

1. Organización y Planificación (Issue #1)

1.1. Gestión de Tiempos

- **Desglose:** Dividir el proyecto en tareas manejables (usando Issues y el Project Board de GitHub).
- **Asignación:** Definir responsables para cada tarea. Aunque trabajemos individualmente, esta práctica es clave.
- **Hito Clave:** Establecer fechas límite internas, dejando un margen de 2-3 días antes de la fecha de entrega real para revisión final.

1.2. Herramientas Colaborativas

- **Control de Versiones:** Usar **Git y GitHub** es obligatorio. Las ramas (**feature/nombre**) aíslan el trabajo y los Pull Requests aseguran la revisión de pares.
- **Comunicación:** Establecer un canal único para dudas rápidas y otro para reuniones formales (ej: Slack, Microsoft Teams).

2. Formato Académico y Diseño (Issue #2)

2.1. Estructura Estándar

- **Partes Clave:** Portada, Índice (generado automáticamente), Introducción, Desarrollo por Secciones, Conclusiones y Referencias.

2.2. Guía de Diseño

Elemento	Especificación
Márgenes	2.54 cm (1 pulgada) en todos los lados.
Fuente	Times New Roman o Arial, tamaño 11pt.
Interlineado	Doble (2.0) en todo el documento.
Alineación	Izquierda.

3. Citación y Referencias (Issue #3)

Todo el contenido no original debe estar debidamente citado para evitar el plagio.

3.1. Estilo y Formato

Se utilizará el formato **APA (7ª Edición)** para las referencias.

Tipo de Cita	Ejemplo
Narrativa	Smith (2023) argumenta que el uso de Git...
Parentética	La colaboración mejora el control de versiones (Smith, 2023).

3.2. Sección de Referencias

La lista de referencias se ordenará alfabéticamente al final del manual.

Ejemplo de Libro: Apellido, A. A. (Año). *Título del libro en cursiva*. Editorial.

4. Presentación y Entrega (Issue #4)

4.1. Revisión Final

Antes de la entrega, realizar una revisión cruzada (o auto-revisión detallada) para verificar:

1. Coherencia de formato.
2. Ortografía y gramática.
3. Completitud de las referencias (que toda cita tenga su referencia y viceversa).

4.2. Versiones y Etiquetado (Tags)

El documento final debe ser etiquetado en GitHub (ej: **v1.0**), y el archivo **CHANGELOG.md** debe reflejar los cambios significativos que componen esa versión final.