

ACTA 4 – REUNIÓN DEL 25 DE NOVIEMBRE DE 2025

Reunión virtual (Google Meet)

1. Propósito de la reunión

Verificar la conexión entre el bucket de Google Cloud Storage y el sistema de procesamiento en el RAC, mostrar el flujo automatizado de descarga y ejecución del modelo MaxEnt, revisar la arquitectura de preprocesamiento–procesamiento–postprocesamiento con programación paralela, y diligenciar en detalle los formatos solicitados por la OTIC/área de tecnología de INVIAS para la provisión de servidores (uno de base de datos y otro de aplicación/procesamiento).

2. Asistentes

- Jairo Iván Coy Coy
- Andres Felipe Hernández Marulanda
- Luis Esteban Gómez Cadavid

(Se hace referencia a Nelson y Jasmín en el contexto de uso futuro del RAC y los accesos, aunque no intervienen en la mayor parte de la grabación.)

3. Desarrollo de la reunión (relato)

La sesión inicia con el propósito práctico de revisar los permisos y el acceso a Google Cloud Storage. Jairo guía a Luis para ubicar y configurar las cuentas de servicio (Service Accounts) y generar el archivo JSON de credenciales que permitirá a los scripts conectarse al bucket. Se verifica que el archivo JSON contiene la clave privada y se acuerda enviarlo comprimido por seguridad.

Posteriormente, Jairo muestra el sistema de descarga automatizada ya desarrollado: a través de un supervisor y scripts en el RAC, el sistema recorre periódicamente el bucket, identifica los archivos relevantes (JSON, shapefiles, TIFF, etc.), los descarga, organiza en una estructura de carpetas y activa el entorno de trabajo para correr MaxEnt en R. Se evidencia cómo la programación paralela (uso de múltiples hilos/procesos) reduce considerablemente los tiempos de ejecución respecto a una versión secuencial.

Se explica que el flujo actual es: Google Earth Engine produce y envía insumos a Google Cloud Storage; desde allí, el sistema en el RAC descarga y ejecuta el preprocesamiento, procesamiento (MaxEnt) y postprocesamiento, generando resultados en carpetas específicas. Se resalta la importancia de tener un bucket de trabajo y un bucket de backup, de forma que el sistema consuma desde el backup estable mientras se actualiza la fuente, mitigando riesgos ante cambios o errores.

A continuación, se discute cómo se integran las funciones de Luis (índices espectrales e insumos satelitales) y las de Nelson (alineamiento, segmentación de vías y postprocesamiento) dentro del mismo flujo, incluyendo el uso de programación paralela por regiones y teselas de Colombia, aprovechando los núcleos múltiples solicitados para el RAC.

En la segunda parte de la reunión, Jairo y Andres se concentran en diligenciar los formatos oficiales de solicitud de servidores para INVIAS. Se separan claramente dos componentes: un servidor de base de datos (PostgreSQL/PostGIS) y un servidor de aplicación/procesamiento (Linux Oracle) donde residirá el entorno del modelo. Se definen CPU, memoria, almacenamiento, sistema operativo, criticidad, frecuencia de backup (semanal), así como los puertos requeridos (típicamente 5432 para Postgres, 3306 para MySQL, 80/8080/5000 para servicios web, entre otros), y se especifica que el acceso debe ser por SSH y, en algunos casos, también por FTP para facilitar el uso por parte del equipo.

Se listan las aplicaciones y herramientas que deberán estar instaladas o disponibles en el servidor de procesamiento: Python, Conda, Git, R, Pip y Nginx, entre otras, aclarando que muchas librerías específicas se instalarán desde los propios entornos del equipo aprovechando los permisos de administrador que se solicitan. Además, se conversa sobre la necesidad de ceder en el futuro los proyectos de Google Cloud al propietario institucional de INVIAS, para no dejar la infraestructura ligada a correos personales y reducir riesgos de responsabilidad.

4. Agenda simplificada

- Configuración de cuentas de servicio y credenciales (JSON) para Google Cloud Storage.
- Demostración del flujo automatizado de descarga y ejecución de MaxEnt en el RAC.

- Revisión de la arquitectura de preprocesamiento–procesamiento–postprocesamiento y programación paralela.
- Diligenciamiento de formatos para solicitud de servidores (base de datos y procesamiento) ante la OTIC/INVIAS.
- Discusión sobre accesos, puertos, backups y herramientas a instalar en los servidores.

5. Revisión de compromisos pasados

- Se constata que el sistema de descarga automatizada desde el bucket ya está implementado y funcional, cumpliendo un compromiso técnico previo.
- Se verifica que el diseño de la arquitectura con dos servidores (base de datos y procesamiento) es consistente con lo acordado en reuniones anteriores.
- Se recuperan y actualizan los puertos y componentes que ya se habían mencionado en versiones previas de los formatos, ajustándolos a la nueva separación de responsabilidades.

6. Próximos pasos y posibles fechas

- Luis enviará las credenciales JSON generadas (comprimidas) a Jairo para integrarlas definitivamente en los scripts del RAC.
- Andres se encargará de enviar a Marli/INVIAS los formatos diligenciados de solicitud de servidores en la fecha acordada (25 de noviembre), de modo que la OTIC pueda provisionar la infraestructura.
- Jairo continuará afinando el flujo de automatización (supervisor, colas, programación paralela) y documentándolo en manuales técnicos.
- Una vez los servidores estén disponibles, el equipo (incluyendo Nelson y Jasmín) realizará pruebas de punta a punta desde el bucket hasta la generación de resultados finales, para luego integrarlo con Hermes/ArcGIS Enterprise.

7. Hitos y conclusiones relevantes

- Queda plenamente demostrado que el flujo desde el bucket de Google Cloud Storage hasta la ejecución de MaxEnt en el RAC es automatizable y ya cuenta con una implementación funcional.

- Se consolida el diseño de arquitectura con dos servidores diferenciados, permitiendo separar la lógica de base de datos de la lógica de procesamiento intensivo.
- Se documentan con precisión las necesidades de puertos, accesos, backups y herramientas, lo que facilita la gestión institucional de la infraestructura.
- Se refuerza la importancia de transferir, en el mediano plazo, la propiedad de los proyectos de nube y la operación cotidiana a INVIAS, para reducir dependencias y responsabilidades sobre el equipo externo.

8. Análisis y recomendaciones

Esta reunión marca un punto importante en la operacionalización del modelo: ya no se trata solo de algoritmos, sino de cómo se conectan los servicios reales (nube, servidores, backups, accesos). El equipo avanza hacia una arquitectura reproducible y defendible ante las exigencias de INVIAS y de la OTIC.

Recomendaciones:

- Mantener actualizados los diagramas de arquitectura (por ejemplo en Draw.io) y compartirlos siempre que se envíen formatos técnicos, para reducir malos entendidos con las áreas de TI.
- Asegurar que todos los scripts de automatización tengan logs claros y mecanismos de alerta, de modo que el equipo pueda detectar fallos en descargas, ejecuciones o escritura de resultados.