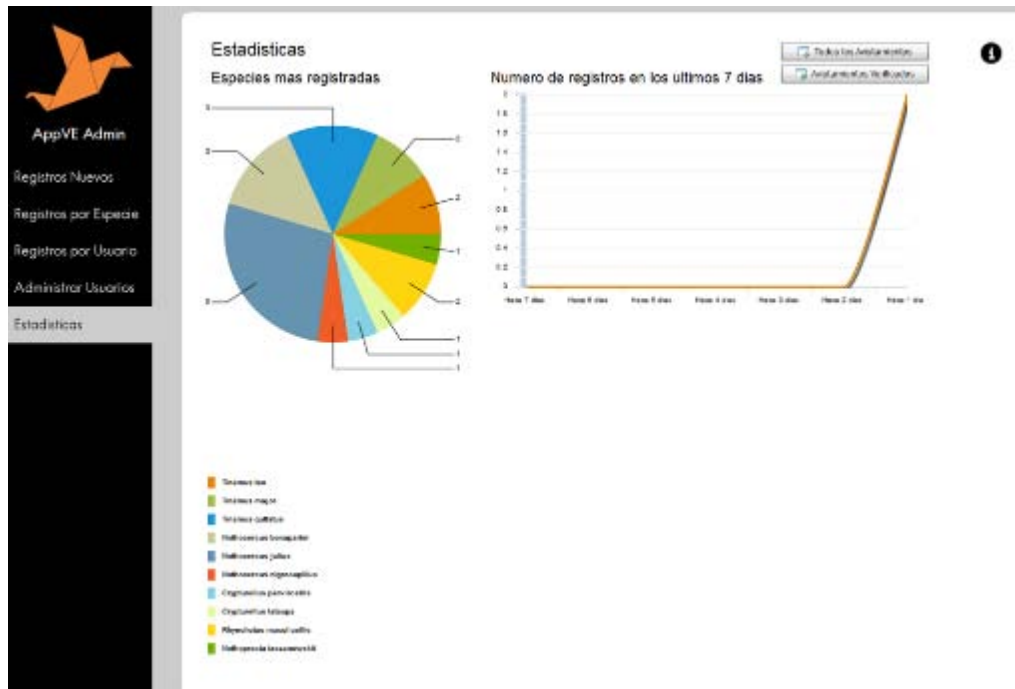
Administrar Usuarios

Zc7tmXgSCGNcP2PVVtS	Héctor Ceron		
5HBUaX5xLY5kevGTI7Xu	5HBUaX5xLY5kevGTI7Xu		
36qR5DIX3mNynb0qJwIR	36qR5DIX3mNynb0qJwIR		

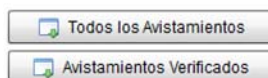
Cada que se escoja alguna opción el usuario asignado a este se pondrá de color distinto. Verde para aprobados, rojo para rechazados y gris para poder dar de alta.

3.4 Exportación de datos.

Para la exportación de datos se tendrá que dar clic en el panel lateral en la opción Estadísticas. Se mostrara una pantalla como la siguiente.

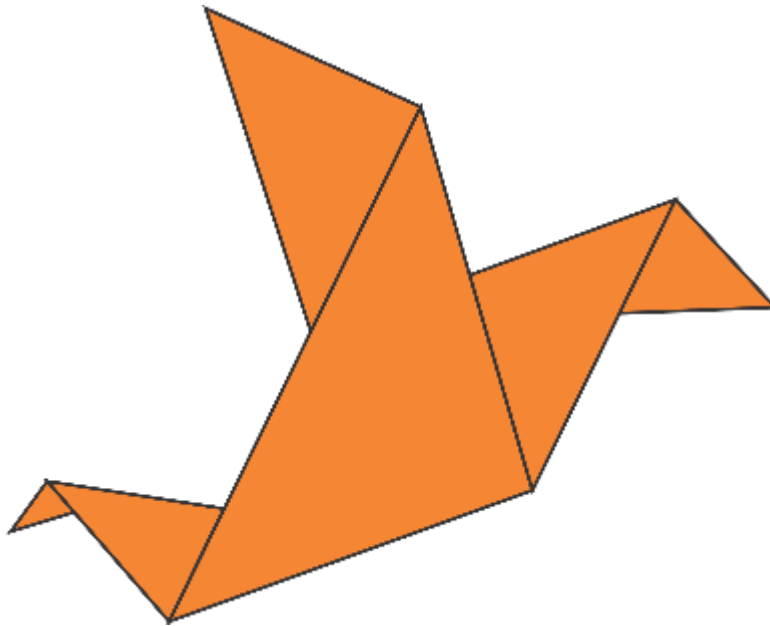


Se mostraran adicional unos gráficos los cuales mostraran los avances, registros y uso de esta aplicación. Para la exportación de datos a un lugar externo se dará clic en cualquier de los siguientes botones.



ANEXO 7 MANUAL DE INSTALACIÓN

PARA EL SISTEMA



AppVe

1. Introducción

Este documento contiene toda la información necesaria para la instalación exitosa del sistema AppVe. La finalidad del documento es ser una guía para la instalación del sistema orientada a gente con conocimientos de manejo de sistemas en la sección de administrador y para cualquier persona en la versión móvil.

2. Requerimientos iniciales

Para la aplicación móvil del sistema, el móvil debe tener los siguientes requerimientos:

- Sistema Operativo Android versión 2.3.6 o posterior
- Espacio en memoria mínimo de 50 MB
- Permisos para instalar aplicaciones de fuentes desconocidas

Para el panel de administrador, el servidor debe tener instalado con las variables del sistema definidas:

- Apache HTTP Server versión 2.4 o superior

Para ver como instalarlo ir a:

<http://httpd.apache.org/docs/2.4/>

- PhpMyAdmin versión 4.0 o superior

Para ver como instalarlo ir a:

<http://docs.phpmyadmin.net/en/latest/setup.html>

3. Instalación del sistema

3.1 Versión Móvil

1. Transferir el archivo AppVe.apk al teléfono inteligente en donde se realizara la instalación.



2. Ingresar dentro del teléfono inteligente a los archivos internos y seleccionar AppVe.apk

3. Al seleccionar se desplegara una pantalla como la siguiente.



Dar clic en instalar y la aplicación quedara instalada en el teléfono inteligente.

3.2 Servicio del administrador

Se compondrá de dos fases, la instalación primero de las bases de datos y después el panel de administrador. Es fundamental que sea este el orden o de lo contrario el sistema podría presentar fallas.

Bases de datos

1. Ingresar a panel de administrador dentro del servidor.
2. Una vez adentro buscar el módulo de 'MySQL Databases' y dar clic.



3. En la parte de arriba se encuentra la opción de crear una nueva base de datos. Introducir aves y dar clic en Crear.
4. Una vez creada la base de datos se procede a crear un usuario con su respectiva contraseña, esta opción se ubica en la parte central de la página. Se recomienda usar una contraseña fuerte que contenga números y símbolos. Se recomienda apuntar el nombre de la base de datos, el usuario y la contraseña en un lugar seguro ya que se utilizar en los siguientes pasos.
5. Ya creado el usuario con su contraseña se procederá a agregar el usuario a la base de datos. Esta opción se encuentra en la parte inferior de la página. Se selecciona el usuario y la base de datos y se le da clic en Agregar.
6. Se abre una nueva página donde se elegirán los permisos del usuario. Para propósitos prácticos se le da autorización a todos los permisos y se le da clic en Aceptar.
7. Regresar a la página principal del administrador.
8. Buscar y entrar al módulo de phpMyAdmin.



9. Seleccionar la base de datos que se utilizara de la lista desplegable del lado izquierdo de la página.

10. Hacer clic en el botón de Importar que se encuentra en la parte superior de la página.
11. Importar el archivo .sql que se le proporcione y dar clic en Aceptar.
12. Abrir los siguientes archivos .php con un editor de texto plano: ./admin/config.php y ./AppVe/agregarAvistamiento.php
13. Cambiar las variables que corresponden al nombre de la base de datos, usuario y contraseña por los valores creados en los pasos 3 y 4.

Panel Administrativo en el Servidor

1. Descargar e instalar un cliente FTP compatible con el sistema operativo del usuario, se recomienda FileZilla, es gratuito y está disponible para todos los sistemas operativos más comunes en la siguiente página <https://filezilla-project.org/>.
2. Una vez instalado el cliente FTP, ejecutarlo y crear una nueva conexión al servidor. Llenar los datos que se piden: usuario, contraseña y dirección ip del servidor incluyendo el puerto 21.
3. Ya que la conexión con el servidor ha sido establecida navegar al folder public_html en el servidor.
4. Arrastrar el contenido de la carpeta public_html del archivo .zip que se proporcionó hacia la carpeta public_html del servidor.
5. Cuando la transferencia de todos los archivos haya terminado, verificar que la estructura sea la correcta y cerrar la conexión al servidor.



**MUSEO DE ZOOLOGÍA "ALFONSO L. HERRERA"
 DEPARTAMENTO DE BIOLOGÍA EVOLUTIVA
 FACULTAD DE CIENCIAS
 UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**



**MI Alejandro Velázquez Mena
 Jefe de la Carrera de Ingeniería Computación
 Facultad de Ingeniería, UNAM
 PRESENTE**

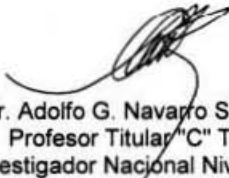
Estimado Maestro Velázquez,

Por este medio queremos expresar que el trabajo de tesis que realizó el C. **Héctor Cerón Guevara** titulado "*Diseño de un sistema informático para la localización de aves de México*" bajo la dirección del Dr. Daniel Trejo Medina, fue desarrollado en colaboración con personal académico del Museo de Zoología de la Facultad de Ciencias y está siendo ampliamente utilizado en diferentes proyectos desarrollamos, pues ha aportado valor a la operación de las distintas actividades de campo y gabinete para académicos y estudiantes del Museo, facilitándonos la administración de datos y mejorando la exactitud de las observaciones que realizamos en campo. Los datos obtenidos a través de su sistema tendrán un impacto positivo en las publicaciones científicas producto de los proyectos de investigación a nuestro cargo.


Se extiende la presente a petición del interesado y para dar referencia del uso y aporte de su trabajo en nuestro Museo.

ATENTAMENTE

Ciudad Universitaria, 1 de agosto del 2105


 Dr. Adolfo G. Navarro Sigüenza
 Profesor Titular "C" TC Def
 Investigador Nacional Nivel III




 Biól. Alejandro Gordillo Martínez
 Técnico Académico TC
 Responsable de Bases de Datos MZFC

APARTADO POSTAL 70-399, MÉXICO D. F. 04510, MÉXICO
 TEL. (+52 55) 5622 5435 FAX (+52 55) 5622 4828

Imagen 29 Carta de confirmación del uso y aporte valor de la tesis y su producto.
 Fuente: Carta del Museo de Zoología