

SISTEMA PARA LA DETECCIÓN Y CLASIFICACIÓN DE FAUNA MEDIANTE VISIÓN POR COMPUTADORA

SYSTEM FOR THE DETECTION AND CLASSIFICATION OF FAUNA THROUGH COMPUTER VISION

Fernando González González

Tecnológico Nacional de México / Instituto Tecnológico Superior de Zacapoaxtla, México
fergonzagonzalez@gmail.com

Luis Alberto Espejo Ponce

Tecnológico Nacional de México / Instituto Tecnológico Superior de Zacapoaxtla, México
jcensan@hotmail.com

Recepción: 31/octubre/2019

Aceptación: 17/diciembre/2019

Resumen

El monitoreo de especies de fauna silvestre es una herramienta que se usa para determinar cambios en el ecosistema, siendo un requisito indispensable en evaluaciones de impacto ambiental, identificando especies mediante muestreo sistemáticos.

La presente investigación desarrolló una aplicación web para interactuar con un sistema clasificador de fauna, que utiliza algoritmos de visión por computadora para realizar la clasificación, para su ejecución se siguió un proceso de cuatro etapas, que va desde la detección del problema hasta la elaboración del sistema. La aplicación web permite generar proyectos para guardar un banco de imágenes correspondientes a las especies a identificar, para después interactuar con el sistema clasificador que tiene como primera funcionalidad la extracción de características del conjunto de imágenes almacenadas, como resultado se tiene un sistema que va cumpliendo las especificaciones necesarias para interactuar con el sistema clasificador, además de validar los requerimientos en el desarrollo de software y generar una herramienta confiable.

Palabra(s) Clave: Fauna, inteligencia artificial, monitoreo, SNAIL, visión por computadora.

Abstract

Wildlife species monitoring is a tool used to identify changes in the ecosystem and is an indispensable requirement in an environmental impact assessment, identifying species through systematic sampling. The present investigation developed a web application to interact with a wildlife classification system, which uses computer vision algorithms to perform the classification, for its execution a four stages process was followed, ranging from the detection of the problem to the development of the system. The web application allows generating projects to save a bank of images corresponding to the species to be identified, and then interact with the classification system that has as its first functionality the extraction of characteristics of the set of stored images, as a result there is a system that is complying the specifications necessary to interact with the classification system, in addition to validating the requirements in software development and generating a reliable tool.

Keywords: *Wildlife, artificial intelligence, monitoring, SNAIL, computer vision.*

1. Introducción

La diversidad o diversidad biológica se define como la variabilidad entre organismos vivientes. Se han desarrollado diferentes parámetros para medir el indicador de especies que habitan en un ecosistema, con fines de conservar y monitorear las plantas y animales. Datos de Comisión Nacional de Áreas Naturales Protegidas (CONANP) afirman que México es considerado uno de los 17 países megadiversos, ocupando el puesto número 5, pues alberga grandes cantidades de animales y plantas [CONANP, 2018]. “México, como en muchos otros países en el mundo, hay creciente interés en avanzar decididamente por rutas de sostenibilidad. La capacidad de estimar la condición de los ecosistemas es clave, pues prácticamente toda acción humana implica su modificación. Por tal motivo, existe urgente necesidad por desarrollar nuevos enfoques analíticos que auxilien a equilibrar las múltiples y frecuentemente contradictorias metas públicas de desarrollo, al mismo tiempo que se mantiene la biodiversidad y la integridad ecosistémica” [CONABIO, 2019].

El monitoreo de fauna es una herramienta utilizada para determinar cambios en el ecosistema y es requisito indispensable en una evaluación de impacto ambiental, identificando especies y capturando información que tiene como finalidad generar base de datos, para, posteriormente se tenga la facilidad de monitoreo.

El protocolo, Monitoreo Comunitario de la Biodiversidad (BIOCOMUNI) obtiene información sobre la fauna presente en un ecosistema acudiendo al lugar de muestra dos veces al año, en temporada seca y temporada de lluvia, usando métodos de monitoreo como el conteo por puntos para las aves, transectos para localizar huellas y excrementos, y el fototrampeo para mamíferos terrestres o especies escurridizas [FMCN, CONAFOR, USAID y USFS, 2018].

Dada la necesidad de agilizar la velocidad de realizar el monitoreo y clasificación de las diferentes especies de fauna silvestre, se desarrolla un sistema que hace uso de visión por computadora, una de las disciplinas de inteligencia artificial.

La visión por computadora trata de imitar a la visión del ser humano, identificando e interpretando la información de manera estadística, que es fundamental para resolver un problema, además le permite procesar, analizar y capturar imágenes, modelándolas matemáticamente, siendo interpretadas por el ordenador que obtiene información respectiva, tomando una decisión sobre qué hacer con ellas de forma independiente [López, 2019].

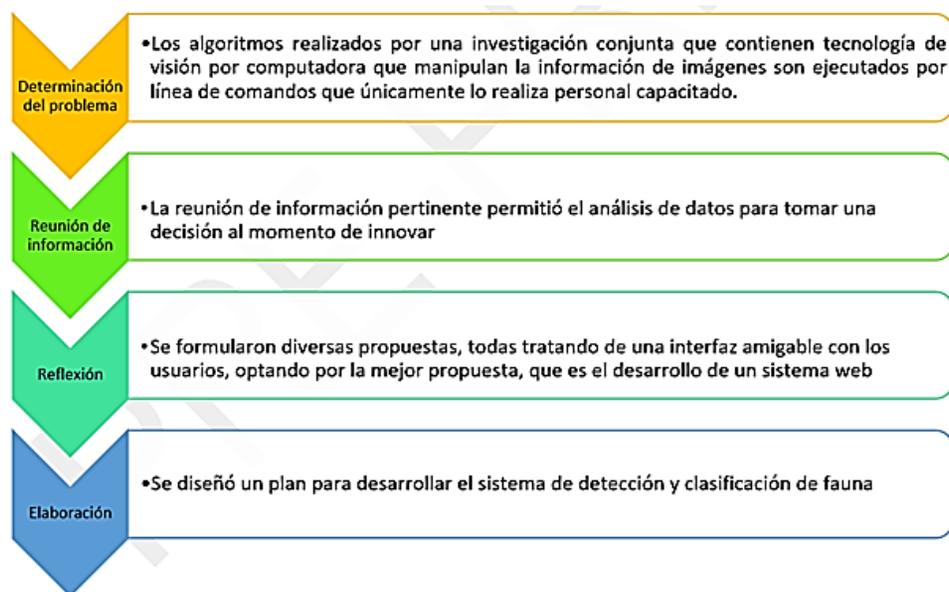
Esta tecnología hace uso de redes neuronales artificiales, son llamadas así por la similitud que tienen con las redes neuronales biológicas, modelando de manera simple el modo en que el cerebro humano procesa la información. [IBM, 2019]. INFAIMON en su blog menciona algunos ejemplos del uso de la visión por computadora como: en automoción, donde se inspecciona la fabricación y el ensamblaje de las piezas de vehículos, en alimentación pues llevan a cabo el control de calidad, entre algunos otros ejemplos más [INFAIMON, 2018].

El sistema que se desarrolló hace uso del resultado de una investigación conjunta, que desarrolló algoritmos que manipulan la información de la imagen, extrayendo sus características, describiendo los objetos, el paisaje y en algunos casos los colores, debido a que tienen implementada tecnología de visión por computadora, además, estos algoritmos son ejecutados por personal capacitado a través de

consola de comandos, lo que limita el acceso a usuarios para hacer uso de esta tecnología, es por ello que se desarrolló un sistema amigable con los usuarios que hace uso de estos algoritmos que identifican y clasifican especies de fauna de una manera más fácil y sencilla para el usuario.

2. Métodos

El proyecto se basa en la metodología de investigación tecnológica descrita por Celso de la Cruz Casaño quien propone una metodología de investigación tecnológica adaptada a las carreras de Ingeniería, definiendo seis pasos [Casaño 2016], sin embargo, la presente investigación definió cuatro de ellas que resultan fundamental para la creación de tecnología, en la figura 1 se muestran los pasos utilizados.



Fuente: Adaptado de [Casaño 2016].

Figura 1 Proceso desarrollado para el proyecto.

Para la construcción del sistema está basada en Software Nativo de Arquitectura Iterativa Lógica, por sus siglas SNAIL, una metodología usada en el desarrollo de sitios web que combina algunas prácticas existentes de metodologías como RUP, XP, Scrum y OOHDM, pues SNAIL, abarca las fases más importantes de estos modelos, formando un híbrido más eficiente [J. R. Molina Ríos et al. 2018].

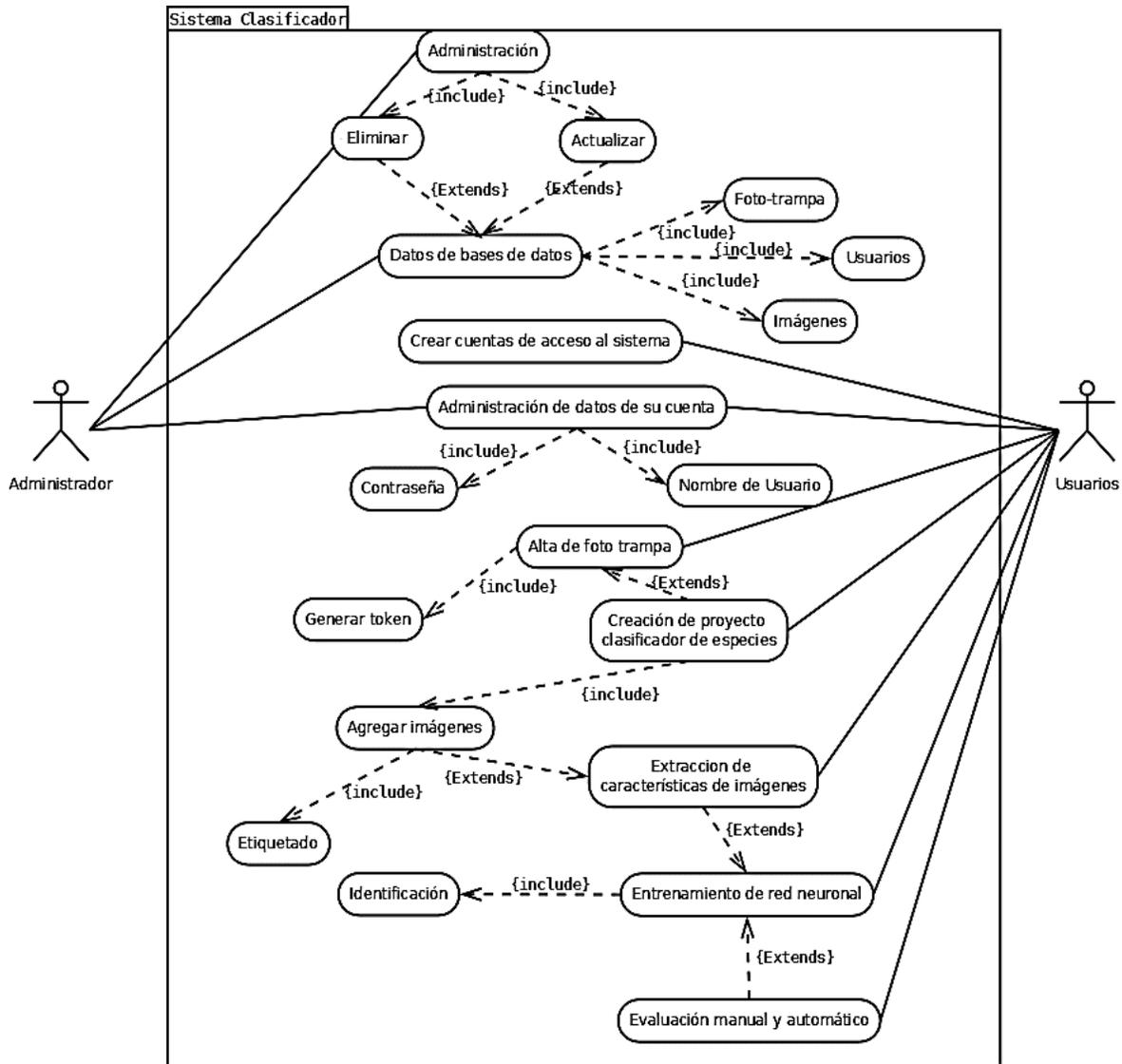
Con base en la metodología, el sistema se desarrolló en seis etapas:

- **La primera etapa es de requisitos:** en esta etapa se definió los actores o usuarios, llenando un documento de requerimientos donde se especifica lo que debe hacer el sistema.
- **La segunda etapa es de planificación:** en esta fase se realizó estimaciones de recursos, entregables y costos, todo ello planificado temporalmente. Todas las estimaciones tienen un tiempo limitado, desde que inicia el proyecto hasta que termina.
- **La tercera etapa es la de diseño:** en esta etapa se diseñó la base de datos, la interfaz de usuario (front-end) y el diseño conceptual.
- **La cuarta etapa es la de programación:** aquí, se escribió código para que el sistema haga lo que se requiere, fusionando la parte visual (front-end) y el código (back-end)
- **La quinta etapa es la de pruebas:** en esta etapa se comprobó el correcto funcionamiento del sistema, verificando que cumpla los requisitos del cliente antes de ser entregado.
- **La sexta fase es la de clausura:** donde se presenta el proyecto ante el cliente y su evaluación.

Primera etapa, requisitos

En la etapa de requisitos se definieron dos actores para interactuar con el sistema, los usuarios privilegiados o administradores y los usuarios comunes. Los usuarios privilegiados tienen el permiso de acceder a los datos que son almacenados en la base de datos, además de tener permiso a modificaciones en el contenido del sistema web. Los usuarios comunes tienen permiso de crear cuentas para tener acceso al sistema, actualizar su perfil de usuario, enlazar con una foto trampa, crear proyectos de detección y clasificación de especies. En esta etapa también se determinó los casos de uso, pues describen la interacción de los actores con el sistema, como se observa en la figura 2:

- **Administración:** se permite al actor de realizar operaciones como modificación y eliminación de datos sobre las tablas de la base de datos.



Fuente: Elaboración propia

Figura 2 Diagrama de caso de usos.

- Crear cuentas de acceso al sistema: las cuentas son necesarias para tener un control de quien utiliza el sistema. El actor llena un formulario para poder acceder al sistema.
- Acceso a los datos de la base de datos: el actor tiene los permisos necesarios para acceder a los datos de las tablas de la base de datos.
- Administración de los datos de su propia cuenta: el actor puede hacer modificaciones con los datos de su propia cuenta, como el cambio de nombre de usuario, actualización de contraseña.

- Creación de proyecto clasificador de especies: en este caso el actor puede crear un proyecto donde puede almacenar sus propias imágenes para la extracción de características y clasificación de las imágenes dependiendo a la especie que corresponda.
- Alta de foto-trampas en el sistema: el actor puede dar de alta una foto-trampa generado un token que se vinculará con la foto-trampa para la captura de imágenes.
- Extracción de características de imágenes: el actor por medio de la aplicación web, hará la extracción de regiones de interés para un posterior entrenamiento de la red neuronal.
- Entrenamiento de la red neuronal: con la ayuda del actuador y el uso de la extracción de características el sistema puede hacer un entrenamiento de clasificación de especies en su categoría correspondiente.
- Identificación de las especies: haciendo uso del entrenamiento de la red neuronal, el sistema procesa la información de la imagen, identifica y clasifica en su categoría correspondiente.
- Evaluación manual y automática: en este caso, el actor evalúa los datos arrojados por el sistema y manualmente los clasifica, ayudando al entrenamiento de la red neuronal; en su defecto, el sistema se encarga de hacerlo automático.

Los casos de uso permiten la creación de requerimientos no funcionales y funcionales, tablas 1 y 2, respectivamente.

Tabla 1 Plantilla para la descripción de requerimientos no funcionales.

RN-#	
Versión	
Autor	
Fuente	
Descripción	
Prioridad	
Comentarios	

Fuente: [J. R. Molina Ríos et al. 2018]

Tabla 2 Plantilla para la descripción de requerimientos funcionales.

RF - <id>	<Nombre descriptivo>	
Versión	<Número de versión actual> (<Fecha de versión actual>)	
Autor	<Autor de versión actual> (<Compañía del autor>)	
Fuente	<Fuente de versión actual> (<Organización fuente del requerimiento>)	
Descripción	El sistema debe comportarse como se describe en la siguiente secuencia de interacciones cuando <evento disparador>	
Alcance Y Nivel	<Que sistema es considerado caja negra en la perspectiva de diseño>	
Precondición	<Lo que se espera del entorno para la realización del requerimiento>	
Condición de Éxito	<Estado del entorno que indica que se realizó con éxito>	
Condición de Fracaso	<Estado del entorno cuando el requerimiento es abortado o falla>	
Actores	<Primario> {<Secundarios>}	
Evento Disparador	<Evento disparador o trigger>	
Secuencia Normal	Paso	Acción

	N	{el { <actor>. Sistema} <acción>
		{el { <actor>. Sistema} <acción> ejecutada por actor/ sistema}. Pasos descritos en <caso de uso (RF-x)> es
	n.1	
...
Postcondición	<Postcondición del caso de uso>	
Excepciones y Extensiones	Paso	Acción
	P	if <condición de excepción>, { el {
...
INFORMACIÓN RELACIONADA		
Prioridad	<grado de prioridad del requerimiento>	
Frecuencia	Este caso de uso se espera ser ejecutado <número de veces> número de veces/ <unidad de tiempo>	
Desempeño	Paso	Acción
	Q	m segundos
...
Canales hacia los Actores	<Archivos, interactivo, base de datos, etc	
Características Abiertas	<Lista de características que pueden afectar las decisiones sobre el caso de uso>	
Superiores	<Lista de requerimientos que incluyen este requerimiento>	
Subordinados	<Vínculos hacia subrequerimientos>	
Comentarios	<comentarios adicionales acerca del requerimiento>	

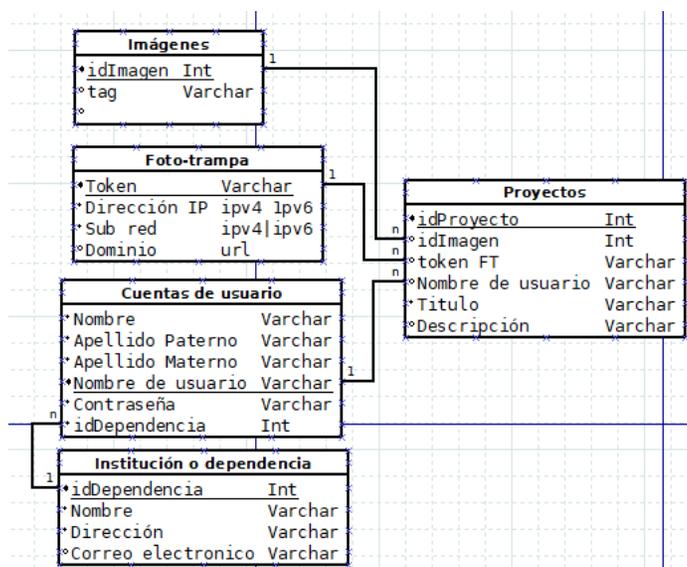
Fuente: [J. R. Molina Ríos et al. 2018]

Segunda etapa, planificación

Para contar con una planificación, se calcularon los recursos, el esfuerzo, además, se detalló el trabajo, definiendo los entregables del sistema y el alcance del proyecto.

Tercera etapa, diseño

En la tercera etapa se diseñaron prototipos del sistema, pues estos fueron cambiando conforme se desarrollaba. Primero, se diseñó la base de datos en su modelo conceptual o modelo de entidad relación, posteriormente se diseñó el esquema lógico de las tablas relacionales, en la figura 3 se puede observar el modelo lógico de la base de datos, como última actividad de esta etapa se comenzó con el diseño de la interfaz gráfica.



Fuente: Elaboración propia.

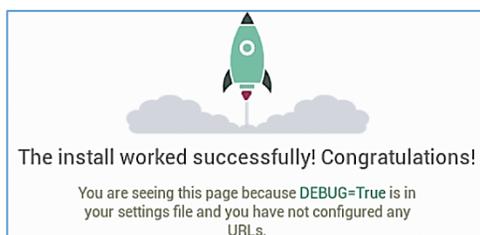
Figura 3 Esquema lógico de la base de datos.

Cuarta etapa, programación

En la cuarta etapa se codificaron las operaciones y funcionalidad con las que cuenta el sistema, esta cuarta etapa se dividió en diferentes fases.

La primera fase fue el maquetado de la página web, para hacer esto se utilizó el lenguaje de marcado de hiper texto en su versión 5 o también llamado HTML5 [Códigofacilito, 2015].

Para el desarrollo del sistema se optó por el uso de Django, que es un framework de alto nivel para desarrollo web rápido y que permite un diseño limpio, se inició con la tarea de crear un entorno virtual, con el fin de ejecutar comandos propios del framework, como crear un proyecto o poner el servidor en marcha en la figura 4 se puede apreciar el inicio del servidor del proyecto [Django, 2005].



Fuente: Elaboración propia.

Figura 4 Servidor de Django en marcha.

Ejecutando un comando en la terminal, con el entorno virtual activado, se creó el proyecto llamado “*algoritmo*”, dentro de este proyecto, se crea automáticamente un paquete llamado ‘*manage.py*’ que ayuda a la administración del sitio, dentro de este proyecto se crearon aplicaciones que desempeñan una actividad específica y en cada aplicación, se crearon modelos que guardan información en la base de datos; la primera aplicación creada dentro del proyecto ‘*algoritmo*’ fue la de ‘*core*’, siendo la base del proyecto, donde se encuentra el *template* que se utilizó en el desarrollo de las demás aplicaciones, la segunda aplicación es la de ‘*registration*’, que está encargada del inicio de sesión y registro de usuarios, la tercera aplicación es la de ‘*pages*’, esta aplicación fue creada para que el administrador pueda agregar páginas, por ejemplo, el aviso de privacidad, los términos y condiciones; la cuarta aplicación es la de ‘*social*’ que se encarga de gestionar las redes sociales de la empresa o administrador, por último se creó la aplicación de ‘*projects*’ que tiene como objetivo de permitir al usuario crear proyectos. Este apartado permite al usuario crear sus proyectos para la identificación y clasificación de faunas mediante el análisis de fotografías, alimentadas en el sistema por medio de una investigación conjunta que vincula foto-trampas.

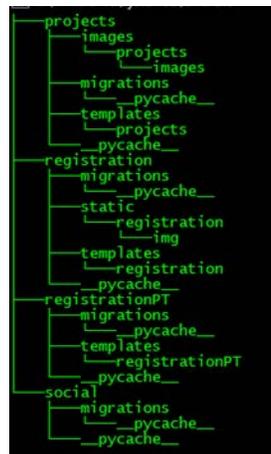
En las figuras 5 y 6 se muestra la estructura de directorios en la ruta de acceso donde se encuentra el proyecto y sus aplicaciones.

Con el uso del framework de Bootstrap, el diseño de la página se adapta a diferentes resoluciones de pantalla de dispositivos móviles, ofreciendo que nuestro sistema web se vea bien desde un teléfono, una tableta o una computadora, en otras palabras, es responsivo. [Mark Otto, Jacob Thornton, and Bootstrap contributors, 2011].



Fuente: Elaboración propia.

Figura 5 Estructura de directorios del proyecto.



Fuente: Elaboración propia.

Figura 6 Continuación de la estructura de directorios del proyecto.

3. Resultados

El uso del sistema de extracción de características y entrenamiento de la red neuronal es resultado de una investigación conjunta así como la alimentación de imágenes por medio de foto trampas, dicho sistema solo realiza su función mediante la ejecución de comandos de consola, la presente investigación tiene como resultado la generación de un sistema web que sirve como herramienta rápida de

clasificación de especies, limitándola a tres por el momento, el sistema es responsivo, adaptándose a diversas resoluciones de pantalla permitiendo verlo desde un teléfono, una tableta o una computadora, además permite el acceso a diversos usuarios. En este proyecto se crearon diferentes aplicaciones para el correcto funcionamiento del sistema, la aplicación de 'core' que es la base de la aplicación, encargada de mostrar la página principal, en la cual se introdujo un título, y una breve descripción del proyecto, además de un video de YouTube simulando el video de introducción al sistema (figura 7).

La aplicación de 'registration' es la encargada de tener el control de acceso, por ello se creó el inicio de sesión (figura 8) y registro de usuarios (figura 9), donde al ingresar sus datos, se enlaza a un perfil (figura 10).



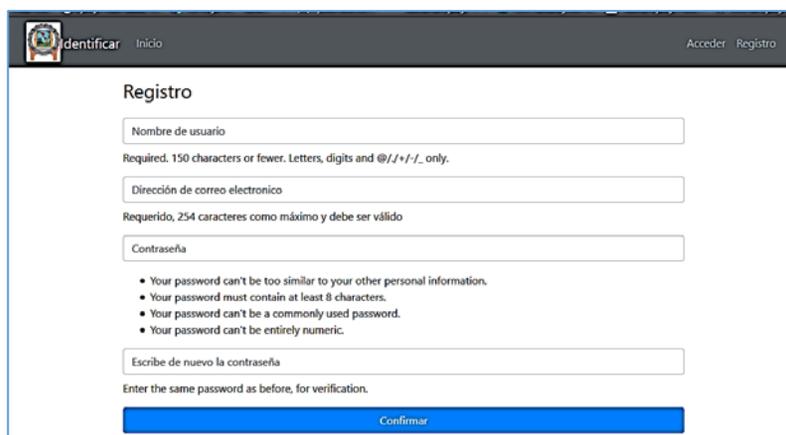
Fuente: Elaboración propia.

Figura 7 Página principal del sistema en fase beta.



Fuente: Elaboración propia.

Figura 8 Interfaz de inicio de sesión al sistema.

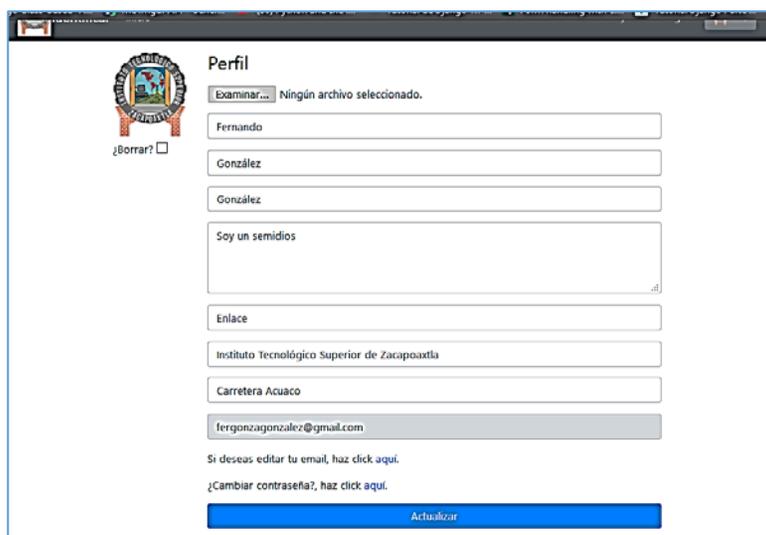


The screenshot shows a registration form titled 'Registro'. At the top left, there is a logo and the word 'Identificar' next to 'Inicio'. At the top right, there are links for 'Acceder' and 'Registro'. The form contains the following elements:

- A text input field for 'Nombre de usuario' with a note: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.'
- A text input field for 'Dirección de correo electrónico' with a note: 'Requerido, 254 caracteres como máximo y debe ser válido'.
- A text input field for 'Contraseña' with a list of requirements:
 - Your password can't be too similar to your other personal information.
 - Your password must contain at least 8 characters.
 - Your password can't be a commonly used password.
 - Your password can't be entirely numeric.
- A text input field for 'Escribe de nuevo la contraseña' with a note: 'Enter the same password as before, for verification.'
- A blue 'Confirmar' button at the bottom.

Fuente: Elaboración propia.

Figura 9 Interfaz de registro para poder acceder al sistema.



The screenshot shows a user profile page titled 'Perfil'. On the left, there is a circular profile picture placeholder with a '¿Borrar?' checkbox. The main content area includes:

- An 'Examinar...' button with the text 'Ningún archivo seleccionado.'
- Text input fields for 'Fernando', 'González', and 'González'.
- A text input field for 'Soy un semidios'.
- A text input field for 'Enlace' containing 'Instituto Tecnológico Superior de Zacapoaxtla'.
- A text input field for 'Carretera Acuaco'.
- A text input field for 'fergonzagonzalez@gmail.com'.
- Links: 'Si deseas editar tu email, haz click aquí.' and '¿Cambiar contraseña?, haz click aquí.'
- A blue 'Actualizar' button at the bottom.

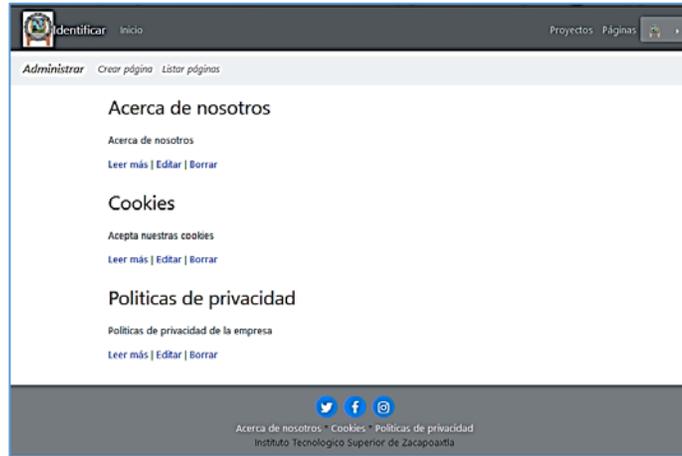
Fuente: Elaboración propia.

Figura 10 Interfaz de perfil de usuario.

La aplicación de 'pages' se creó para que el usuario administrador pueda crear, modificar o eliminar páginas informativas (figura 11). La aplicación 'social' se creó para que la empresa o administrador, enlace sus redes sociales (figura 12).

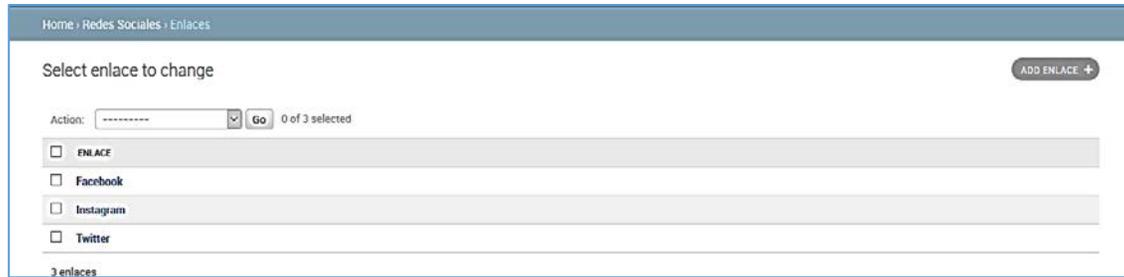
La aplicación de 'projects' (figura 13), se creó con la finalidad de que el usuario desarrolle proyectos pulsando el botón de '+ Proyecto' (figura 14), pues una vez creado (figura 15), dentro de este proyecto (figura 16), el usuario agrega imágenes (figura 17), etiquetándolas con una de tres diferentes categorías disponibles (armadillos, tlacuaches y coyotes), además de realizar un enlace a foto-trampas que

son configuradas en una investigación conjunta, debido a que sincroniza las fotografías tomadas por las fotos-trampas al sistema web.



Fuente: Elaboración propia.

Figura 11 Listado de páginas creadas por el usuario administrador.



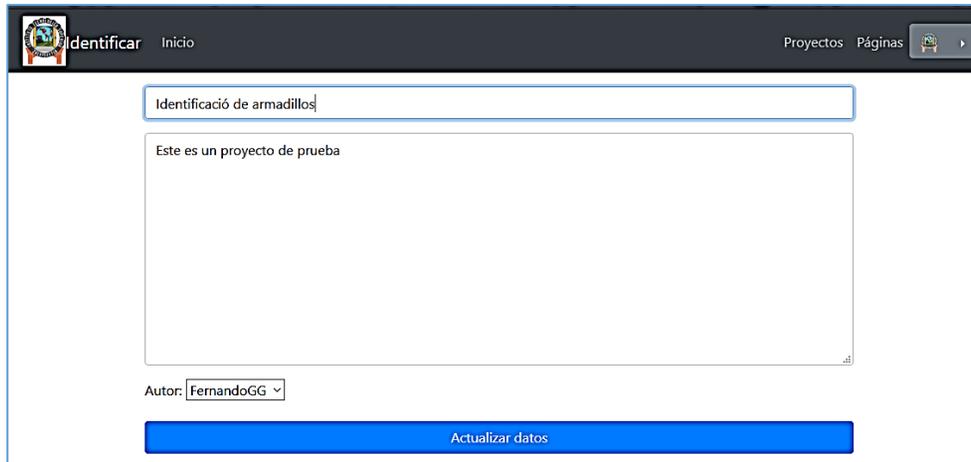
Fuente: Elaboración propia.

Figura 12 Interfaz de enlaces a redes sociales.



Fuente: Elaboración propia.

Figura 13 Interfaz de proyectos.



Fuente: Elaboración propia.

Figura 14 Interfaz de creación de proyectos.



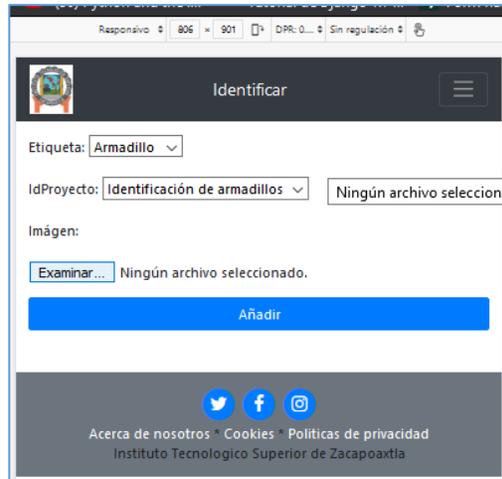
Fuente: Elaboración propia.

Figura 15 Lista de proyectos creados, en este caso, sólo es uno.



Fuente: Elaboración propia.

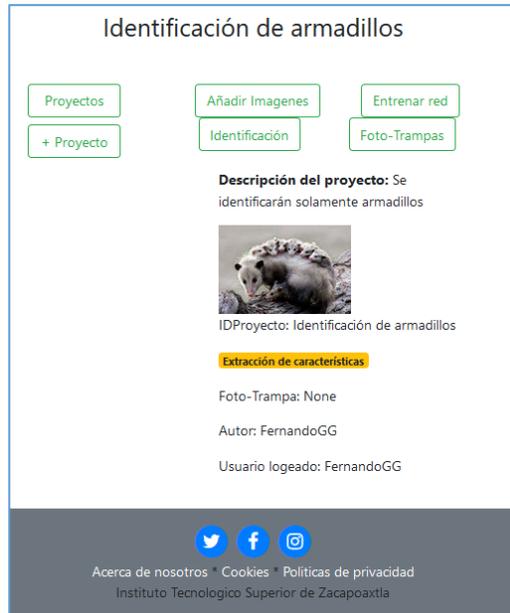
Figura 16 Detalles del proyecto en una resolución de teléfono móvil.



Fuente: Elaboración propia.

Figura 17 Interfaz donde se agregan las imágenes al proyecto.

Como se puede apreciar en la figura 18, el sistema es responsivo, pues se adapta a diferentes resoluciones de pantalla, sin embargo, también se puede notar que tiene un botón de color amarillo, que es el encargado de extraer las características de la imagen, en la figura 19 se muestra una etiqueta de extracción de características exitosa.



Fuente: Elaboración propia.

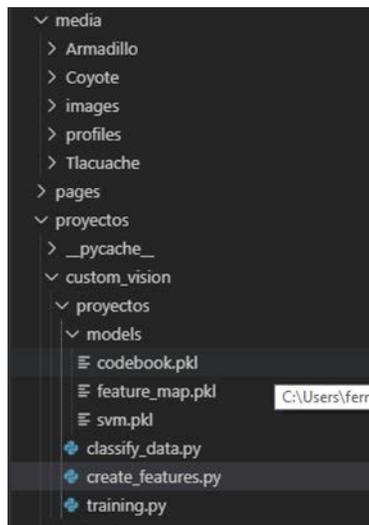
Figura 18 Imagen agregada al proyecto.



Fuente: Elaboración propia.

Figura 19 Extracción de características de la imagen exitosa.

En la figura 19 se manifiestan directorios donde se encuentran alojadas imágenes que son adheridas desde la interfaz del sistema, asimismo, se ve el algoritmo empleado para la extracción de características de la figura 20, que es ejecutado al presionar el botón amarillo que se observó en la figura 18, crea un archivo en la ruta de models/ llamado codebook.pkl que acumula información que es presentada en otros procesos.



Fuente: Elaboración propia.

Figura 20 Extracción de características de la imagen exitosa.

4. Discusión

La investigación se centra en el desarrollo de la aplicación web funcional para interactuar con el sistema clasificador, hasta el momento el sistema permite al acceso, la creación de proyectos, el agregado de imágenes, y la extracción de características, este último apartado interactúa con el sistema clasificador en la etapa de extracción de características, permitiendo crear un archivo que aloja los puntos de interés de cada una de las imágenes de las tres clases almacenadas en el sistema representativas a cada especie, aunque se cuenta con un sistema responsivo y funcional en los apartados ya desarrollados es indispensable integrar las otras etapas que permiten la clasificación de una imagen diferente a la almacenada en el banco de imágenes.

Las etapas que faltan integrar para que la aplicación cumpla al 100% la funcionalidad del clasificar es: la generación de banco visual que sirve para separar las clases en características de cada especie, así como el entrenamiento y la fase final del etiquetado o identificación de la imagen analizada. Además, el sistema se tiene que vincular con un dispositivo foto trampa que es el encargado de alimentar las imágenes que el sistema utiliza para extraer las características y posteriormente pasar al entrenamiento e identificación que servirá para clasificar las especies.

5. Bibliografía y Referencias

- [1] Códigofacilito. (2015, octubre). ¿Qué es HTML? Del sitio Web Códigofacilito: <https://www.codigofacilito.com/articulos/que-es-html>.
- [2] Comisión Nacional de Áreas Naturales Protegidas. (2018, 6 septiembre). México Megadiverso. México: CONABIO: <https://www.gob.mx/conanp/articulos/mexico-megadiverso-173682>.
- [3] Comisión Nacional para el Conocimiento y Uso de la Biodiversidad. (2019, 30 septiembre). ¿Qué es la biodiversidad? de CONABIO: https://www.biodiversidad.gob.mx/biodiversidad/que_es
- [4] IBM. (s.f.). El modelo de redes neuronales. de IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html.

- [5] De La Cruz, C. (2016). Metodología de la investigación tecnológica en ingeniería. *Ingenium*, Vol.1(1), pp.43-46.
- [6] Django. (2005). The Web framework for perfectionists with deadlines: <https://www.djangoproject.com/>.
- [7] FMCN, CONAFOR, USAID, & USFS. (2018). Manual para trazar la Unidad de Muestreo en bosques, selvas, zonas áridas y semiáridas. 31/oct/2019, de FONNOR: <https://fonnor.org/wp-content/uploads/2019/02/3.-Manual-de-Monitoreo-de-Fauna.pdf>.
- [8] Guevara A. (2016, 28 octubre). ¿Qué es Bootstrap? 31 de octubre de 2019, de devcode: <https://devcode.la/blog/que-es-bootstrap/>.
- [9] INFAIMON. (2018, 18 de enero). Visión por computador: qué es y cuáles son sus usos más comunes: <https://blog.infaimon.com/vision-computador-soluciones-permite/>.
- [10] López, R. E. (s.f.). Visión por computadora - Libro online de IAAR. 31 de octubre de 2019, de IAAR: <https://iaarbook.github.io/vision-por-computadora/>
- [11] Luna, R., Castañón, A., & Raz, A. (2010). La biodiversidad en México: su conservación y las colecciones biológicas. 31/octubre/2019, de Universidad Nacional Autónoma de México: <https://www.revistaciencias.unam.mx/es/103-revistas/revista-ciencias-101/843-la-biodiversidad-en-mexico-su-conservacion-y-las-colecciones-biologicas.html>.
- [12] Molina, J., Zea, M., Redrován, F., Loja, N., Valarezo, M., & Honores, J. (2018). "SNAIL", una metodología híbrida para el desarrollo de aplicaciones web: <https://www.3ciencias.com/wp-content/uploads/2018/05/Methodolog%C3%ADa-Hibrida-SNAIL.pdf>.
- [13] Otto, M., Thornton, J., & Bootstrap contributors. (2019, 19 de agosto). Bootstrap: <https://getbootstrap.com/>.
- [14] Secretaría de Medio Ambiente y Recursos Naturales. (2016, 21 de mayo). México país megadiverso. México: SEMARNAT: <https://www.gob.mx/semarnat/articulos/mexico-pais-megadiverso-31976>.
- [15] Sucar, L.E., & Gómez G. (s.f.). Visión Computacional. México: INAOEP: <https://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>.