

# Scrum

Ingeniería de Software 1 (CC3090) - 2026

---

# Scrum

Semestre 01, 2026

## Problemática

---

El software es difícil de planificar con exactitud desde el inicio.

- Los requisitos cambian.
- El cliente no siempre sabe lo que quiere hasta que lo ve.
- Un error detectado tarde cuesta mucho más que uno detectado temprano.

## Waterfall

El modelo tradicional de desarrollo se llama Waterfall (cascada).

Se trabaja en fases secuenciales. Una fase no comienza hasta que la anterior termina.

Requisitos → Diseño → Desarrollo → Pruebas → Entrega

## El problema de Waterfall

- Si el cliente cambia un requisito a mitad del proyecto → se rehace una fase entera.
- Si se detecta un error en pruebas → se regresa al desarrollo.
- El cliente ve el producto por primera vez al final → puede no ser lo que esperaba.
- Si el proyecto dura 12 meses → el mercado puede haber cambiado.

Waterfall funciona bien cuando los requisitos son estables y conocidos desde el inicio.

## Agile

---

Agile no es una metodología.

Es una filosofía — un conjunto de valores y principios para guiar decisiones.

Sobre esa filosofía se construyen marcos de trabajo concretos.

### Agile vs Waterfall

Waterfall:

- Planificación completa al inicio.
- Se entrega al final del proyecto.
- Los cambios son costosos y difíciles.
- La relación con el cliente es puntual.

Agile:

- Planificación iterativa y continua.
- Se entrega de forma frecuente e incremental.
- Los cambios son bienvenidos y esperados.
- La relación con el cliente es continua.

## Marcos de trabajo ágiles

Agile se implementa a través de marcos concretos:

- Scrum → equipos que trabajan en ciclos cortos.
- Kanban → flujo continuo de trabajo.
- XP (Extreme Programming) → enfocado en prácticas técnicas.
- SAFe → Agile a escala empresarial.

## Scrum

---

Scrum es un marco de trabajo ágil para gestionar proyectos complejos.

Se basa en ciclos cortos de trabajo llamados Sprints.

Al final de cada Sprint se entrega un incremento funcional del producto.

[La guía oficial de Scrum \(Español\)](#)

## Pilares

- Transparencia — todos comparten la misma comprensión del trabajo y del avance.
- Inspección — el equipo revisa el progreso con frecuencia.
- Adaptación — el plan se ajusta cuando algo no funciona.

## **Sprint**

Un Sprint es un ciclo de trabajo de 1 a 4 semanas.

- Duración fija — no se extiende.
- Al final se produce un incremento funcional.
- El siguiente Sprint comienza inmediatamente después.

## **Roles**

---

Scrum define tres roles. Cada uno tiene responsabilidades claras.

Un equipo Scrum típico tiene entre 3 y 9 personas.

### **Scrum Master**

Es el facilitador del equipo.

- Guía al equipo en la aplicación de Scrum.
- Elimina impedimentos que bloquean el trabajo.
- Protege al equipo de interrupciones externas.
- Organiza y facilita las ceremonias.

No es un jefe. No asigna tareas ni toma decisiones de producto.

### **Product Owner**

Representa la voz del cliente.

- Define y prioriza el Product Backlog.
- Responde preguntas del equipo sobre los requisitos.
- Acepta o rechaza el trabajo al final del Sprint.

- Puede introducir cambios entre Sprints, no durante.

Es el responsable de maximizar el valor del producto.

## **Development Team**

El equipo que construye el producto.

- Auto-organizado: se distribuye el trabajo internamente.
- Multifuncional: tiene todas las habilidades necesarias.
- Selecciona las tareas en el Sprint Planning.
- Responsable colectivamente de la entrega.

No hay jerarquía interna. Todos son "Developers".

## **Artefactos**

---

Los artefactos son las herramientas que hacen visible el trabajo y el valor.

### **Product Backlog**

Lista ordenada de todo lo que se necesita en el producto.

- Lo mantiene y prioriza el Product Owner.
- Evoluciona durante todo el proyecto.
- Cada elemento es una User Story (historia de usuario).

### **User Story**

Forma de escribir requisitos desde la perspectiva del usuario.

Como [tipo de usuario],  
quiero [funcionalidad]  
para [beneficio].

Ejemplo:

Como estudiante, quiero ver mis calificaciones en tiempo real para planificar mis estudios.

## Story Points

Cada historia tiene un estimado de esfuerzo en Story Points.

Se usa la escala de Fibonacci simplificada:

- 1 → Trivial
- 2 → Pequeño
- 3 → Mediano
- 5 → Grande
- 8 → Muy grande

No miden tiempo — miden esfuerzo relativo.

## Sprint Backlog

Subconjunto del Product Backlog que el equipo se compromete a completar en el Sprint.

- Se define en el Sprint Planning.
- Pertenece al equipo de desarrollo.
- No se modifica durante el Sprint.

## **Incremento**

El resultado tangible del Sprint.

- Es la suma de todo el trabajo completado.
- Debe ser funcional y potencialmente entregable.
- Cumple con la Definición de Terminado.

## **Ceremonias**

---

Las ceremonias son eventos formales que estructuran el Sprint.

- Sprint Planning — inicio del Sprint, 1-2 horas.
- Daily Scrum — cada día, 15 minutos.
- Sprint Review — final del Sprint, 1 hora.
- Sprint Retrospective — tras la Review, 45 min. Se mejora el proceso.

## **Sprint Planning**

Se responden dos preguntas:

- ¿Qué se puede entregar al final del Sprint?
- ¿Cómo se logrará ese trabajo?

El equipo selecciona historias del Product Backlog y define el Sprint Goal.

## **Daily Scrum**

Reunión diaria de máximo 15 minutos.

Cada miembro responde tres preguntas:

1. ¿Qué hice ayer?

2. ¿Qué voy a hacer hoy?
3. ¿Hay algo que me está bloqueando?

No es una reunión de reporte. Es una sincronización del equipo.

## **Sprint Review**

Se presenta el incremento al cliente o stakeholders.

- Se demuestra lo que fue completado.
- El Product Owner acepta o rechaza el trabajo.
- Se recibe retroalimentación para el siguiente Sprint.

No es una presentación formal — es una conversación.

## **Sprint Retrospective**

El equipo reflexiona sobre su propio proceso.

Se responden tres preguntas:

- ¿Qué salió bien?
- ¿Qué se puede mejorar?
- ¿Qué se va a hacer diferente en el próximo Sprint?

El objetivo es la mejora continua del equipo.

## **Tablero**

---

Herramienta visual para rastrear el progreso del Sprint.

Hace visible el estado de cada historia en tiempo real.

## Columnas

TO DO	IN PROGRESS	DONE
Historia 1	Historia 3	Historia 5
Historia 2		

- To Do → historias comprometidas, aún no iniciadas.
- In Progress → trabajo en curso.
- Done → historias que cumplen la Definición de Terminado.

## Definición de Terminado

Criterios que debe cumplir una historia para considerarse completada.

Se define por el equipo antes del Sprint.

- El código fue revisado por otro miembro del equipo.
- Las pruebas pasan sin errores.
- El Product Owner aprobó la funcionalidad.