

# **DESARROLLO DEL SISTEMA INTEGRAL FOTO-TRAMPA PARA LA DETECCIÓN DE FAUNA**

## *DEVELOPMENT OF THE INTEGRAL PHOTO-TRAP SYSTEM FOR WILDLIFE DETECTION*

**Laura Santos Luna**

TecNM / Instituto Tecnológico Superior de Zacapoaxtla, México  
*26laura2013B@gmail.com*

**José Miguel Méndez Alonso**

TecNM / Instituto Tecnológico Superior de Zacapoaxtla, México  
*xtrem\_jam@hotmail.com*

**Luis Alberto Espejo Ponce**

TecNM / Instituto Tecnológico Superior de Zacapoaxtla México  
*jcensan@hotmail.com*

**Recepción:** 31/octubre/2019

**Aceptación:** 9/diciembre/2019

### **Resumen**

La estimación de la abundancia poblacional es un problema tanto teórico como práctico en el manejo y la conservación de la fauna silvestre. El propósito de esta investigación es monitorear a los animales en su hábitat sin perturbarlos con actividades humanas, para su conservación, por lo que, este trabajo contiene la investigación para el desarrollo del sistema integral foto-trampa para la detección de fauna, el cual proporciona imágenes a través de una cámara que se activa al momento en que el sensor envía información de una posible especie, teniendo como resultado la captura, esto con la finalidad de dar estimaciones de abundancia y densidad de la población. Así mismo contiene el desarrollo de una aplicación web embebida en el sistema, con la que el usuario pueda administrar los datos del sensor como la máxima, mínima distancia y la configuración de la cámara (brillo, contraste, estabilización) con la intención de facilitar la utilización de las foto-trampas.

**Palabra(s) Clave:** Detección de fauna, Foto-trampa, implementación, sistema embebido

## **Abstract**

*The estimation of population abundance is a theoretical and practical problem in the management and conservation of wildlife. The purpose of this investigation is to monitor the animals in their habitat without disturbing them with human activities, for their conservation, so, this work contains the research for the development of the integral photo-trap system for the detection of fauna, which provides images through a camera that is activated at the time the sensor sends information of a possible species, resulting in capture, this in order to give estimates of abundance and population density. It also contains the development of a web application embedded in the system, with which the user can manage the sensor data such as maximum, minimum distance and camera settings (brightness, contrast, stabilization) with the intention of facilitating the use of photo-traps.*

**Keywords:** *Detection, embedded system, implementation, photo-trap.*

## **1. Introducción**

México es uno de los países más diversos del planeta desde el punto de vista biológico, debido a que, pueden encontrarse en el país casi todos los tipos de vegetación que existen en el mundo [CONABIO, 2006]. Su compleja fisiografía e historia geológica y climática han creado una variada gama de condiciones que hacen posible la coexistencia de especies de origen tropical y boreal, y que también han permitido con el paso del tiempo, una intensa diversificación de muchos grupos taxonómicos en las zonas continentales de su territorio y a lo largo de sus zonas costeras y oceánicas [Espinosa *et al.*, 2008]. Es por ello, que las zonas más destacables por su riqueza y número de endemismos son las regiones Neártica y Neotropical, como el Eje Neovolcánico y las Sierras Madre Oriental y Occidental [Koleff *et al.*, 2008]. Así mismo existen, especies que sólo se encuentran en nuestro territorio, es decir, seres vivos, que incluyen tanto la flora como la fauna, cuya distribución se restringe a una determinada zona geográfica y que a su vez son conocidas como especies endémicas, las cuales complementan de manera importante la riqueza biológica. Es por esta causa, que es necesario el monitoreo y la clasificación de las especies, para conocer una estimación poblacional de fauna.

Sin embargo, muchas especies son difíciles de observar debido a sus patrones de conducta, bajas densidades y carácter elusivo, entre otras razones [Wilson *et al.*, 1996] [Karanth *et al.*, 2004], por lo que es necesario, implementar técnicas y protocolos que nos permitan obtener suficiente información, sobre sus poblaciones, que sea útil en la toma de decisiones, para su manejo y conservación a largo plazo. Lo que da como resultado una tarea difícil, en ocasiones imposible, debido a que muestrear o monitorear poblaciones de animales con métodos tradicionales, basados en detecciones visuales directas o con métodos que involucran la captura, marcaje y radio-seguimiento de los individuos, suelen ser costosos y peligroso de implementar si no se tiene el conocimiento adecuado. Por lo que, una alternativa factible es la utilización de foto-trampas, a causa, de que se han vuelto una herramienta muy útil para el muestreo y el monitoreo de poblaciones de especies de carácter críptico o raras, porque permiten ampliar nuestras observaciones de las especies en el tiempo y el espacio sin interferir con su conducta [Lynam 2002] [Karanth y Nichols, 2002] [Karanth *et al.*, 2004a] [Silver, 2004] [Kays y Slauson, 2008], generando así información muy valiosa sobre la biología y ecología de estas especies, que de otra manera sería difícil de obtener. En cuanto, a la realización del proceso de detección entre la foto-trampa y los animales, ha sido basado en la distancia a la que la cámara detecta el movimiento de los de los mismos. Por lo general, las cámaras suelen detectar el movimiento en un rango comprendido entre 0 a 15 o 20 metros.

El detectar a los animales es de gran importancia, por lo que cuanto mayor distancia de detección se tenga, más se facilitará el muestreo [Sanz, 2018]. Además, se permite la identificación exacta de especies e incluso en algunos casos de individuos, diferenciación de animales diurnos y nocturnos, la confirmación de especies cuyas huellas no se diferencian y existe la posibilidad de registrar algunas especies no detectadas por otros métodos [Maffei *et al.*, 2002].

Por lo que el objetivo principal de esta investigación es generar un sistema integral foto-trampa la cual sirva como herramienta de apoyo para el monitoreo y clasificación de especies, que sea capaz de capturar imágenes en tiempo real y que, a su vez, contenga un sistema embebido, el cual permita administrar los datos

obtenidos, así como la configuración de la misma. La importancia de esta investigación radica en la protección, cuidado, manejo y mantenimiento de los ecosistemas, los hábitats, las especies y las poblaciones de la vida silvestre, con la intención de lograr las condiciones óptimas de conservación de la fauna.

## 2. Métodos

Para esta investigación se utilizó la metodología de investigación Tecnológica, la cual, está basada en los autores Pacheco y Cruz [2006] y consta de los pasos indicados en la figura 1.



Fuente: Elaboración Propia.

Figura 1 Pasos de la metodología de investigación Tecnológica.

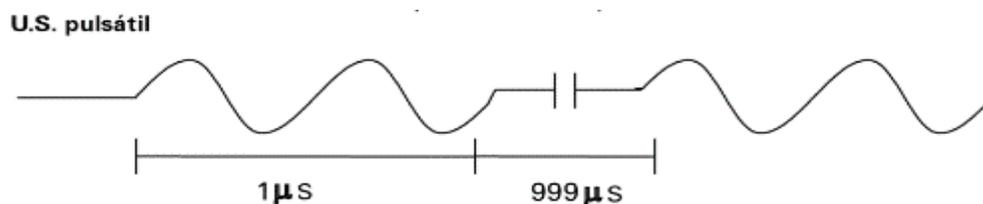
Así mismo, se usó el enfoque cualitativo que es la recogida de información basada en la observación de comportamientos naturales, discursos, respuestas abiertas para la posterior interpretación de significados.

Es por ello, que el presente proyecto se consideró la elección del tipo de sistema de foto-trampa, el cual fue el sistema pasivo. Este sistema detecta por medio de un sensor (receptor) el movimiento o el calor generado por un animal u objeto dentro de un área conocida como “zona de sensibilidad” [Chávez *et al.*, 2013]. Para la creación y selección del sensor, se realizó una investigación basada en ondas

ultrasónicas, debido a que, utilizan las técnicas pulso-eco, que dan una zona específica de sensibilidad. La siguiente información fue el apoyo para la creación del prototipo.

### Teoría de las ondas ultrasónicas.

Las ondas ultrasónicas, son una forma de radiación no ionizante, que, con frecuencia alta, son transmitidas a través de un medio. Junto a las propiedades de estas, se une la frecuencia y la intensidad para la definición de las propiedades de la transmisión. A su vez, estas se asemejan a un haz de luz, en el sentido de que pueden ser enfocadas, refractadas, reflejadas o diseminadas, a nivel de las interfaces entre los medios diferentes. Para la medición de estas, se empleó al sonido, debido a que su medida es el decibelio (dB), y permite medir la variación de intensidad de los pulsos ultrasónicos del instrumento, mediante el centro de voltaje aplicado al transductor.



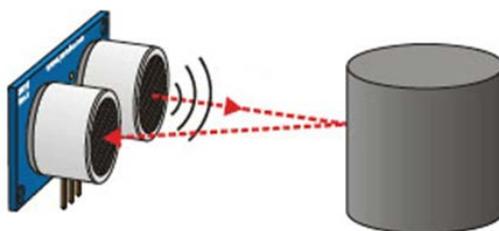
Fuente: [Mad, 2006].

Figura 2 Ondas Ultrasónicas.

Esta teoría consiste, en que el pulso ultrasónico y la onda reflejada (el eco), se reciban al mismo transductor después de un tiempo determinado; es decir, el frontal del transductor registra la señal durante ciertos ciclos y posteriormente permanece en reposo durante un intervalo de tiempo, antes de emitir el siguiente pulso ultrasónico. Al intervalo entre el comienzo y el siguiente se denomina periodo de repetición de pulsos (PRP), tal como se muestra en la figura 2.

La velocidad de las ondas ultrasónicas depende de la densidad y la facilidad de comprensión del medio a través del cual se transmiten. Es por esta teoría, lo que permitió la elección del sensor de ultrasonidos HC-SR04, que es, un dispositivo para medir distancias. Su funcionamiento se basa en el envío de un pulso de alta

frecuencia, no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia el sensor, que dispone de un micrófono adecuado para esa frecuencia y que, a su vez, mide el tiempo entre los pulsos, así se estima la distancia del objeto, contra la superficie del impacto como se muestra en la figura 3. El rango de medición teórico del sensor es de 2 a 400 cm, con una resolución de 0.3cm.



Fuente: [Llamas, 2015].

Figura 3 Función del sensor.

### Selección de la cámara

Para la captura de imágenes, se decidió utilizar la picamera que es una cámara original para Raspberry Pi (figuras 4 y 5), cuenta con 8 megapíxeles, es capaz de lograr imágenes estáticas de 3280 x 2464 píxeles, tamaño óptico de 1/4 y con una dimensión de 25 x 23 x 9 mm, algunos de los ajustes que permite realizar son el contraste, nitidez, brillo, saturación, sensibilidad (ISO), estabilidad, balance de blancos, modo de exposición, rotación, calidad de la imagen entre otros, lo que permite la obtención de imágenes de acuerdo a las necesidades que se requieran.



Fuente: [Mercado Libre, 2019].

Figura 4 Picamera.



Fuente: [Amazon, 2019].

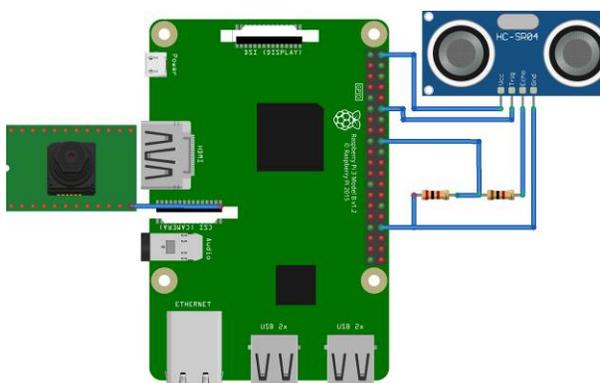
Figura 5 Raspberry.

La elección se llevó a cabo en base a que, se decidió utilizar la microcomputadora Raspberry Pi para el procesamiento de los datos, debido a que soporta varios

componentes necesarios en un ordenador común y es capaz de comportarse como uno.

### Construcción del Prototipo

Una vez, seleccionados los instrumentos para la creación del prototipo, se diseñó el circuito en el software Fritzing, este es un hardware de código abierto que hace que la electrónica sea accesible como material creativo para cualquier persona. También, ofrece un sitio web comunitario y servicios de procesamiento en Arduino, por lo que, permite a los usuarios documentar sus prototipos, compartirlos con otros, enseñar electrónica en un aula y diseñar y fabricar proyectos profesionales. El resultado de ello se muestra en la figura 6.

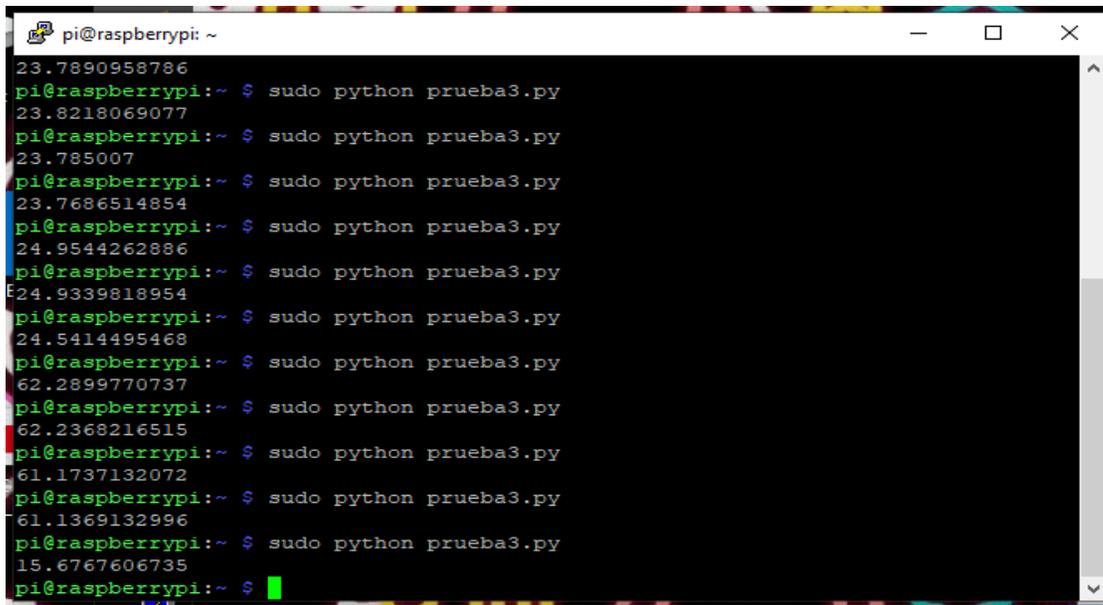


Fuente: Elaboración propia.

Figura 6 Diseño de la conexión (Circuito).

### Desarrollo y obtención de los datos.

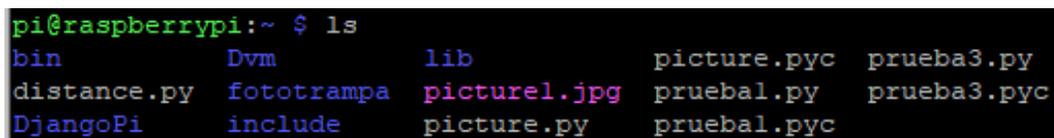
En este apartado, se programaron múltiples algoritmos propios, estos permiten la toma de la fotografía y la detección de los objetos, así como la lectura de los datos. Dichos algoritmos se desarrollaron con el lenguaje Python y se ejecutaron en el mismo sistema de la Raspberry (figura 7). Esto llevó a la visualización de los alcances del sistema, del sensor y de la cámara, a su vez, se conoció la confiabilidad de los datos que ofrece el sensor, porque, puede obtener la distancia en metros o centímetros del animal o del objeto en sí según la preferencia. Como función principal se tiene que al detectar al objetivo se manda la señal para la captura de la imagen. Esta imagen es almacenada en el sistema (figura 8).



```
pi@raspberrypi: ~
23.7890958786
pi@raspberrypi:~ $ sudo python prueba3.py
23.8218069077
pi@raspberrypi:~ $ sudo python prueba3.py
23.785007
pi@raspberrypi:~ $ sudo python prueba3.py
23.7686514854
pi@raspberrypi:~ $ sudo python prueba3.py
24.9544262886
pi@raspberrypi:~ $ sudo python prueba3.py
24.9339818954
pi@raspberrypi:~ $ sudo python prueba3.py
24.5414495468
pi@raspberrypi:~ $ sudo python prueba3.py
62.2899770737
pi@raspberrypi:~ $ sudo python prueba3.py
62.2368216515
pi@raspberrypi:~ $ sudo python prueba3.py
61.1737132072
pi@raspberrypi:~ $ sudo python prueba3.py
61.1369132996
pi@raspberrypi:~ $ sudo python prueba3.py
15.6767606735
pi@raspberrypi:~ $
```

Fuente: Elaboración propia.

Figura 7 Ejecución del algoritmo de manera desde la Shell.



```
pi@raspberrypi:~ $ ls
bin          Dvm          lib          picture.pyc  prueba3.py
distance.py  fototrampa  picture1.jpg pruebal.py  prueba3.pyc
DjangoPi    include     picture.py   pruebal.pyc
```

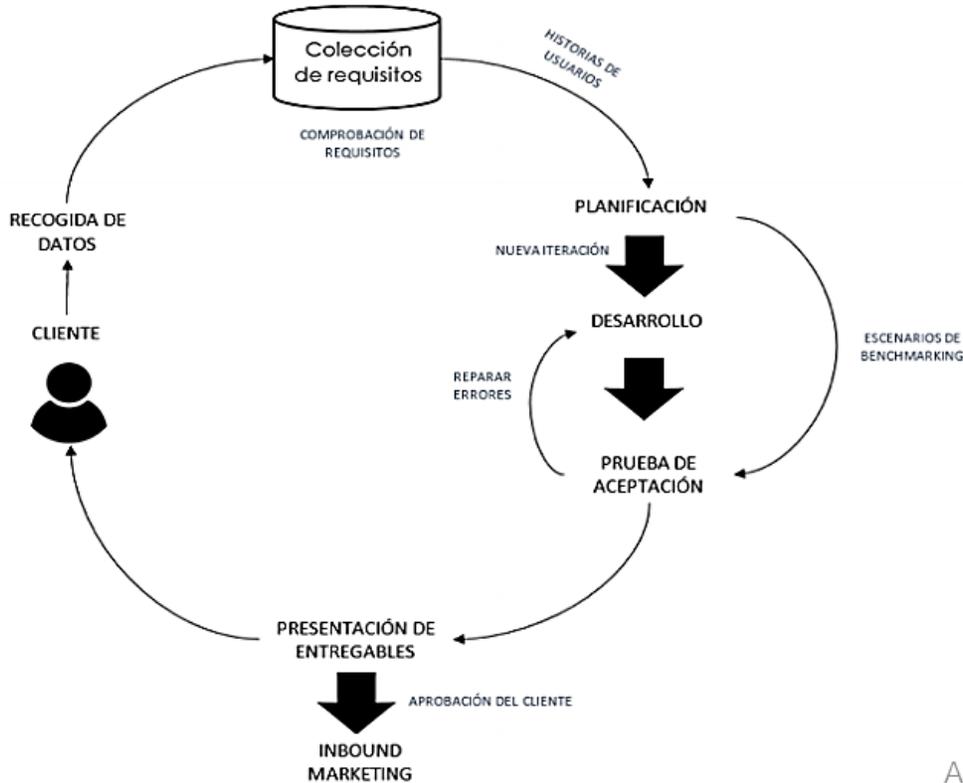
Fuente: Elaboración propia.

Figura 8 Imagen Almacenada.

Se decidió sólo guardar una imagen para no hacer redundancia de almacenamiento, debido a que la presentación final de los datos, se llevaría a cabo en un sistema web que permitiera la administración del prototipo.

### Aplicación web embebida en el sistema

Para el inicio del desarrollo de esta aplicación se investigó las ventajas y desventajas de las metodologías para desarrollo web, lo que dio como resultado la selección de la metodología SNAIL (Software Nativo de Arquitectura Iterativa Lógica), que es una metodología híbrida para el desarrollo web, esta se basa en la simplicidad, comunicación y planificación del código desarrollado. Si bien el ciclo de vida de un proyecto SNAIL es muy dinámico, se puede separar en las siguientes Fases, representadas en este modelo como se muestra en la figura 9.



Ac  
Fuente: [3Ciencias, 2019].

Figura 9 Fases de la metodología SNAIL.

### Construcción de la aplicación web embebida

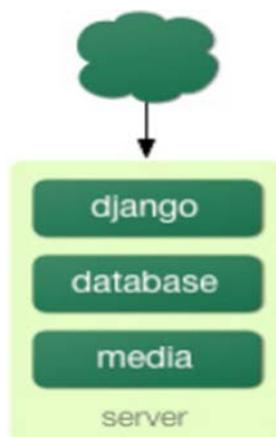
Para la construcción de la aplicación web se utilizó Django, framework de desarrollo web, que permite construir en profundidad y de forma dinámica, sitios interesantes en un tiempo extremadamente corto, también, proporciona abstracciones de alto nivel a patrones comunes del desarrollo web, agrega atajos para tareas frecuentes de programación, pero lo más importante es que, está basado en la metodología Modelo Vista Controlador la cual, para definir y acceder a los datos (el modelo) está separado del pedido lógico de asignación de ruta (el controlador), que a su vez, está separado de la interfaz del usuario (la vista). Una ventaja clave de este enfoque es que los componentes tienen un acoplamiento débil entre sí. Eso significa que cada pieza de la aplicación Web que funciona sobre Django tiene un único propósito clave, así mismo, puede ser modificado independientemente sin afectar las otras piezas. Además, ofrece la integración de base de datos, en donde se puede crear tablas relacionadas entre sí, para la

manipulación de variables y datos. Lo que resultó más apremiante para la fusión con el prototipo es que este framework está totalmente escrito en el lenguaje Python lo que facilitó su uso. Ya estando el framework instalado se creó un proyecto llamado fototrampa dentro del cual se crearon varias aplicaciones, que a continuación se describen:

- Core: Es el núcleo de la aplicación web embebida, en ella recaen las plantillas base que se estarán utilizando en las demás aplicaciones, así como, la creación de los modelos que administrarán la página principal.
- Information: Esta aplicación, como su nombre lo indica es solo para la información de la fototrampa, como el modelo, número de serie, autor entre otros.
- Monitoring: Es la aplicación que administrará al sensor y a la cámara, también contiene las interfaces para la visualización de las imágenes y la lectura del sensor (la distancia), así como los ajustes.
- Registration: Esta contiene las interfaces para inicio, registro y perfil del usuario. También contiene las reglas para un registro exitoso.

### Despliegue de la Aplicación.

Como es un sistema embebido se utilizó la configuración mostrada en figura 10 para poner al sitio web en producción.



Fuente: [Django, 2015].

Figura 10 Modelo de ejecución (Servidor único).

Se utilizaron las siguientes aplicaciones para poner en marcha el servicio:

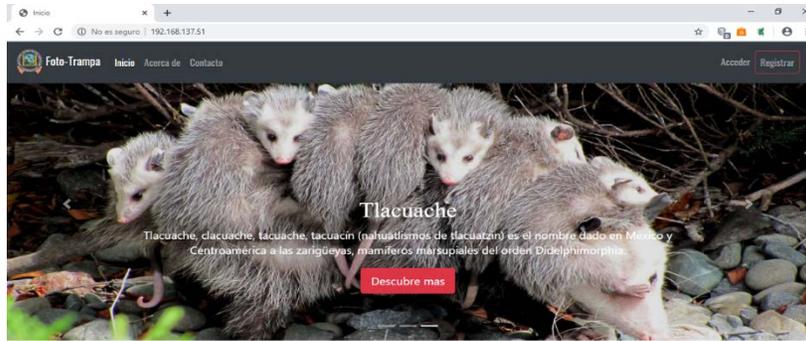
- Virtual environment: Se utilizó, con el objetivo de aislar recursos como librerías y el entorno de ejecución del sistema principal.
- Unicorn: es un servidor WSGI HTTP para Python, este es compatible con el framework Django. Se utilizó, porque, permite administrar peticiones simultáneas que la aplicación reciba.
- Supervisor: es un sistema cliente/servidor que permite a sus usuarios monitorear y controlar un número de procesos en sistemas operativos tipo unix; se utilizó, debido, a que facilita el manejo de un número de programas o procesos en ejecución por un largo tiempo y provee una interfaz consistente mediante la cual, esos procesos se pueden monitorear y controlar.
- Nginx: Ofrece una arquitectura asíncrona y controlada por eventos, es uno de los servidores más confiables para velocidad y escalabilidad. Se utilizó principalmente porque puede procesar miles de solicitudes sin dificultad.

### **3. Resultados**

Se ha diseñado e instrumentado un prototipo de foto-trampeo el cual fue construido con un sensor y una cámara, se implementó el sistema Raspbian nativo del microcontrolador Raspberry con el lenguaje de programación Python versión 3 que trae por defecto el sistema, para la lectura de los datos del sensor y la captura de imágenes se generaron algoritmos individuales y por medio de subprocesos se envía la información a la base de datos dicha información se muestra en la aplicación embebida desarrollada con el framework Django.

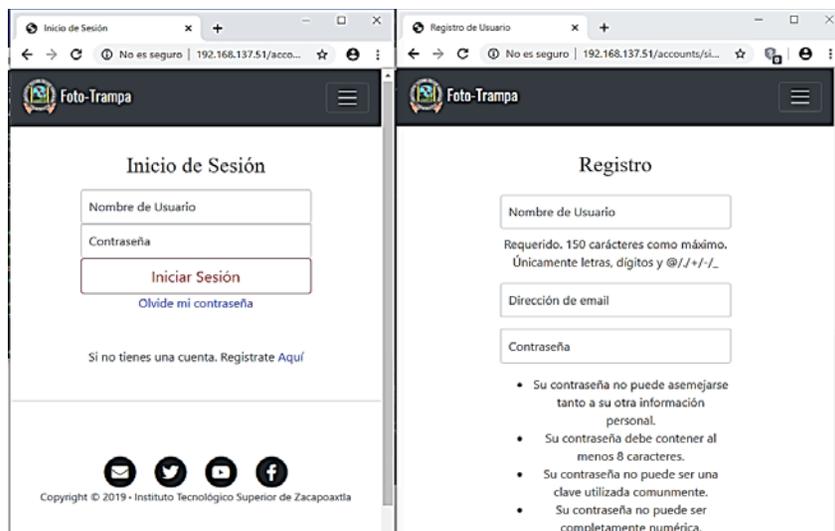
En las siguientes imágenes se muestra la funcionalidad de la aplicación embebida, iniciando con página principal contenida en la aplicación Core, como se muestra en la figura 11. En la figura 12 se muestra el inicio de sesión y el registro de usuarios, contenido en la aplicación "Registration", más adelante en la figura 13 se muestra el perfil del usuario. En la figura 14 se muestra la información de la fototrampa.

En las figuras 15, 16 y 17 se muestran las interfaces para la selección del sensor, para ver los registros de la distancia, y para su administración, en donde se define la máxima y mínima distancia, esto elaborado en la aplicación Monitoring.



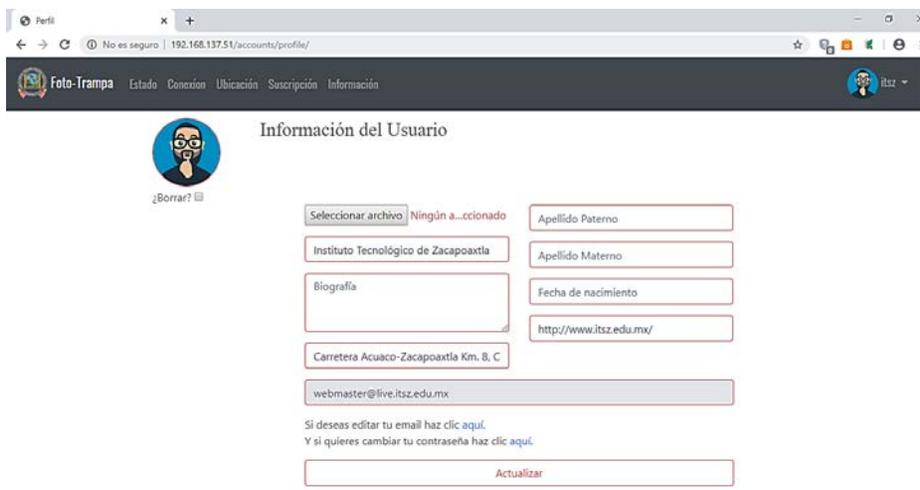
Fuente: Elaboración propia.

Figura 11 Página Principal.



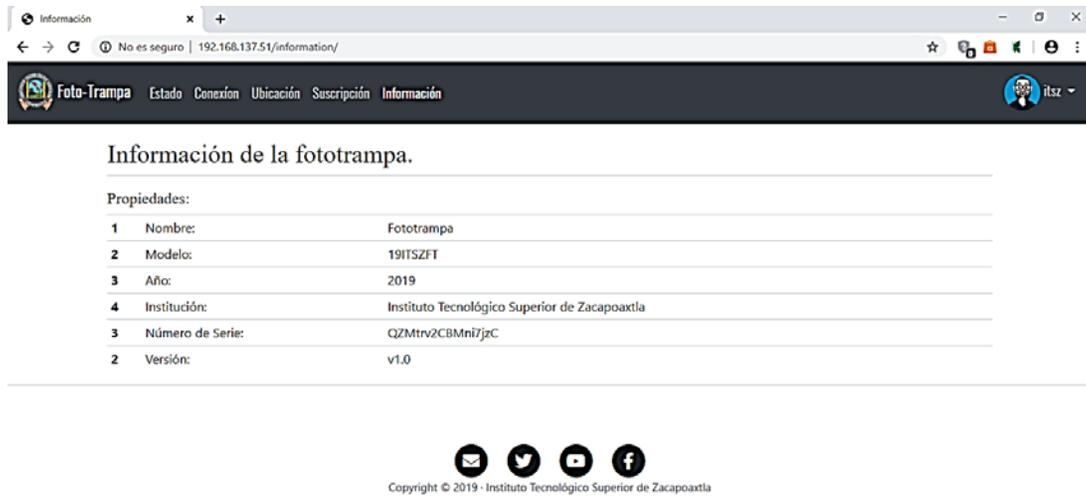
Fuente: Elaboración propia.

Figura 12 Inicio de sesión y registro de usuario



Fuente: Elaboración propia.

Figura 13 Perfil del usuario



Fuente: Elaboración propia.

Figura 14 Información de la Fototrampa.



Fuente: Elaboración propia.

Figura 15 Interfaz para la selección del sensor.



Fuente: Elaboración propia.

Figura 16 Interfaz para la administración del sensor

Distancia (cm)	Fecha
58.5813641548	11 de Noviembre de 2019 a las 20:38
15.1779174805	11 de Noviembre de 2019 a las 17:18
36.4605307579	11 de Noviembre de 2019 a las 17:16
28.0742406845	9 de Noviembre de 2019 a las 19:10
19.822883606	9 de Noviembre de 2019 a las 17:57
15.6399607658	9 de Noviembre de 2019 a las 17:56
53.6460876465	9 de Noviembre de 2019 a las 17:55
55.7191491127	9 de Noviembre de 2019 a las 17:55

Fuente: Elaboración propia.

Figura 17 Interfaz de Registro de Datos (distancia)

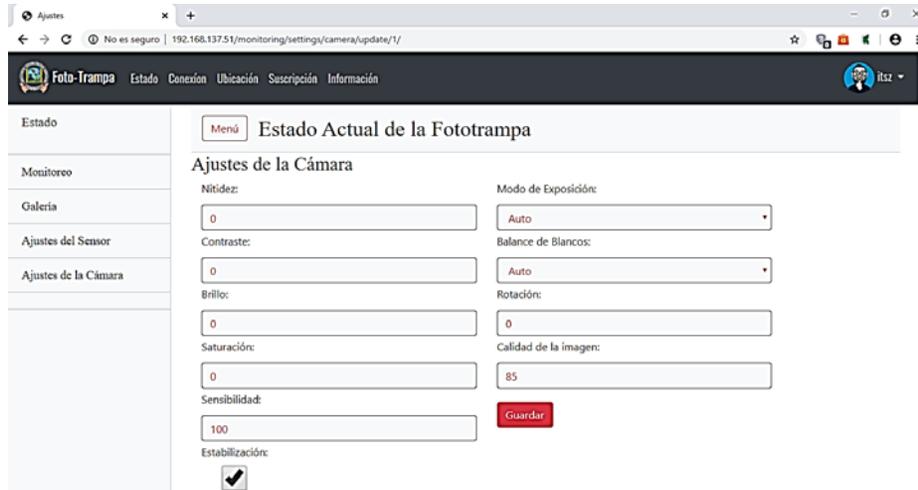
En la misma aplicación (Monitoring) se encuentran los ajustes de la cámara, la interfaz para seleccionarla y la interfaz para cambiar sus ajustes como se muestra en las figuras 18 y 19.

Así mismo, cuenta con una interfaz llamada galería (figura 20), esta muestra las imágenes capturadas, cuando se detectó al objeto, también se muestra la interface para cambiar el título de la imagen y posiblemente la imagen, si se desea subir imágenes externas (figura 21).

Nombre	Acción
Pi Camera	Editar

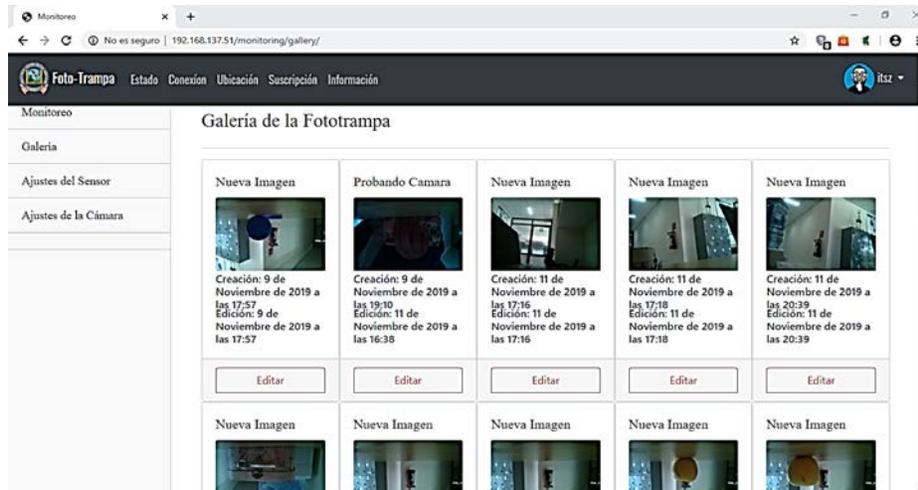
Fuente: Elaboración propia.

Figura 18 Interfaz para la selección de la cámara



Fuente: Elaboración propia.

Figura 19 Interfaz de los Ajustes de la cámara



Fuente: Elaboración propia.

Figura 20 Galería de la Fototrampa.



Fuente: Elaboración propia.

Figura 21 Interfaz para la edición de título o imagen.

## **4. Discusión**

Si bien hoy en día, las fotos de las foto-trampas permiten acercar a los investigadores y a los pobladores de las zonas a la fauna, generando una mayor conciencia sobre la biodiversidad, cambiado la forma de ver la naturaleza, es por esta causa, que en esta investigación se realizó un sistema integral foto-trampa para la clasificación y monitoreo de especies, en la que se le facilite al usuario la configuración y utilización de la herramienta. Sin embargo, se está consciente de que para el comienzo de las pruebas del prototipo se utilizará una cámara compatible con el sistema, pero se podría sustituir por una cámara digital comercial en donde se tendría una enorme cantidad de procesos de retoque informáticos que facilite la mejora de la calidad de la imagen, por los diversos ajustes que contienen: filtros, modificación de colores y contrastes, enfoque/desenfoco, añadido de textos, fechas o comentarios sonoros, modificación del tamaño entre otros. Además de utilizar el formato digital el cual se basa en el almacenamiento de la imagen mediante dígitos (números) que se mantendrán inmutables a lo largo del tiempo, con lo que la calidad de la imagen no disminuirá nunca. El modelo de este prototipo podría ser útil para replicarlo en otras organizaciones de la región y con ello mantener el monitoreo de la fauna silvestre.

### **Para el Futuro**

Se desea generar un servicio de streaming de la cámara a la página web, que permita la vista de los posibles animales que detecte, también se desea generar la ubicación, por medio de un marco web geográfico del propio framework de desarrollo llamado GeoDjango, debido, a que su objetivo es facilitar lo más posible la creación de aplicaciones web SIG y aprovechar el poder de los datos habilitados espacialmente y por último vincular las imágenes a un sistema para identificación y clasificación de fauna mediante visión por computadora.

## **5. Bibliografía y Referencias**

- [1] Alfaro Ferres L, et. al. (2001). Manual de Telepatología (Telepathology Manual).

- [2] Angelica Díaz-Pulido, Esteban Payán Garrido. (2012). Manual de fototrampeo: Una herramienta de investigación para la conservación de la biodiversidad en Colombia: <http://repository.humboldt.org.co/bitstream/handle/20.500.11761/31415/240.pdf?sequence=1&isAllowed=y>.
- [3] Andrés Illana, Diana Paniagua. (2015). Fototrampeo de Carnívoros en el Parque Natural de Izki: [http://www.faunadealava.org/adjuntos/faunadealavaDocumentos/49\\_archivo.pdf](http://www.faunadealava.org/adjuntos/faunadealavaDocumentos/49_archivo.pdf).
- [4] Asociación Internacional del Foto-Trampeo. (2008, 25 de agosto). Fototrampeo@Fototrampeo.Es. Obtenido de: <https://www.fototrampeo.es/>
- [5] Bahga, A. and Madiseti, V. (2014). Laboratory Training Guide: Cloud Computing: A Hands-On Approach. Hyderabad: Universities Press (India).
- [6] Blog.hackedexistence (2019). HACKED: <http://blog.hackedexistence.com>
- [7] CONABIO. (2018) Sistema de información sobre especies invasoras en México. Comisión Nacional para el Conocimiento y Uso de la Biodiversidad: [www.conabio.gob.mx/invasoras](http://www.conabio.gob.mx/invasoras). Fecha de consulta: agosto de 2012.
- [8] Cuauhtémoc Chávez, Antonio De La Torre, *et.al.* (2013). Manual De Fototrampeo Para Estudio De Fauna Silvestre El Jaguar En México Como Estudio De Caso: <http://biblioteca.semarnat.gob.mx/janium/Documentos/Ciga/Libros2013/CD002776.pdf>.
- [9] Django. (2015). 2005-2019 Django Software Foundation y colaboradores individuales. Django es una marca registrada de Django Software Foundation: <https://www.djangoproject.com>.
- [10] Espinosa, D., S. Ocegueda, et al. (2008). El conocimiento biogeográfico de las especies y su regionalización natural. En: Conabio. Capital Natural de México, Volumen I: Conocimiento actual de la biodiversidad. Comisión Nacional para el Conocimiento y Uso de la Biodiversidad. México.
- [11] Gómez Martínez, D. (2009). Centros Hospitalarios de Alta Resolución de Andalucía (CHARES). Alcalá de Guadaíra, Sevilla: MAD.
- [12] Karanth, K.U. y J.D. Nichols (eds.). (2002). Monitoring tigers and their prey, a manual for researchers, managers and conservationist in tropical Asia. Centre for Wildlife Studies, Bangalore, India.

- [13] Hochstein, L. (2014). *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. O'Reilly Media.
- [14] Holovaty, A. and Kaplan-Moss, J. (2009). *The definitive guide to Django*. Berkeley, Calif.: Apress.
- [15] Lynam, A. (2002). Métodos de trabajo de campo para definir y proteger poblaciones de gatos grandes: los tigres indochinos como un estudio de caso. Pp. 62 en: R. Medellín, C. Equihua, C. Chetkiewics, A. Rabinowitz, P. Crawshaw, A. Rabinowitz, K. Redford, J. G. Robinson, E. Sanderson y A. Taber (eds.). *El jaguar en el nuevo milenio: una evaluación de su estado, detección de prioridades y recomendaciones para la conservación de los jaguares en América*. Fondo de Cultura Económica, Universidad Nacional Autónoma de México y Wildlife Conservation Society. México.
- [16] Molina Ríos, J. R., Zea Ordóñez, M. P., Redrován Castillo, F. F., Loja Mora, N. M., ValarezoPardo, M. R. y Honores Tapia, J. A. (2018). *SNAIL, Una metodología híbrida para el desarrollo de aplicaciones web*. 3Ciencias: <https://www.3ciencias.com/libros/libro/snail-una-metodologia-hibrida-para-el-desarrollo-de-aplicaciones-web/>.
- [17] Serway, R., Faughn, J. and Escalona y García, H. (2001). *Física*. México: McGraw-Hill.
- [18] Wilson, K.R. y D.R. Anderson. (1985). Evaluation of a nested grid approach for estimating density. *Journal of Wildlife Management* 49:675-678.