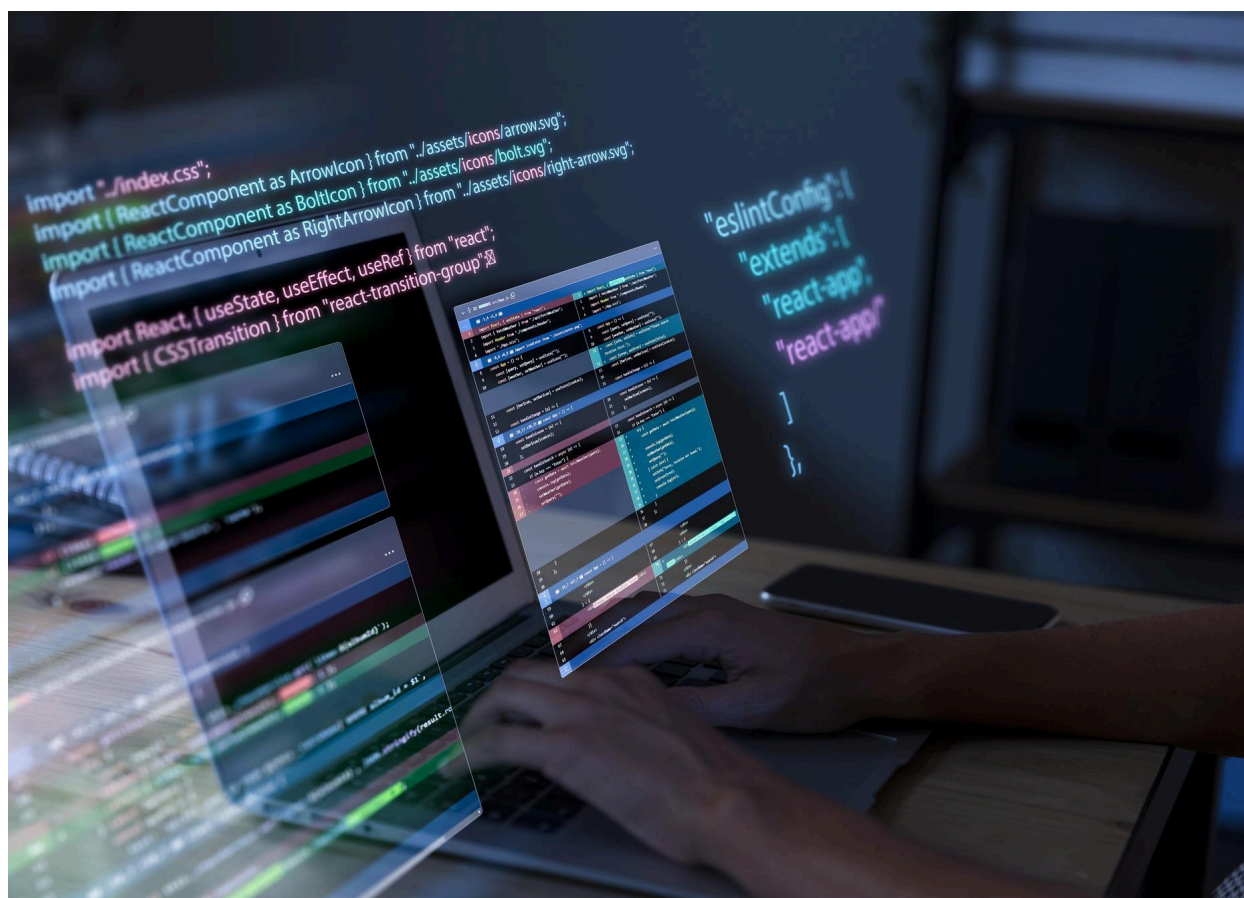


DAMI DAW1 / Entornos de desarrollo

Desarrollo de software

1.5 Métodos Agiles



Autor: Lisa ERIKSEN

Fecha: 2025 / 2026

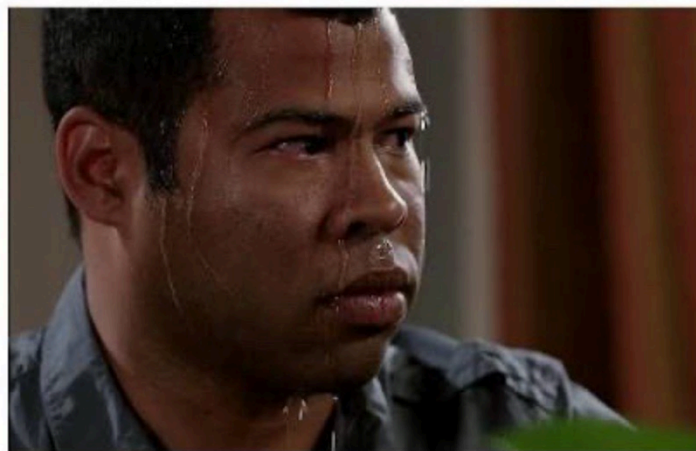
1. Objetivo de aprendizaje

- Comprender la importancia de las metodologías ágiles en el desarrollo de software.
- Conocer el marco de trabajo Scrum y sus componentes principales.
- Identificar y diferenciar otras metodologías ágiles como Kanban, XP y Lean Development.
- Evaluar las ventajas y desventajas de las metodologías ágiles en diferentes contextos de desarrollo.

2. Introducción

En un mundo en constante cambio, los proyectos de software necesitan ser flexibles y adaptativos. Las metodologías ágiles surgieron como respuesta a esta necesidad, ofreciendo ciclos cortos de desarrollo llamados **sprint**, con entregas frecuentes y una comunicación continua con el cliente.

CHECKING YOUR CODE IN ON THE LAST DAY OF THE SPRINT






3. SCRUM

3.1 Organización del Equipo

Scrum es una forma de trabajar en equipo con ciclos cortos llamados *sprints*. Cada sprint tiene un objetivo claro, y el equipo colabora para entregar un producto funcional.

Los 3 roles principales:

Rol	Responsabilidad
 Scrum Master	
 Product Owner	
 Equipo de Desarrollo	

Busca en internet encontrar las responsabilidades de cada uno y completa la tabla:

- *Prioriza las tareas y define el objetivo*
- *Programan y prueban el código*
- *Vigila que el equipo siga los pasos del sprint y resuelve bloqueos*

3.2 Reuniones en SCRUM

Actividad en grupo: Forma un equipo de al menos 3 integrantes y trabajarán juntos en un pequeño proyecto ágil. El trabajo se desarrollará en un Sprint, dividido en varias fases.

👉 Primero, elige a tu equipo y definan los roles de cada miembro según el marco de trabajo Scrum

Rol	Alumnos	Descripción
Scrum Master		Facilita el trabajo en equipo, ayuda a resolver problemas y asegura que se sigan las reglas de Scrum
Product Owner		Elige el programa que desarrollar
Desarrollador/a		Escribe el código, colabora con los demás y prueba el programa

Fase 1 – Sprint Planning (5 min)

👉 El Product Owner elige y presenta el objetivo del sprint al equipo: puede seleccionar uno de esos programas o inventar otro.

🧠 *Quiz: Presenta preguntas y muestra la puntuación final del jugador.*

🎲 *Simulador de dado: Lanza una moneda o un dado y muestra el resultado aleatorio.*

🕒 *Reloj: Muestra la hora actual o una cuenta regresiva.*

🔢 *Conversor: Convierte entre grados, monedas, o distancias (km, millas...).*

👉 Luego, todo el equipo decide qué tareas se realizarán para realizar el programa y las anotan en el Product Backlog:

	Tareas del Sprint (Product Backlog)
1	<i>Ejemplo: Inventar las preguntas y respuestas del Quiz...</i>
2	<i>Ejemplo: Crear el programa en python o java..</i>
3	<i>Ejemplo: Pruebas...</i>
4	

👉 ¿Cómo ha decidido el equipo qué tareas son las más importantes?

Fase 2 – Trabajo del Sprint (10 min)

👉 Durante esta fase, los desarrolladores programan el pequeño proyecto en Python o Java (aquí todo el equipo puede ayudar!)

Fase 3 – Daily Scrum (5 min)

👉 Simulad una breve reunión de equipo (máximo 15 minutos en la vida real, aquí sólo 5).

👉 El equipo responde a tres preguntas:

1. ¿Qué parte del programa he creado?
2. ¿Qué parte del programa podría hacer o mejorar ahora?
3. ¿Qué problema he tenido o tengo ahora?

Fase 4 – Sprint Review (5 min)

👉 Cada equipo presenta su programa a otro grupo de la clase.

👉 El otro grupo actuará como *cliente* y dará feedback.

Fase 5 – Sprint Retrospective (5 min)

El equipo reflexiona sobre el proceso del sprint:

👉 ¿Qué funcionó bien en el equipo?

👉 ¿Qué se puede mejorar?

👉 ¿Crees que Scrum ayuda a organizar mejor el trabajo en grupo? ¿Por qué?

4. Otros marcos y metodologías Ágiles

4.1 Kanban

Método visual para gestionar el trabajo en progreso mediante un tablero con columnas (por hacer, en progreso, hecho)

👉 Diseña tu propio tablero Kanban con 3 columnas: *Por hacer* – *En progreso* – *Hecho*. Incluye en él algunas tareas iniciales (por ejemplo: diseñar la interfaz, crear la base de datos, probar el sistema...). Puedes usar una herramienta digital como [Trello](#)

4.2 Extreme Programming (XP)

Enfocado en mejorar la calidad del software mediante prácticas como programación en parejas y revisión continua.

👉 Busca una ventaja y una desventaja de este método

4.3 Lean Development

Basado en principios de manufactura lean, busca maximizar el valor al cliente reduciendo desperdicios.

👉 Busca una ventaja y una desventaja de este método.

👉 Elige la metodología correcta

Lee cada situación y elige la metodología ágil más adecuada.

Situación	Tu elección
Una app móvil con cambios frecuentes según el feedback	
Mantenimiento de un sistema con tickets constantes	
Un portal web complejo con un gran equipo	
Un proyecto donde se busca eliminar tareas innecesarias	

Para ayudarte:

Se centra en reducir desperdicios y maximizar valor ☐ Lean Development

Equipo grande, funcionalidades complejas, alta calidad del código. ☐ Extreme Programming

Equipo pequeño, cambios frecuentes basados en feedback de usuarios. ☐ Scrum

Actualizaciones frecuentes basadas en solicitudes de usuarios. ☐ Kanban