

SCRUM

¿Qué es Scrum?

Scrum es un marco de trabajo ágil utilizado para desarrollar productos complejos, como aplicaciones web, de manera iterativa e incremental. Se enfoca en la colaboración constante, la entrega continua de valor y la mejora continua mediante ciclos de trabajo llamados *sprints*. No es una metodología rígida, sino una guía flexible con roles, eventos y artefactos que promueven la organización y transparencia del trabajo.

Roles

- **Product Owner:** Define qué se debe hacer y prioriza el backlog.
- **Scrum Master:** Guía el proceso, elimina obstáculos.
- **Equipo de Desarrollo:** Construye el producto (devs, UX, QA, etc).

Eventos

- **Sprint (1-4 semanas):** Ciclo de trabajo.
- **Sprint Planning:** Planificación de tareas del sprint.
- **Daily Scrum:** Reunión diaria de 15 minutos.
- **Sprint Review:** Revisión del producto con stakeholders.
- **Sprint Retrospective:** Evaluación del proceso.

Artefactos

- **Product Backlog:** Lista priorizada de funcionalidades.
- **Sprint Backlog:** Tareas del sprint actual.
- **Incremento:** Producto funcional al final del sprint.

Objetivo de Scrum:

Entregar productos funcionales de forma iterativa, rápida y con mejora continua.

Práctica: Gestión de un Proyecto Web con GitHub Projects

Objetivo:

Organizar y dar seguimiento al desarrollo de una aplicación web mediante **GitHub Projects**, usando tableros Kanban, issues y tareas asignadas.

Requisitos Previos

- Tener una cuenta de GitHub.
- Un repositorio creado (puede ser nuevo o uno existente).
- Acceso a **GitHub Projects** (puedes usar la vista "Projects" o "Projects Beta").

1. Crear un GitHub Project

- Ve al repositorio → pestaña **"Projects"** → **"New Project"**.
- Elige el tipo **Board** (Kanban).
- Asigna un nombre: DesWebIntegral_[Iniciales]
- Agrega una descripción.

2. Configurar columnas (Kanban)

Cambiar el nombre de la etiqueta

- BackLog
- Listo
- En Progreso
- En Revisión
- Hecho

3. Crear Issues

Crea al menos **5 issues** representando tareas reales del proyecto:

Issue	Título	Asignar a
#1	Crear estructura base del frontend	Dev 1
#2	Diseñar formulario de registro	Dev 2

Issue	Título	Asignar a
#3	Implementar autenticación con JWT	Dev 3
#4	Conectar frontend con API REST	Dev 1
#5	Subir proyecto a Vercel	Dev 2

Editar los Labels como: frontend, backend, bug, urgente.

4. Asignar y mover tareas

- Asigna cada issue a un colaborador.
- Mueve las tareas según su estado:
 - En cuanto alguien empieza: mover a En Progreso.
 - Cuando finalice y se apruebe: mover a Done.

5. Vincular Pull Requests a Issues

- Cuando trabajes en una tarea, crea una rama:
`git checkout -b feature/formulario-registro`
- Al hacer el **pull request (PR)**, vincúlalo con el issue:
"Closes #2" en la descripción.
- Al hacer **merge**, el issue se cerrará automáticamente y se moverá de columna (si se configura).

6. Monitorea el progreso

- Usa filtros, vistas por persona o etiquetas.
- Puedes activar **charts** en Projects Beta para ver métricas de flujo de trabajo.

Resultado Esperado

- Un tablero visual claro con tareas moviéndose de To Do → In Progress → Done.
- Issues bien definidos, asignados y conectados a PRs.
- Mejor organización, visibilidad del avance y colaboración entre el equipo.