

1. Buenas prácticas de organización del equipo

1. Buenas prácticas de organización del equipo

1.1. Comunicación

- Definir un canal principal (WhatsApp, Telegram, Discord).
- Establecer horarios aproximados de respuesta.
- Registrar decisiones importantes dentro del repositorio (en Issues o en el README).
- Preferir comunicación escrita para mantener evidencia.

1.2. Roles y responsabilidades

- **Coordinador del equipo:** organiza tareas y reuniones.
- **Encargado de documentación:** administra el contenido del manual.
- **Encargado del control de versiones:** gestiona ramas, PR y merges.
- **Encargado de revisión:** valida contenido antes de aprobar cambios.
- **Encargado de diseño o formato:** estandariza estilos y estructura visual.

1.3. Reuniones y planificación

- Realizar reuniones cortas de seguimiento (10–15 minutos).
- Documentar acuerdos y tareas en GitHub Projects.
- Crear Issues para cada actividad asignada.
- Utilizar el tablero Kanban (To Do → In Progress → Done).

1.4. Gestión del tiempo

- Dividir el trabajo en tareas pequeñas y manejables.
- Establecer fechas claras para cada entregable.
- Utilizar etiquetas en Issues:
 - `prioridad-alta`
 - `documentación`
 - `en-revisión`
 - `completado`

2. Formato del documento

2.1. Estructura sugerida

- Portada
- Índice
- Introducción
- Desarrollo (con subtítulos claros)
- Conclusiones
- Bibliografía
- Anexos

2.2. Reglas de estilo

- Fuente recomendada: Arial, Calibri o Roboto.
- Tamaño: 11–12 pt.
- Interlineado: 1.15 o 1.5.
- Márgenes estándar: 2.54 cm.
- Evitar bloques grandes de texto; usar listas, tablas y títulos descriptivos.

2.3. Imágenes y tablas

- Guardar archivos en la carpeta `/imagenes`.
- Nombrar archivos como: `figura_01.png`, `tabla_02.jpg`.
- Referenciar en el texto de esta forma:

2.4. Escritura

- Redacción formal y clara.
- Evitar muletillas y repeticiones.
- Revisar ortografía antes de enviar un Pull Request.
- Utilizar lenguaje inclusivo y profesional.

3. Cómo trabajar con ramas en Git

3.1. Flujo recomendado

1. Crear una rama por sección:

```
git checkout -b seccion-formato
```

Utilizar etiquetas en Issues: prioridad-alta, documentación, en revisión, completado.

3.2. Nombres recomendados para ramas

- `seccion-buenas-practicas`
- `seccion-formato`
- `seccion-rama`
- `seccion-conflictos`
- `PDF-final`

3.3. Buenas prácticas

No trabajar directamente en main. Hacer commits pequeños y descriptivos. Sincronizar antes de empezar → `git pull origin main`.

4. Citas y referencias

4.1. Estilo recomendado

Usar el formato APA (7.^a edición) para todos los trabajos académicos.

4.2. Cómo citar dentro del texto

Autor y año entre paréntesis: (Pérez, 2020) Cita textual menor de 40 palabras: "Texto citado..." (Pérez, 2020, p. 15). Cita de más de 40 palabras: usar sangría.

4.3. Lista de referencias

Debe incluir:

- Autor
- Año
- Título
- Editorial o fuente
- Enlace (si aplica)
- Ejemplo: Pérez, M. (2020). Fundamentos de investigación académica. Editorial Alfa.

4.4. Herramientas recomendadas

Google Scholar (citar automáticamente) Zotero Mendeley Generador APA de Scribbr o Normas APA

4.5. Buenas prácticas

Verificar que todas las citas aparezcan en la bibliografía. Evitar enlaces rotos. No citar Wikipedia directamente: usar fuentes primarias. Mantener las referencias en la carpeta /referencias.

5. Generación del PDF final

5.1. Si el manual está en Markdown

Puedes convertirlo usando un convertidor online o Pandoc: `pandoc manual.md -o manual.pdf`

5.2. Si está en Google Docs o Word

Exportar como PDF. Asegurarse de que los títulos y numeraciones estén correctos. Guardar el PDF en la carpeta /documentos.

5.3. Versión final

Etiquetar la versión → v1.0. Registrar cambios en CHANGELOG.md: Secciones añadidas Correcciones Revisiones Subir el PDF con un mensaje claro: "Versión final del manual para entrega – v1.0"