

# Guía de Taller Práctico

## Gestión Integrada Scrum + GitHub

**Asignatura:** Desarrollo de Software

**Duración:** 120 minutos

**Metodología:** Aprendizaje Basado en Proyectos (ABP)

### 1. Introducción

El objetivo de este taller es simular un entorno real de trabajo. No solo programaremos; gestionaremos un proyecto utilizando **Scrum** para la organización y **GitHub** para la colaboración técnica. El resultado final no será solo una aplicación, sino un **Documento de Gestión Maestro** construido de forma colaborativa mediante ramas de Git.

### 2. Formación de Equipos y Roles (15 min)

El grupo se dividirá en células de **3 personas**. Cada integrante asumirá un rol de gestión y una parte del documento final:

1. **Integrante A (Scrum Master / Facilitador):** Responsable de la estructura del documento y la sección de *Planificación*.
2. **Integrante B (Product Owner / Analista):** Responsable de la sección de *Historias de Usuario y Trazabilidad*.
3. **Integrante C (QA / Developer):** Responsable de la sección de *Evidencias de Integración y Retrospectiva*.

### 3. Configuración del Entorno (15 min)

**En Trello o Jira:**

- Crear un tablero con las columnas: **Backlog, To Do, In Progress, Done**.
- Cargar las Historias de Usuario (HU) definidas en el punto 6.

**En GitHub:**

- **Integrante A:** Crea un repositorio público llamado taller-scrum-gestion.
- **Integrante A:** Agrega al Integrante B y C como colaboradores (Settings > Collaborators).
- **Integrante A:** Crea un archivo llamado PROYECTO\_GESTION.md con los títulos principales y hace el primer push a la rama main.
- **Integrantes B y C:** Clonar el repositorio en sus computadoras.

## 4. El Desafío: Flujo de Trabajo (70 min)

### Regla de Oro: El Flujo Git-Flow

Nadie puede trabajar sobre la rama main. Cada integrante debe seguir estos pasos para completar su sección del documento:

1. **Crear una rama propia:**
  - Integrante A: git checkout -b rama-planificacion
  - Integrante B: git checkout -b rama-historias
  - Integrante C: git checkout -b rama-evidencias
2. **Mover la tarjeta** en el tablero de Trello/Jira a "In Progress".
3. **Editar el documento:** Escribir la parte asignada en el archivo PROYECTO\_GESTION.md.
4. **Subir cambios:**
  - git add .
  - git commit -m "feat: completada sección X del documento"
  - git push origin [nombre-de-tu-rama]
5. **Abrir un Pull Request (PR):** En GitHub, solicita fusionar tu rama con main.
6. **Revisión por Pares:** Un compañero debe revisar tu PR, escribir un comentario y aprobarlo.
7. **Merge:** Una vez aprobado, fusionar la rama con main.

## 5. Estructura del Documento Final

### (PROYECTO\_GESTION.md)

Cada equipo debe entregar un archivo Markdown con la siguiente estructura:

#### Sección 1: Planificación (Integrante A)

- Nombres de los integrantes y roles.
- Enlace al tablero de gestión.
- Captura de pantalla del tablero al inicio del Sprint.

#### Sección 2: Historias de Usuario (Integrante B)

- Listado de las 4 historias de usuario trabajadas.
- **Tabla de Trazabilidad:**

ID Historia	Funcionalidad	Rama de Git	Estado
---	---	---	---
HU-01	Ver lista	rama-historias	Finalizado

#### Sección 3: Evidencias y Retro (Integrante C)

- Captura de pantalla del historial de Network en GitHub (donde se vean las ramas).
- **Retrospectiva:**
  - ¿Qué fue lo más difícil de sincronizar?
  - ¿Cómo resolvieron los conflictos de código/texto?

## 6. Historias de Usuario a Desarrollar (Producto)

Mientras documentan, los desarrolladores (equipo) deben crear una app de "Lista de Tareas" muy simple (HTML/JS) para evidenciar el trabajo técnico:

1. **HU-01:** Como usuario, quiero ver una lista de tareas.
2. **HU-02:** Como usuario, quiero agregar tareas mediante un input.
3. **HU-03:** Como usuario, quiero marcar tareas como completadas.
4. **HU-04:** Como usuario, quiero eliminar tareas.

## 7. Criterios de Evaluación

Criterio	Puntaje
<b>Uso de Ramas:</b> Cada integrante trabajó en su rama y no en main.	25%
<b>Pull Requests:</b> Existen PRs con comentarios y aprobaciones de compañeros.	25%
<b>Trazabilidad:</b> Las tarjetas de Trello coinciden con los commits en GitHub.	25%
<b>Documento Final:</b> El archivo MD está completo y bien formateado.	25%