

Taller 4 - Gestión de Proyectos de Investigación y Ciencia Abierta

Nombre del Estudiante: Julio Alejandro Gomez Orduz

Código: 202315618

Desarrollo previo al taller:

Dado que mi desarrollo del Taller 3 presentó deficiencias iniciales, fue necesario reforzar y reaprender aspectos fundamentales relacionados con la correcta creación y gestión de un repositorio Git, así como la sincronización adecuada de commits hacia GitHub. En este proceso decidí reconstruir el pipeline de trabajo desde cero, asegurándome de que quedara estructurado de forma reproducible y coherente.

En consecuencia, el Taller 4 se desarrolló a partir de la versión mejorada del proyecto correspondiente al Taller 3 (versión 3.2), ya depurada y funcional.

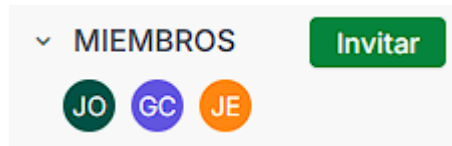
Desarrollo del taller:

PARTE 2 - GESTIÓN DEL PROYECTO CON QUIRE

1. *Vea el tutorial de Quire disponible en Bloque Neón y utilice la aplicación para gestionar la solución de este taller. Configure un proyecto para el taller en su organización y planifique su solución. Para esto, necesitará leer todo el taller primero para conocer las tareas a realizar.*
 - a. **Creé la planificación en Quire y asigné fechas de entrega (todo para el 22 de Febrero)**
 - b. **Creé un nuevo proyecto de git (debido a que básicamente la idea es replicar el taller 3)**
 - i. **Más adelante utilizaré los mismos scripts y replicaré el trabajo base para trabajar con las branches**
2. *Realice su planificación dentro de Quire. Escriba sus tareas, use etiquetas cuando y como sea necesario y cambie el estado de las tareas según corresponda.*
 - a. **Creé una planificación en Quire incluyendo todas las etapas que consideré necesarias**

3. Invite a j.echeverryd@uniandes.edu.co y a ga.castillo@uniandes.edu.co como miembros.

a. **Los invité - OK**



✓	1. Configuración del proyecto y entorno	Hoy	JO
✓	Inicializar repositorio	Terminado Hoy	JO
✓	Configurar Renv (para reproducibilidad)	Terminado Hoy	JO
✓	Crear Tree de carpetas	Hoy	JO
✓	Configurar gitignore	Hoy	JO
✓	2. Implementación de pipeline reproducible	Hoy	JO
✓	Crear Script Simulación	Hoy	JO
✓	Crear Script Procesamiento	Hoy	JO
✓	Crear Script Analisis	Hoy	JO
✓	Crear Script Plot	Hoy	JO
✓	Crear Runall.ps1	Hoy	JO
✓	3. Control de versiones con GitHub	Hoy	JO
✓	Commits por fase (aprovechar la IA de github anclada a VSCODE)	Hoy	JO
✓	Simular conflictos		JO
✓	Push al repositorio (branch master)	Hoy	JO
✓	Hacer pruebas	Hoy	JO
✓	Crear un readme.md	Hoy	JO
✓	Crear una nueva branch para trabajar un análisis de datos más avanzado		JO
✓	Crear una rama para modelado		JO
✓	Unir branches (merge) a main		JO
✓	4. Diagrama de workflow en Mermaid	Hoy	JO
✓	Definir flujo científico	Hoy	JO
✓	5. Diagrama de workflow en Draw.io	Hoy	JO
✓	Replicar flujo	Hoy	JO
✓	Optimizar metodología	Hoy	JO
✓	Exportar drawio e imagen	Hoy	JO
○ ↑	6. Documentación y entrega final	Hoy	JO
	Redactar documento PDF	Hoy	JO
	Adjuntar diagramas	Hoy	JO
	Validar repositorio público	Hoy	JO

b. **Así mismo le coloqué fechas de hoy y me asigné todas las tareas**

- c. **Dado que ya me había adelantado el paso de inicializar el repositorio, lo marqué cómo completado y procedí a continuar con el taller**
4. *Dado que Quire realiza un seguimiento de las actividades en el proyecto, no necesitará entregar nada para esta parte. Puede ver los datos de progreso del proyecto en la pestaña “Overview”, que presenta un gráfico circular con información general sobre el estado de varias tareas, y una línea de tiempo con información similar. TENGA EN CUENTA que para la calificación el profesor deberá tener acceso al proyecto Quire. También tenga en cuenta que el historial de actividades de lo que sucede con las tareas (fecha de creación, comentarios, archivos adjuntos, cambios de estado, etc.) permanece dentro del proyecto, por lo que evite intentar hacerlo todo en el último minuto (planificación ex-post) para simular como si realmente estuviera usando la herramienta, ya que será evidente. El objetivo de este ejercicio es únicamente practicar el uso de esta herramienta en un entorno real y seguir los obstáculos y el progreso*
- a. **Se puede evidenciar que fui desarrollando paso a paso usando QUIRE (excepto parte de la primera sección, que me adelanté un poco).**

PARTE 2 - PRÁCTICA CON GITHUB (25%)

- *Siga el tutorial en el cuaderno R markdown sobre conflictos, fusiones y ramas de GitHub. Esaá disponible en Bloque Neón*
 - **Se desarrolló el tutorial de la forma más paralela. En un punto tuve un merge problemático y lo resolví. Creo que fué el mejor ejemplo (a diferencia de la receta pues, yo no estaba haciendo recetas y eso no es coherente con el taller).**
 - **El flujo de trabajo de acuerdo a la guía del ejercicio fue la siguiente:**
 - **Creación de un nuevo repositorio en GitHub.**
 - **Implementación de la estructura de directorios del Taller 3.**
 - **Uso de ramas (branches) para análisis exploratorio y modelado.**
 - **Fusión (merge) de ramas hacia la principal.**
 - **Generación y resolución de un conflicto de merge real.**
 - **Uso de commit --amend para corregir un commit.**
 - **Recuperación de un archivo eliminado accidentalmente.**

- **Commits realizados y pusheados, se puede verificar en el repositorio Github de carácter público:** <https://github.com/juagomezor-max/GPI-Workshop4v2>

PARTE 3 - DIAGRAMAS DE FLUJO DE TRABAJO CON MERMAID JS

Desarrollé el flujo de trabajo usando MERMAID y me apalanque revisando el código original con Copilot de VSCODE. Al final el resultado me pareció bastante satisfactorio. Dado que en un punto incluí unas funciones auxiliares, que finalmente no empleé demasiado, decidí no mejorar esa parte.

La mejora metodológica en este punto, podría darse en ese sentido. Planificar más desde el uso de funciones generales. Sin embargo, dado que realmente no reciclé muchas funciones, no lo ví necesario. Sin embargo, las funciones por ejemplo para generar gráficos estandarizados, serían de buen uso para una siguiente fase del desarrollo de mi código.

PARTE 4 - DIAGRAMAS DE FLUJO DE TRABAJO CON [DRAW.IO](https://draw.io)

Usando [Draw.io](https://draw.io) repliqué el mismo gráfico de mermaid. Me gustó especialmente que puedo interactuar más con la distribución espacial del mapa mental. Considero que el ejercicio visual es positivo, para poder hacer una planificación más efectiva.

REFLEXIÓN FINAL

El desarrollo de este taller permitió corregir las fallas evidenciadas en el taller 3, mejorar mis habilidades de Gestión estructurada de proyectos, manejar un control de versiones distribuido, gestión de conflictos (básicos pero reales) entre branches y recuperación ante errores. Finalmente, con MERMAID y [DRAW.IO](https://draw.io), aprendí a generar flujos de trabajo desde código y gráficamente.

FIN.