



PRESTAR SERVICIOS PROFESIONALES PARA ATENDER TEMAS
TECNICOS
RELACIONADOS PARA
LA ELABORACIÓN DE DOCUMENTOS DE LINEAMIENTOS
TÉCNICOS, DE LOS PROGRAMAS, PROYECTOS, CONTRATOS Y/O
CONVENIOS A
CARGO DEL INVIA

Segundo informe de avance

INSTITUTO NACIONAL DE VÍAS TERRITORIAL ANTIOQUIA PROYECTO: MAPA DE VULNERABILIDAD FAUNÍSTICA

Avance del segundo mes de trabajo desarrollo del entorno de trabajo y desarrollo, capacitación del equipo de trabajo en el uso del entorno de trabajo, integración de las funciones de cálculo y aplicación del modelo de procesamiento, arquitectura del sistema del proyecto del Mapa de vulnerabilidad Faunística

Ing. Jairo I. Coy
Equipo de DevOps

PROLOGO

Este documento toma como base la GTC 185, la M80 Guía para la presentación de informes técnicos de avance de programas y proyectos de CTel y en la ISO 10013. El formato se realiza como una herramienta para la gestión de documentación y generación de reportes mensuales del equipo de trabajo de desarrollo del Mapa de vulnerabilidad faunística del segundo periodo de 2025.



Contenido

RESUMEN	4
1. INTRODUCCIÓN	5
2. OBJETIVOS	6
2.1. Objetivo general	6
2.2. Objetivos específicos.....	6
3. ESTRUCTURA DE TRABAJO	7
3.1. Definición del alcance y objetivos	7
3.2. Análisis de requisitos	7
3.3. Diseño de arquitectura o sistema	7
4. METODOLOGÍA.....	8
4.1. Modelos matemáticos.....	9
5. DESARROLLO Y RESULTADOS.....	10
5.1. Organización y estructura del trabajo	10
5.2. Estructura y diseño del entorno de trabajo sistema de procesamiento ...	11
5.3. Rediseño del flujo de datos	13
5.4. Tecnologías y componentes técnicos capacitación del equipo de trabajo	
	13
6. CONCLUSIONES.....	15
7. RECOMENDACIONES.....	16
BIBLIOGRAFÍA	17
ANEXOS.....	18

Listas de figuras

Figura 1. Esquema de la metodología de la fase II.	9
Figura 2. Flujo y componentes del proyecto automatización del mapa de vulnerabilidad faunística.....	11
Figura 3. Estructura del sistema de central de trabajo.	12
Figura 4. Rutas del sistema de procesamiento.....	12
Figura 5. Flujo de datos del sistema de automatización del mapa de vulnerabilidad faunística.	13
Figura 6. Interfaz de pruebas y visualización de capas.....	14

Listas de tablas

Tabla 1. Lenguajes integrados.	8
-------------------------------------	---

Listas de ecuación

No se encuentran elementos de tabla de ilustraciones.

Listas de anexos

Anexo 1. librerías instaladas en el entorno de procesamiento.	18
Anexo 2. Especificaciones técnicas del RACK de procesamiento.....	18

RESUMEN

Durante el segundo periodo de trabajo, comprendido entre el 1 al 31 de octubre, los avances logrados en el proyecto del Mapa de vulnerabilidad Faunística se centraron en el desarrollo del sistema de trabajo (sistema central de procesamiento), capacitación del equipo en el uso de las herramientas desarrolladas para el procesamiento de la información, primer avance en la integración del modelo en el sistema central de procesamiento e inicio del desarrollo de la interfaz gráfica del entorno.

Durante esta fase, se realiza un reestructuración y modificación al diseño como también al diseño del flujo de datos, de igual forma se robustece y se integran nuevos componentes a la arquitectura planteada en la primera fase, esto en pro de ofrecer un ecosistema más estable y eficiente.



1. INTRODUCCIÓN

El proyecto del Mapa de vulnerabilidad faunística requiere de la integración y aplicación de modelos complejos, procesamiento de matrices y vectores de gran tamaño. A sí mismo también requiere de una infraestructura sólida para el procesamiento de datos y la representación de los datos integrados en un motor de mapas.

En base a estas características y requisitos mencionados, el sistema debe estar en capacidad de gestionar datos, almacenar archivos, extraer información y procesarla con la mínima gestión humana. El diseño de la estructura para el procesamiento de datos y generación de resultados emplea el entorno de trabajo Django.

Django tiene como lenguaje base a Python, el cual favorece y facilita el desarrollo de funciones de procesamiento de grandes valores con mayor facilidad a diferencia de otros lenguajes de programación. Así mismo, brinda la oportunidad para generar ambientes para el procesamiento matricial y vectorial, como también la generación de entornos gráficos e interactivos que se pueden desplegar en ambientes web.



2. OBJETIVOS

2.1. Objetivo general

Prestar servicios profesionales especializados para abordar aspectos técnicos relacionados con la elaboración y desarrollo del sistema digital, generación de documentación y capacitación del equipo de trabajo para el procesamiento y aplicación de modelos de análisis de niveles de riesgo en las carreteras nacionales de Colombia.

2.2. Objetivos específicos

- Desarrollar y configurar el entorno de trabajo para el procesamiento de modelos para la estimación de los niveles de riesgo de la fauna silvestre.
- Integrar los modelos de análisis y estudio en el entorno central de trabajo.
- Capacitar al equipo de trabajo en el uso del sistema central de procesamiento.
- Capacitar al equipo de trabajo en el uso del sistema de generación de documentos técnicos.
- Capacitar al equipo de trabajo en el sistema de control de versiones en el sistema de gestión del proyecto.

INVIAS
INSTITUTO NACIONAL DE VIAS

3. ESTRUCTURA DE TRABAJO

Para el desarrollo del entorno de trabajo se llevan a cabo una serie de etapas y procesos que permitan realizar el despliegue en servicio local, al igual que se seleccionan una serie de herramientas que permitan su instalación y configuración. Las librerías instaladas, en el entorno de trabajo, son solicitadas por el equipo de análisis, otras librerías, son necesarias para que la arquitectura del sistema central funcione de forma apropiada.

3.1. Definición del alcance y objetivos

En esta fase de trabajo, se realiza la instalación e integración de herramientas y componentes que permiten el funcionamiento del entorno de procesamiento (Django), como las librerías y elementos que permitan su buen funcionamiento, al igual que se realizan una serie de modificaciones en el diseño de la arquitectura inicialmente propuesta en la fase I, no obstante se identifica una serie de limitaciones que se buscan reducir y suprimir favoreciendo en tiempos de procesamiento, capacidad de almacenamiento y estabilidad del sistema.

3.2. Análisis de requisitos

El estudio de los componentes y elementos técnicos parten de la selección de los requisitos técnicos, como de la selección de los elementos que permitan su buen funcionamiento y generar los resultados deseados. Los requisitos técnicos, se identifican basándose en el tipo de resultados deseados como de la clase de información que se desea procesar. Esta selección se basa en tres principios que brinden un sistema sólido, estable, confiable, escalable y seguro. Esto involucra tanto la infraestructura del sistema de procesamiento, como de los elementos requeridos para su despliegue en un ambiente de producción (servidor comercial del INVIAS).

3.3. Diseño de arquitectura o sistema

Su arquitectura se fundamenta en el lenguaje Python como motor de procesamiento y lenguaje base de la infraestructura de procesamiento (entorno de trabajo). Para la representación gráfica se implementan tres lenguajes de desarrollo web, de uso común, estos lenguajes son descritos en la Tabla 1, al igual que el lenguaje de gestión de datos. En conjunto, todos los lenguajes permiten crear entornos interactivos, dinámicos, de fácil acceso y con una experiencia de usuario atractivo en los servidores web sin necesidad de que el desarrollo sea instalado en múltiples dispositivos.

Tabla 1. Lenguajes integrados.

Lenguaje	Logo	Descripción
Python		Lenguaje de programación base. Usado para estructurar las funcionalidades y servicios dentro del proyecto.
HTML5		Lenguaje de marcado y maquetado de páginas web.
CSS3		Lenguaje de estilos y animación de páginas web.
JAVASCRIPT		Lenguaje de estructuración, procesamiento y funcionalidades en páginas web.
JSON		Lenguaje de gestión de datos, servicios y microservicios en la web.

4. METODOLOGÍA

INSTITUTO NACIONAL DE VIAS

La metodología de trabajo empleada en esta fase se compone de cuatro etapas, identificación y selección de componentes, pruebas y testing, integración de componentes de funcionamiento y cierra con la capacitación del equipo de trabajo. Durante cada etapa se busca que el sistema de procesamiento se fortalezca y se logre una infraestructura sólida al igual que funcione de forma adecuada.

La primera parte se enfoca en la identificación y selección de los componentes del entorno de procesamiento. Parte de los requisitos son solicitados por el equipo de análisis y los otros son los que permiten que el sistema funcione de forma adecuada.

La segunda parte de la metodología se enfoca en realizar las pruebas técnicas de funcionamiento del entorno de trabajo. Todo el despliegue se realiza en servicio local.

La tercera parte de la metodología se enfoca en la integración de los modelos de procesamiento. Durante esta etapa, se realiza la instalación de los requisitos necesarios para que los modelos se ejecuten de forma apropiada. Durante esta etapa solo se verifica que los modelos funcionen, más no en la calidad y precisión de los resultados.

En la cuarta etapa, se capacita al equipo de trabajo en tres actividades, generación de documentos técnicos en el sistema de gestión, uso del sistema de procesamiento y trabajo en el sistema de gestión de versiones.

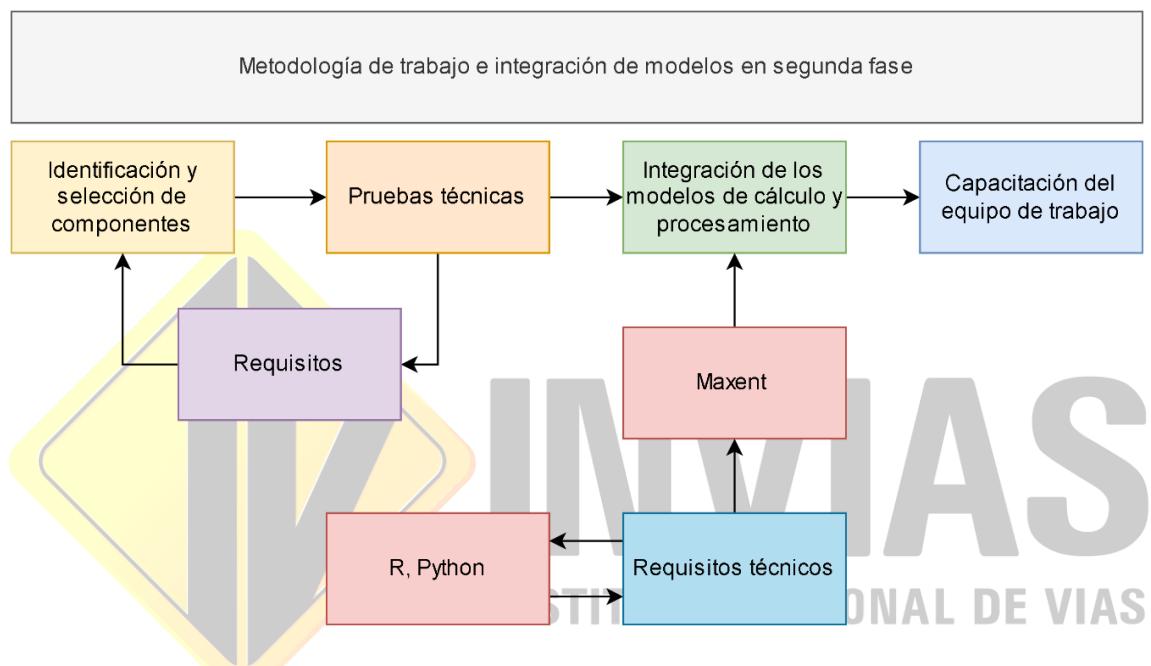


Figura 1. Esquema de la metodología de la fase II.

4.1. Modelos matemáticos

Los modelos matemáticos empleados durante esta fase son descritos en el informe del equipo de análisis. Se recomienda revisar el informe técnico entregado por el equipo de análisis.

5. DESARROLLO Y RESULTADOS

5.1. Organización y estructura del trabajo

El proyecto se compone de tres sistemas interconectados. Cada uno de estos sistemas cumple un papel específico y no redundante. Como mecanismo de comunicación y puente de transferencia de información se desarrolla unas API's, adicional que cada sistema contiene un elemento de almacenamiento como soporte de información, esto con el objetivo de mantener un ecosistema que evite la pérdida de información, en caso de fallos o desconexión y/o baja del sistema.

El primer sistema corresponde al ecosistema de Google Earth Engine (GEE), en este sistema se realiza el llamado de información de forma automática, como también se realiza la adecuación de la información para ser aceptada por el modelo de procesamiento. También contiene los módulos de almacenamiento donde el sistema de procesamiento se conectará y llamará la información para ser procesada adquirida por fuentes paralelas.

El segundo contiene el sistema central de procesamiento. En él, se encuentra integrado y configurado el Modelo de máxima entropía (MAXENT). Este se compone de una serie de componentes y elementos para su adecuado y óptimo funcionamiento. Este sistema también se usará como elemento de almacenamiento de soporte (backup) como también para almacenar información no procesada. El segundo sistema, se encarga de generar los enlaces de acceso o endpoints, es a través de ellos se va a adquirir la información ya procesada que será consumida por el tercer sistema.

El tercer sistema corresponde al sistema central de visualización del INVIAS, el sistema se conecta por medio de una API para alimentar el sistema de visualización y/o mapas del proyecto ERMES.

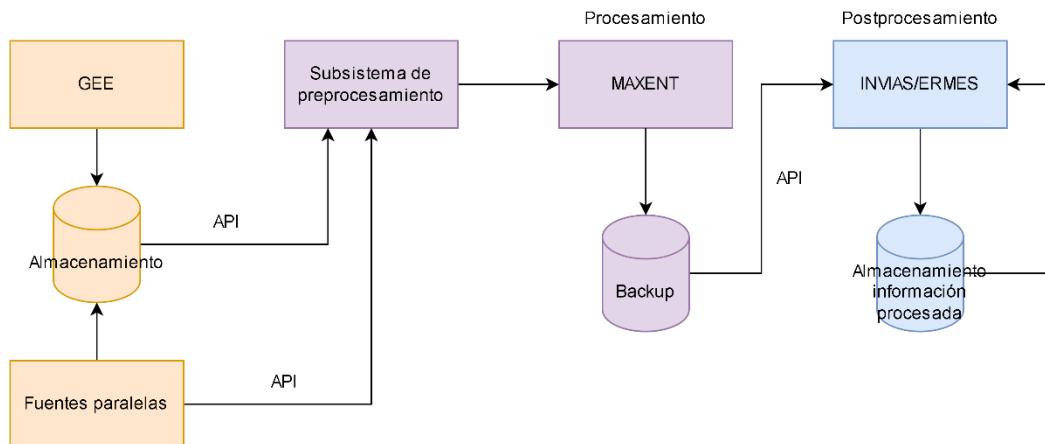


Figura 2. Flujo y componentes del proyecto automatización del mapa de vulnerabilidad faunística.

5.2. Estructura y diseño del entorno de trabajo sistema de procesamiento

El sistema de procesamiento contiene tres aplicativos backend, frontend y demos y como subsistema central esta inviasvivo. Este último, es la columna del sistema de procesamiento. Como se indicó el entorno de trabajo (Framework) es Django, el cual permite desarrollar en forma modular. Cada módulo corresponde a una aplicativo con unas funcionalidades independientes.

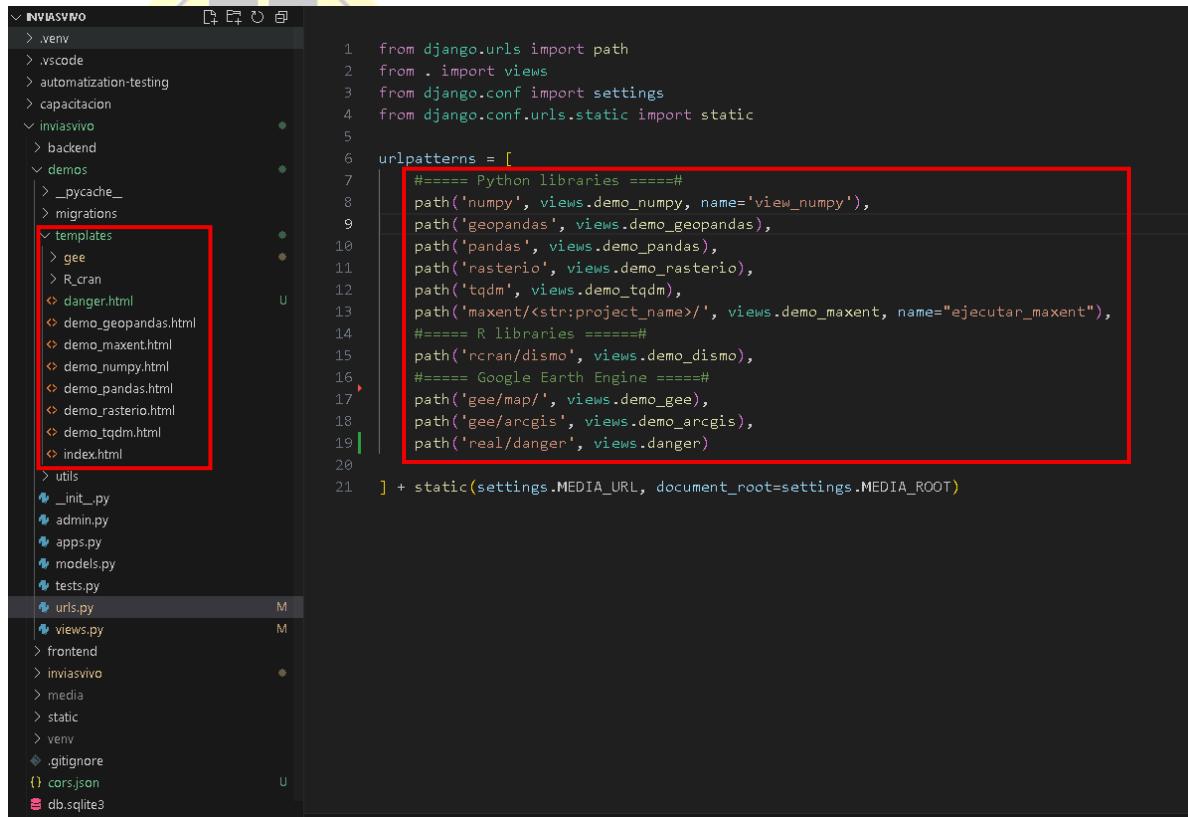
El aplicativo backend se encarga de procesar la información, en él se encuentra integrado el modelo MAXENT. Así mismo, general los endpoints donde se almacenará al información procesada y lista para ser consumida por el sistema de visualización. Este modulo contiene las librerías y los lenguajes de procesamiento integrado (R+).

Para esta etapa, el sistema contiene, dos módulos adicionales, demos y frontend. Estos módulos, se toman como aplicativos de testing. El modulo demos contiene todas las librerías solicitadas por el equipo de análisis. El modulo explica como se debe usar cada librería dentro del ecosistema de procesamiento. En cuanto al frontend permite contiene un sistema provisorio de visualización de resultados.

Directorio: C:\Program Files\Ampps\www\inviasvivo\inviasvivo

Mode	LastWriteTime	Length	Name
d----	9/9/2025 11:31 PM		backend
d----	10/17/2025 7:56 AM		demos
d----	9/9/2025 11:24 PM		frontend
d----	9/5/2025 9:58 AM		inviasvivo
d----	10/22/2025 4:14 PM		media
d----	10/17/2025 8:21 AM		static
d----	10/11/2025 3:36 PM		venv
-a----	10/19/2025 11:49 AM	3627	.gitignore
-a----	10/17/2025 2:10 AM	131072	db.sqlite3
-a----	9/4/2025 10:03 AM	688	manage.py
-a----	10/22/2025 4:08 PM	1078	README.md
-a----	10/13/2025 1:40 PM	986	requirements.txt

Figura 3. Estructura del sistema de central de trabajo.



```

INVIASVIVO
├── .venv
├── .vscode
├── automation-testing
├── capacitacion
└── inviasvivo
    ├── backend
    └── demos
        ├── __pycache__
        ├── migrations
        └── templates
            ├── gee
            ├── R_cran
            ├── danger.html
            ├── demo_geopandas.html
            ├── demo_maxent.html
            ├── demo_numpy.html
            ├── demo_pandas.html
            ├── demo_rasterio.html
            ├── demo_tqdm.html
            └── index.html
    ├── utils
    ├── __init__.py
    ├── admin.py
    ├── apps.py
    ├── models.py
    ├── tests.py
    ├── urls.py
    └── views.py
    ├── frontend
    └── inviasvivo
        ├── media
        ├── static
        └── venv
            ├── .gitignore
            ├── cors.json
            └── db.sqlite3

```

1 from django.urls import path
2 from . import views
3 from django.conf import settings
4 from django.conf.urls.static import static
5
6 urlpatterns = [
7 ##### Python libraries #####
8 path('numpy', views.demo_numpy, name='view_numpy'),
9 path('geopandas', views.demo_geopandas),
10 path('pandas', views.demo_pandas),
11 path('rasterio', views.demo_rasterio),
12 path('tqdm', views.demo_tqdm),
13 path('maxent<str:project_name>', views.demo_maxent, name="ejecutar_maxent"),
14 ##### R libraries #####
15 path('rcran/dismo', views.demo_dismo),
16 ##### Google Earth Engine #####
17 path('gee/map/', views.demo_gee),
18 path('gee/arcgis', views.demo_arcgis),
19 path('real/danger', views.danger)
20
21] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

Figura 4. Rutas del sistema de procesamiento.

5.3. Rediseño del fujo de datos

Con el objetivo de mejorar los tiempos de proceso y evitar la caída del sistema por saturación y/o problemas en la limitación de memoria, se realiza un rediseño en la arquitectura en general. Como estrategia, se integra un RACK que se usará como sistema de procesamiento y almacenamiento. El sistema del INVIAS/ERMES actuará como consumidor y su procesamiento se enfocará en la generación de mapas y representación de las capas procesadas. En cuanto al sistema GEE, brindará la información que alimente el modelo de procesamiento.

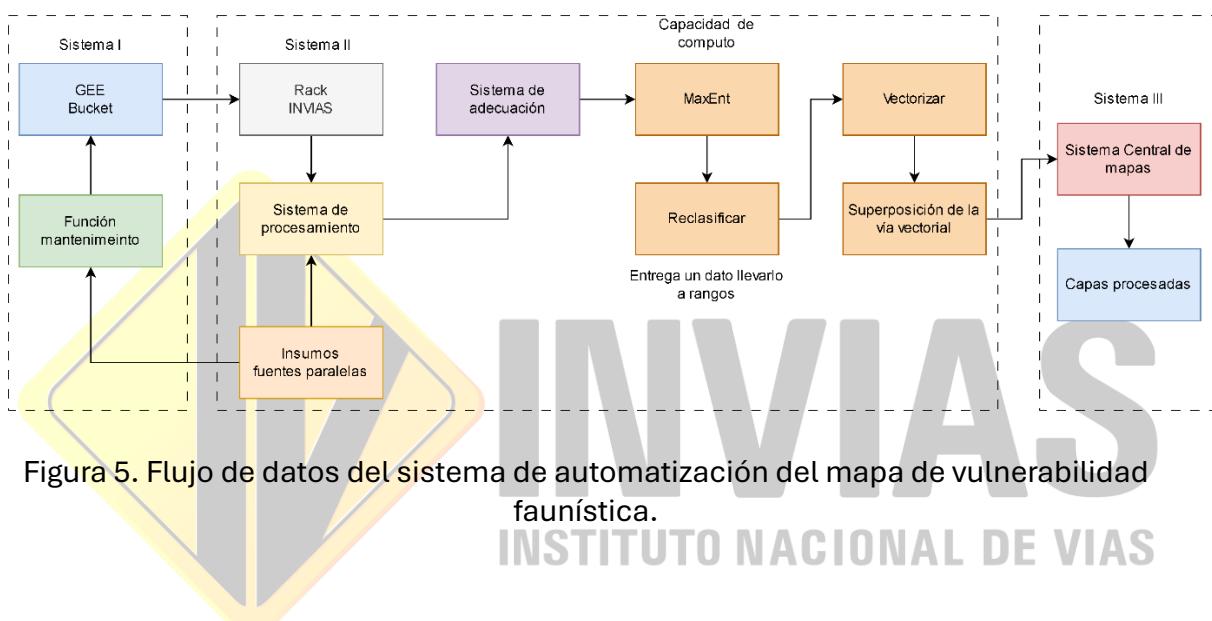


Figura 5. Flujo de datos del sistema de automatización del mapa de vulnerabilidad faunística.

INVIAS
INSTITUTO NACIONAL DE VIAS

5.4. Tecnologías y componentes técnicos capacitación del equipo de trabajo

Como proceso de soporte se capacita al equipo de trabajo en el uso de la tecnología desarrollada, se muestran los componentes, funciones y localización. De igual forma se capacita al equipo de trabajo en el proceso de actualización de repositorios. Esto con el objetivo de mantener una trazabilidad clara de la evolución del código durante el desarrollo del proyecto.

Para el uso y gestión de repositorios se genera una video capacitación adjunta en el siguiente enlace.

[capacitacion_gestion_repositorios.](#)

Como proceso de capacitación en el uso del entorno de procesamiento desarrollado, se realiza una herramienta de visualización y se realiza un despliegue en local donde se muestra cómo se debe operar con las funciones. La interfaz desarrollada durante el proceso de capacitación se usará como herramienta de testeo y visualización de resultados. Esta aplicación permite mostrar y ocultar las capas procesadas e integradas en el mapa base y visualizar los resultados de forma gráfica.

Este sistema, se conecta directamente con el sistema de almacenamiento en la nube usado como fuente primaria (Google Cloud Engine) y llama la capa para ser integrada en el mapa del sistema de procesamiento. Para esta etapa, el sistema no pasa por el modelo de procesamiento.

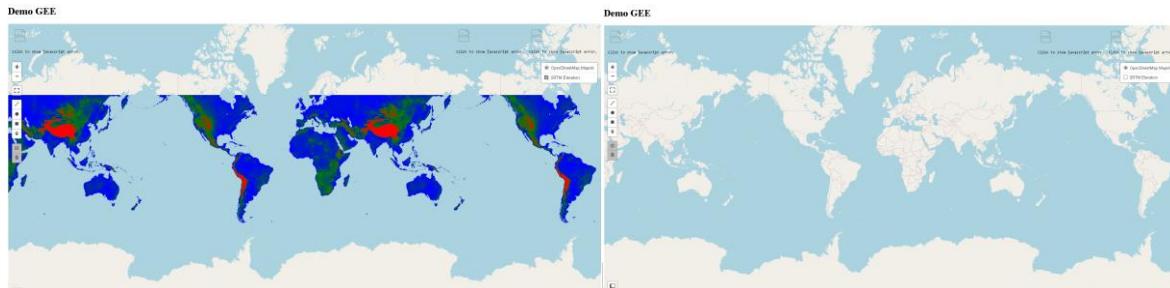


Figura 6. Interfaz de pruebas y visualización de capas.

Debido a las limitaciones y falta de compatibilidad que presenta R con Python en Django, se realiza una compresión de R+ y se trabaja con una versión forjada de miniconda. Este cambio, permite correr con éxito el modelo de procesamiento, y queda en un estado, suficiente para activarlo generar los resultados esperados. Las librerías que se requieren para poder correr el modelo y generar las capas procesadas se listan en el siguiente archivo. Miniconda actúa como entorno virtual de desarrollo. Este actúa como un espacio aislado para desarrollo dentro de Django sin afectar la instalación global de Python, favorable para desarrollar proyectos específicos.

[librerias sistema procesamiento.](#)

6. CONCLUSIONES

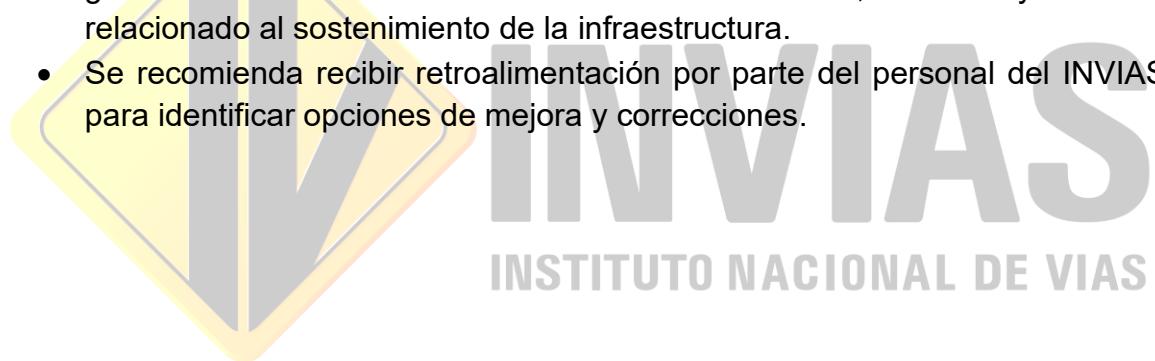
- La configuración del entorno de trabajo para el procesamiento de modelos permitió establecer una base tecnológica sólida para la estimación de riesgos en fauna silvestre facilitando el análisis eficiente y reproducible de datos ecológicos.
- La incorporación de modelo de estudio en el entorno central optimizó los flujos de trabajo, promoviendo la coherencia metodológica y mejorando la capacidad de respuesta ante escenarios de riesgo faunístico.
- La capacitación en el sistema central de procesamiento permitió al equipo técnico tener control sobre el manejo competente de las herramientas analíticas y fomentando la autonomía operativa.
- La formación en el sistema de generación de documentos técnicos, contribuyó a estandarizar la producción de informes, garantizando claridad, precisión y trazabilidad en los resultados obtenidos en la etapa.
- La capacitación en el sistema de control de versiones permitió una administración más ordenada del desarrollo del proyecto, facilitando la colaboración, el seguimiento de cambios y la preservación del historial de trabajo.



INVIAS
INSTITUTO NACIONAL DE VIAS

7. RECOMENDACIONES

- Se recomienda documentar de forma detallada la configuración del entorno de trabajo y capacitar al personal del invias en la integración, instalación y despliegue del entorno de procesamiento, para facilitar futuras actualizaciones o replicaciones.
- Se recomienda integrar herramientas de automatización como scripts de instalación o herramientas CI/CD para reducir los errores durante el proceso de despliegue.
- Se recomienda realizar pruebas de validación cruzada entre los sistemas y modelos para asegurar la compatibilidad y precisión en los resultados.
- Se recomienda establecer un sistema de control de calidad para verificar que los modelos integrados generen resultados consistentes.
- Se recomienda generar unas sesiones de capacitación y práctica en el uso de la infraestructura desarrollada hacia el personal del INVIAIS.
- Se recomienda definir plantillas estandarizadas por el INVIAIS para la generación de los documentos de mantenimiento, usuario y todo lo relacionado al sostenimiento de la infraestructura.
- Se recomienda recibir retroalimentación por parte del personal del INVIAIS para identificar opciones de mejora y correcciones.



BIBLIOGRAFÍA

- S. Chacon and B. Straub, *Pro Git*, 2nd ed. Apress, 2014. [Online]. Available: <https://git-scm.com/book/en/v2>
- S. Chacon y B. Straub, *Pro Git*, 2.^a ed. Apress, 2014. [En línea]. Disponible en: <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git>
- J. Álvarez, "Aprende Git y GitHub – Curso desde cero," FreeCodeCamp, 2022. [En línea]. Disponible en: <https://www.freecodecamp.org/espanol/news/aprende-git-y-github-curso-desde-cero/>
- Django Software Foundation, "Django Documentation, Release 5.2," 2025. [En línea]. Disponible en: <https://docs.djangoproject.com/en/5.2/>
- Google Cloud, "Documentación oficial de Google Cloud," 2025. [En línea]. Disponible en: <https://docs.cloud.google.com/docs?hl=es-419>



ANEXOS

```
● (venv) PS C:\Program Files\Ampps\www\inviavivo\inviavivo> pip list
● Package           Version
-----
affine            2.4.0
attrs             25.4.0
certifi           2025.10.5
cffi              2.0.0
click              8.3.0
click-plugins     1.1.1.2
cligj              0.7.2
colorama          0.4.6
geopandas         1.1.1
Jinja2             3.1.6
MarkupSafe        3.0.3
numpy              2.3.3
packaging          25.0
pandas             2.3.3
pip                25.2
pycparser          2.23
pyogrio            0.11.1
pyparsing          3.2.5
pyproj              3.7.2
python-dateutil   2.9.0.post0
pytz               2025.2
rasterio           1.4.3
rpy2               3.6.4
rpy2-riinterface   3.6.3
rpy2-robjects       3.6.3
shapely             2.1.2
six                 1.17.0
tqdm               4.67.1
tzdata              2025.2
tzlocal             5.3.1
○ (venv) PS C:\Program Files\Ampps\www\inviavivo\inviavivo>
```

Anexo 1. librerías instaladas en el entorno de procesamiento.

Condiciones mínimas del sistema físico	
Característica	Capacidad
Tipo	VPS dedicada
CPU	Core i7
RAM	16GB
Almacenamiento	Expansible
GPU	Dedicada N-series

Anexo 2. Especificaciones técnicas del RACK de procesamiento.