

Taller 4: Gestión de Proyectos y Flujos de Trabajo Colaborativo

Gestión de Proyectos de Investigación y Ciencia Abierta

Fecha de entrega: Ver Bloque Neón

Objetivos

- Implementar herramientas de gestión de proyectos para la investigación reproducible
- Practicar el manejo de colaboración en GitHub con ramas y fusiones
- Desarrollar habilidades para la visualización de flujos de trabajo científicos
- Documentar procesos computacionales mediante diagramas estandarizados

Parte 1: Gestión de Proyectos con Quire (25 %)

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.

1. Cree un proyecto llamado “GPI-workshop4”
2. Invite a j.echeverryd@uniandes.edu.co y a ga.castillo@uniandes.edu.co como miembros.
3. 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.
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.

NOTA: Si lo desea, Quire también tiene una aplicación móvil gratuita para teléfonos Android e iOS (yo mismo la uso para la gestión de proyectos). Si le resulta más cómodo tenerla en su teléfono móvil, está invitado a hacerlo. La actividad realizada allí sigue registrándose dentro del proyecto Quire.

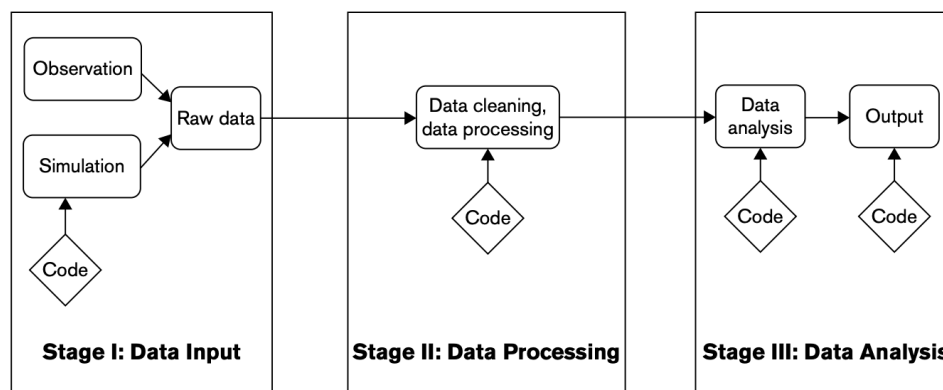
Parte 2: Práctica con GitHub (25 %)

- Siga el tutorial en el cuaderno R markdown sobre conflictos, fusiones y ramas de GitHub. Está disponible en Bloque Neón [aquí](#).

1. Asegúrese de que todos sus cambios estén confirmados (commit) y enviados a GitHub
2. Asegúrese de que su repositorio sea público
3. Envíe la URL de su repositorio de GitHub

Parte 3: Diagramas de Flujo de Trabajo con Mermaid JS (25 %)

Usted ha escrito previamente un flujo de trabajo automatizado en el Taller Semanal 3 basado en un ejercicio de análisis ficticio que se ha llevado a cabo a lo largo de los talleres anteriores. Ahora seguirá los pasos de los autores de los estudios de caso en Kitzes et al. (2017). Recuerde que ya vimos buenos ejemplos de diagramas de flujo de trabajo en los Estudios de Caso 10 y 11, así como este que representa el marco centrado en datos que proporciona Kitzes et al. (2017):



Three main stages of data-centric scientific research.

La tarea será precisamente producir un gráfico de flujo de trabajo como este. Si sus rutinas no son lo suficientemente adecuadas para un proceso adecuado o no está satisfecho con ellas, está perfectamente bien revisarlas y mejorarlas. Puede guiarse siguiendo [la documentación](#) del sitio web oficial.

Exporte su diagrama como una imagen e inclúyalo en el documento del Taller 4. Además, incluya el código que produjo ese diagrama en un archivo separado llamado `diagram.mermaid`. Utilice la extensión `.mermaid`; si no está seguro de cómo guardar un archivo de texto plano con una extensión personalizada, recuerde el uso del comando `New-Item` en PowerShell (y `touch` para los usuarios de macOS) del Taller 1.

Parte 4: Diagramas de Flujo de Trabajo con Draw.io (25 %)

A continuación, tome el mismo flujo de trabajo que graficó previamente en la parte 3 y use la aplicación gratuita en línea [Diagrams.net](#) (anteriormente llamada Draw.io). Dado que esta herramienta nos permite crear gráficos usando una interfaz gráfica de usuario más interactivamente, podría considerarse mucho más fácil y tentador mejorar nuestro diagrama escrito con Mermaid JS. Incluya una característica adicional al diagrama de la Parte 3. Puede inspirarse en cualquiera de los estudios de caso de Kitzes et al. (2017). Recuerde que todo el libro es de libre acceso en <http://www.practicereproducibleresearch.org/>.

Exporte su diagrama como una imagen e inclúyalo en el documento. También incluya en su envío el archivo `.drawio` que utilizó. Tenga en cuenta que esta extensión de archivo también es un formato de texto plano, más precisamente un formato basado en XML que codifica la estructura, los metadatos y el estilo del diagrama, que luego se representa como gráficos vectoriales por la aplicación draw.io (diagrams.net). Menciono esto para recordarle que no sería necesario ignorar este archivo con gitignore, dado que no es en sí un archivo “binario”.

Entrega

- Documento PDF con sus respuestas e imágenes de diagramas
- Archivo `diagram.mermaid` con el código Mermaid
- Archivo `.drawio` con su diagrama
- URL del repositorio GitHub con el ejercicio de la Parte 2
- La parte 1 se evaluará directamente desde el proyecto Quire compartido