

Acta de Reunión

Fecha: Miércoles, 22 de octubre de 2025

Lugar: Sesión virtual (Teams)

Proyecto: Mapa de Vulnerabilidad Faunística – INVÍAS

Ecosistema y herramientas: Google Earth Engine (bucket y funciones de mantenimiento), rack de procesamiento on-premise, sistema central de consumo, integración Python–R (MaxEnt), repositorio con wiki y Projects, formatos JSON/GeoJSON, TIFF y SHP

Duración aproximada: 1 h 07 min

1. Propósito de la reunión

Alinear el inicio de la fase de integración para producir resultados, validar el flujo de datos Google Earth Engine → rack (MaxEnt) → sistema central, acordar la logística de insumos (carpeta Jackknife en el repositorio), y definir criterios técnicos para preprocesamiento, teselado y buffers a fin de optimizar desempeño.

2. Asistentes

- Luis Esteban Gómez Cadavid
- Andrés Felipe Hernández Marulanda
- Jairo Iván Coy Coy
- Nelson Aníbal Miranda Ríos
- Jasmín Marín Perez

3. Desarrollo de la reunión (narrativa)

Se abrió la sesión indicando que esta reunión marca el paso de diseño a resultados, dando inicio a la fase de integración. Se reportó la integración completa del flujo Python–R para el algoritmo de MaxEnt en el entorno de trabajo. El sistema ejecuta la función de workflow (preparación y ejecución), pero no genera salidas por falta de insumos; se solicitó subir la carpeta Jackknife al repositorio (zip o carpetas). Se reiteró que el repositorio es el eje de trazabilidad: allí deben quedar actas, informes, algoritmos y adjuntos.

Desde TI se planteó y socializó el flujo operativo: Google Earth Engine alimenta un bucket con funciones de mantenimiento (limpieza/rotación), el rack on-premise ejecuta el procesamiento (MaxEnt) y el sistema central consume y publica salidas. El objetivo es automatizar abastecimiento desde GEE y evitar sobrecargar el front. Se propuso que los insumos vectoriales (vías, puntos de atropellamiento) se eleven también a GEE y allí se rastericen/recorten, dejando al rack el alineamiento definitivo y la ejecución del modelo.

Nelson explicó las transformaciones de salida: el raster de probabilidad de MaxEnt se reclasifica en rangos entre 0 y 1, se vectoriza y se transfiere al eje vial para asignar a cada segmento su probabilidad. Esta salida segmentada es la que verá el cliente en su ecosistema (ArcGIS/Hermes). Se aclaró que el mapa temporal del equipo es solo demostrador mientras se integran las capas al entorno del cliente.

Se discutió la gestión de teselas y buffers para descargas desde GEE. Aumentar el buffer de 100 m a 200 m multiplica el área y el costo computacional de manera no lineal; para evitar cortes por discretización con celdas de 100 m se propuso usar un buffer operativo de $100\sqrt{2}$ m. Se dejó como criterio ajustar simultáneamente el tamaño de píxel y el buffer, documentando la regla en la wiki. Asimismo, se enfatizó el alcance contractual: trabajar solo con la red vial de INVÍAS (no toda la red IGAC), lo cual reduce significativamente volumen y tiempos.

Se ordenó el pipeline de datos en tres etapas: preprocesamiento (alineación, distancias euclidianas, estandarización a grilla), procesamiento (corridas de MaxEnt) y posprocesamiento (reclasificación, vectorización, segmentación de la vía y preparación para publicación). Se acordó que las adecuaciones

necesarias para que MaxEnt reciba capas ya alineadas deben quedar encapsuladas antes de la corrida del modelo. Finalmente, se definió que esta semana el foco para modelación será el preprocesamiento, TI continuará con despliegue e integración, y se programará una conversación con INVÍAS para precisar la interfaz esperada (mapa, dashboard u otra).

4. Agenda (simplificada)

- Inicio de fase de integración y trazabilidad en el repositorio (actas, informes, algoritmos, adjuntos).
- Flujo acordado: GEE → bucket (mantenimiento) → rack (MaxEnt) → sistema central (consumo).
- Logística de insumos: carpeta Jackknife al repositorio; formatos de intercambio (GeoJSON/CSV para puntos, SHP/GeoJSON para vías).
- Criterios técnicos: rasterización/recorte en GEE; alineación en rack; salida segmentada por vía para el cliente.
- Optimización: teselado, tamaño de píxel y buffer operativo; uso de $100\cdot\sqrt{2}$ m con celdas de 100 m.
- Alcance: limitar a la red INVÍAS; preparación de reunión para definir UI/UX con el cliente.

5. Revisión de compromisos pasados

- Integración Python–R del workflow de MaxEnt operativa; pendiente de insumos para producir salidas.
- Capacitación en uso de repositorios realizada; estructura de proyectos y wiki disponible para carga de evidencia.
- Arquitectura de tres sistemas socializada; se mantiene el plan de bucket en GEE, rack de procesamiento y front de consumo.

6. Próximos pasos y posibles fechas

- Subir al repositorio la carpeta Jackknife (rasterIn, bandas, distancias, puntos de atropellamiento y vías) – inmediato.
- Elevar vías y puntos a GEE y definir qué capas se rasterizan/recortan en GEE vs. en rack – 23/10/2025.
- Implementar en preprocesamiento la alineación de capas y cálculo de distancias euclidianas antes del modelo – 24/10/2025.
- Documentar en la wiki la regla de buffer con celdas de 100 m ($100\cdot\sqrt{2}$ m) y el criterio de teselado – 24/10/2025.
- Correr prueba completa de MaxEnt con insumos del Jackknife y publicar raster reclasificado de ejemplo – 25/10/2025.
- Vectorizar y segmentar la vía con rangos de probabilidad; preparar capa para consumo en el ecosistema del cliente – 26/10/2025.
- Agendar reunión con INVÍAS para definir UI/UX de la salida (mapa vs. tablero) y requerimientos de publicación – semana del 27/10/2025.

7. Hitos / Conclusiones

- Integración del workflow Python–R de MaxEnt en el entorno de trabajo.
- Flujo de abastecimiento y procesamiento acordado (GEE → rack → consumo).
- Criterio operativo para buffers y teselado definido para celdas de 100 m.
- Alineamiento con el alcance contractual: trabajar sobre la red vial de INVÍAS.

8. Análisis y recomendaciones

Formalizar el preprocesamiento antes del modelo reduce iteraciones fallidas y acelera la puesta en producción. Elevar a GEE la mayor parte de rasterización/recortes y reservar al rack la alineación final y la

corrida del modelo balancea costo y desempeño. La regla de $100\sqrt{2}$ m con celdas de 100 m evita cortes por discretización sin disparar el área procesada. Se recomienda mantener checklist por corrida (teselas, buffer, grilla, SR, NoData), registrar en la wiki el mapa de dependencias entre frentes y, con el cliente, cerrar pronto la definición de interfaz para no retrasar el posprocesamiento.