

Acta de Reunión

Capacitación en GitHub – INVIAS - Equipo de desarrollo Mapa de Vulnerabilidad Faunística

Fecha: viernes, 5 de septiembre de 2025

Lugar: Conference Room (JI CC)

Duración: ~1 h 06 min

Organización/Proyecto: INVIAS – Equipo de desarrollo Mapa de Vulnerabilidad Faunística

1. Propósito de la reunión

Capacitar al equipo en el uso de GitHub como plataforma central del proyecto de INVIAS, abarcando repositorios, wikis, tableros (Projects), roles y flujo de trabajo con ramas y pull requests. Alinear el stack tecnológico y dejar tareas inmediatas para iniciar la implementación.

2. Asistentes

- Luis Esteban Gómez
- Andrés Felipe Hernández
- Nelson Aníbal Miranda Ríos
- Jasmin
- Jairo Coy

3. Desarrollo de la reunión

Se presentó la organización en GitHub de INVIAS con cuatro repositorios: estructura de base de datos, interfaz gráfica, modelos analíticos y documentación. Se demostró el uso de la wiki para centralizar manuales, especificaciones y capítulos con jerarquías y páginas. Se formalizaron cinco equipos de trabajo (Backing, Database, DevOps, Frontend, Integration) y se solicitó aceptar las invitaciones enviadas por GitHub a más tardar el domingo. Se revisaron responsabilidades y se precisó que la fase de integración involucra a todo el equipo. Se mostró GitHub Projects como tablero para registrar el backlog individual (microservicios, ETL, automatización y modelos), asignar responsables y calendarizar en formato tipo Gantt. Cada integrante debe listar sus tareas y autoasignarse para facilitar la planeación. En el plano técnico se definió el stack: Django (framework web), Python (lenguaje), SQL con motor PostgreSQL/PostGIS, e integración de R mediante rpy2 para implementar el modelo de máxima entropía (MaxEnt). Se acordó explorar la integración con ArcGIS Portal (Enterprise), dado el licenciamiento vigente de INVIAS. Se creó un Project específico para que cada integrante liste las librerías de Python y R requeridas; con base en ello se generará un script de instalación automatizada en el servidor. Se discutió la conveniencia de manejar datos vía APIs (p. ej., Google Earth Engine) y persistir capas en base de datos para mejorar rendimiento. Como práctica de calidad, se explicó el flujo de ramas en tres niveles (desarrollo → supervisión → despliegue) y el uso de pull requests para revisión previa. Se pactó una capacitación corta adicional para PRs. Finalmente, se estableció una cadencia de reuniones quincenales (viernes 10:00–11:00) más seguimientos semanales focalizados.

4. Agenda (simplificada)

- 1 Organización en GitHub y uso de wikis.
- 2 Equipos, roles y aceptación de invitaciones.
- 3 Backlog por persona y planificación en Projects (Gantt).
- 4 Stack tecnológico: Django, Python, PostgreSQL/PostGIS, R (rpy2), ArcGIS Portal.
- 5 Listado de librerías y automatización de instalación en servidor.
- 6 Roles del aplicativo y estrategia de datos (APIs/BD).
- 7 Flujo de ramas y pull requests; capacitación breve próxima.
- 8 Cadencia de reuniones (quincenales + seguimientos).

5. Revisión de compromisos pasados

Invitaciones a la organización y equipos: enviadas; pendiente aceptación general (incluida Jasmin). Se retoma la necesidad de formalizar los requerimientos de infraestructura para INVIAS mediante correo técnico el mismo día.

6. Próximos pasos

- Aceptar invitaciones a la organización/equipos en GitHub (plazo: domingo 7 de septiembre de 2025).
- Registrar backlog individual en el Project de 'Modelamiento y Automatización' y asignarse tareas.
- Listar librerías de Python y R en el Project 'Librerías de modelos de procesamiento' para automatizar instalación.
- Enviar a INVIAS (Iván/Marley) correo con requerimientos de infraestructura (hoy).
- Explorar integración con ArcGIS Portal y presentar avances en reunión 1:1 del lunes siguiente.
- Programar mini-capacitación sobre pull requests y flujo de ramas (5–10 min).
- Crear repositorio de actas y centralizar PDFs/DOCs.
- Mantener reuniones quincenales (viernes 10:00–11:00) + seguimientos semanales focalizados.

7. Hitos y conclusiones

• Stack acordado: Django + Python + PostgreSQL/PostGIS; integración de R (rpy2) para MaxEnt. • Confirmación del uso de ArcGIS Portal como motor de mapas institucional. • Cinco equipos en GitHub (Backing, Database, DevOps, Frontend, Integration) con responsabilidades definidas. • Flujo de calidad: pull requests y capa de supervisión antes de deploy. • Tareas inmediatas: aceptar invitaciones, crear backlog y listar librerías. • Pendiente por decidir: módulo de roles/administración en la aplicación.

8. Análisis y recomendaciones

La sesión logró alinear expectativas y estándares de trabajo. Recomendado definir un Definition of Done (DoD) por tipo de tarea, activar plantillas de issues con criterios de aceptación y establecer protección de ramas con revisiones obligatorias. Documentar versiones de R/paquetes (dismo, maxnet, raster/terra) y dependencias de rpy2 para asegurar reproducibilidad. En infraestructura, precisar SO, CPU/RAM, almacenamiento, acceso (VPN), dominio/SSL y políticas de backup/restauración. Para datos, mantener un catálogo de APIs/capas con frecuencia, autenticación y responsables, priorizando cacheo en BD para rendimiento.