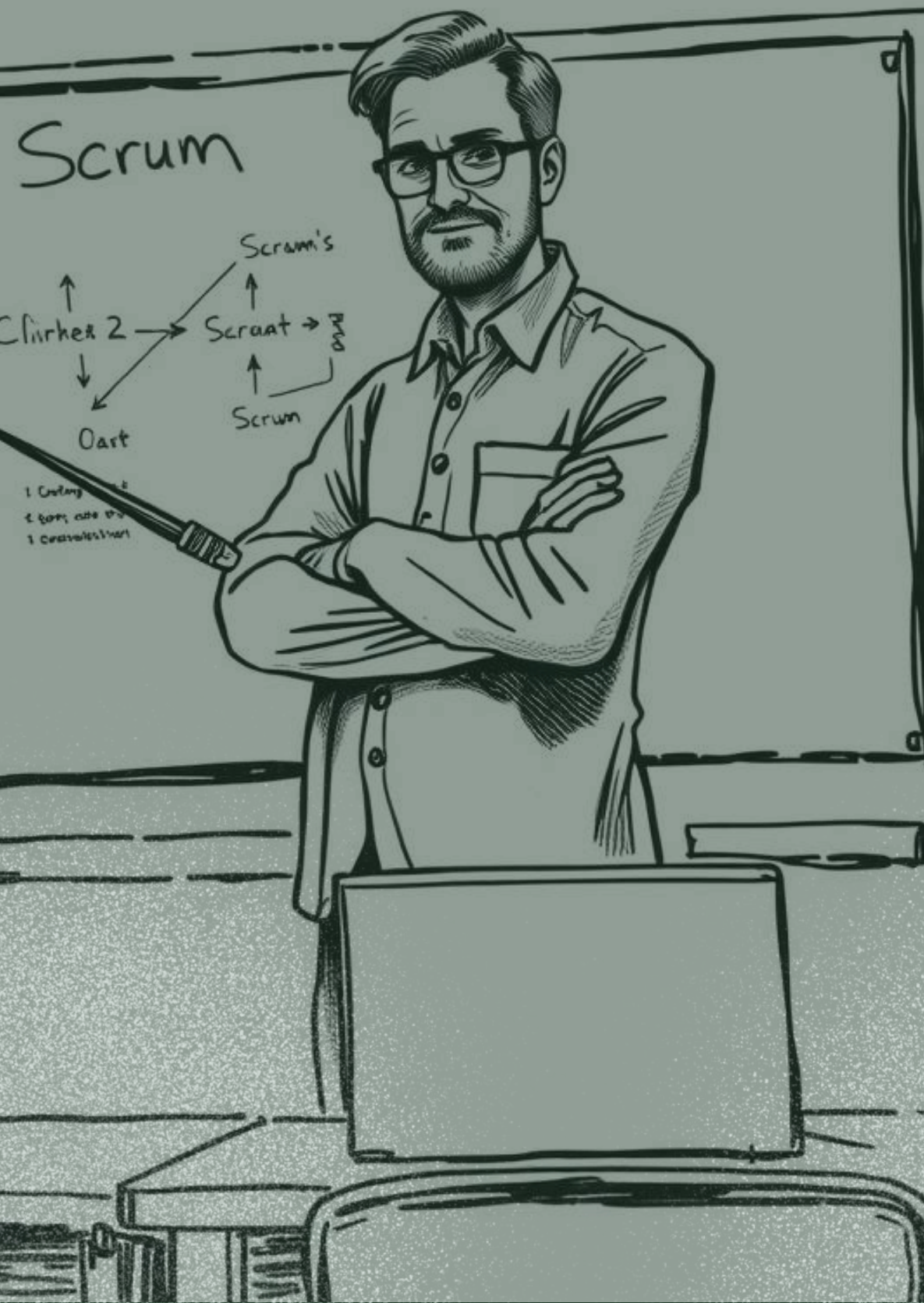


# SCRUM: GUÍA DIDÁCTICA - PRÁCTICA PARA EL AULA Y EL LABORATORIO



 **ALUMNI  
EDITORA  
2025**

PRIMERA EDICIÓN


# SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio

## Autores

### Marco Antonio Checa Cabrera

---

- Ingeniero en Sistemas Computacionales
- Máster en Ingeniería de Software y Sistemas Informáticos

 <https://orcid.org/0000-0002-4169-581X>


 [mcheca@ist17dejulio.edu.ec](mailto:mcheca@ist17dejulio.edu.ec)

---

### Joffre Omar Díaz Ayala

---

- Ingeniero en Sistemas Computacionales

 <https://orcid.org/0009-0005-8795-007X>


 [jdiaz@ist17dejulio.edu.ec](mailto:jdiaz@ist17dejulio.edu.ec)

---

### Paulina Johanna Jácome Ayala

---

- Ingeniera en Sistemas Computacionales
- Máster en Análisis y Visualización de Datos Masivos

 <https://orcid.org/0009-0001-7046-7226>

 [pjacome@ist17dejulio.edu.ec](mailto:pjacome@ist17dejulio.edu.ec)

---


# SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio

## Autores

### Christian Hernán Montalvo Loza

---

- Ingeniero en Sistemas Computacionales
- Especialista en Seguridad Informática

 <https://orcid.org/0009-0005-3405-0043>


 [cmontalvo@ist17dejulio.edu.ec](mailto:cmontalvo@ist17dejulio.edu.ec)

---

### Ana Mercedes Montenegro Rivera

---

- Ingeniera en Sistemas Computacionales

 <https://orcid.org/0009-0001-7290-360X>


 [amontenegro@ist17dejulio.edu.ec](mailto:amontenegro@ist17dejulio.edu.ec)

---

### José Luis Narváez Quinteros

---

- Ingeniero en Sistemas Computacionales

 <https://orcid.org/0000-0002-2196-1615>

 [jlmarvaez@ist17dejulio.edu.ec](mailto:jlmarvaez@ist17dejulio.edu.ec)

---


# SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio

**Autores**

## **Geovana Gabriela Valladares Correa**

---

- Ingeniera en Sistemas Computacionales

 <https://orcid.org/0009-0006-2523-2432>

 [gvalladares@ist17dejulio.edu.ec](mailto:gvalladares@ist17dejulio.edu.ec)

---

# SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio

## Catalogación Bibliográfica

<b>Autores</b>	<ul style="list-style-type: none"><li>• Marco Antonio Checa Cabrera</li><li>• Joffre Omar Díaz Ayala</li><li>• Paulina Johanna Jácome Ayala</li><li>• Christian Hernán Montalvo Loza</li><li>• Ana Mercedes Montenegro Rivera</li><li>• José Luis Narváez Quinteros</li><li>• Geovana Gabriela Valladares Correa</li></ul>
<b>Título</b>	SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio
<b>Descriptor</b>	SCRUM, Metodologías ágiles, Gestión de proyectos, Desarrollo colaborativo, Productividad ágil
<b>Dewey</b>	371.3
<b>Thema</b>	UXJ
<b>Publicación</b>	Febrero 2025
<b>Edición</b>	Primera
<b>ISBN</b>	978-9942-7307-8-7
<b>DOI</b>	<a href="https://doi.org/10.70625/alumned/11">https://doi.org/10.70625/alumned/11</a>
<b>Editorial</b>	Alumni Editora
<b>Pais - Ciudad</b>	Ecuador - Atuntaqui
<b>Formato</b>	Adobe Acrobat Reader
<b>Páginas</b>	113

**Cámara Ecuatoriana del Libro**



Todo el contenido de este libro tiene una licencia de Creative Commons Attribution License.

Reconocimiento-No Comercial-No Derivados 4.0 Internacional (CC BY-NC-ND 4.0).

El contenido del texto y sus datos en su forma, corrección y confiabilidad son de exclusiva responsabilidad del autor y no representan necesariamente la posición oficial de Alumni Editora. Se permite descargar la obra y compartirla siempre que se den los créditos al autor, pero sin posibilidad de alterarla de ninguna forma ni utilizarla con fines comerciales.

# SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio

## Editor en Jefe

Santiago Andrés Otero, PhD., Alumni Editora, Ecuador

## Equipo Editorial

- Óscar Gómez Jiménez, PhD., Universidad Internacional de Valencia (VIU), España
- Shashi Kant Gupta, PhD., Eudoxia Research University, Estados Unidos
- Anabell Fondón Ludeña, PhD., Universidad Rey Juan Carlos, España
- Edwin Ricardo Flores Hernández, PhD., Universidad Salvadoreña Alberto Masferrer, El Salvador
- Gopi Devarajan, PhD., SRM Institute of Science and Technology, India
- Flérida Moreno Alcaraz, PhD., Universidad Autónoma de Sinaloa, México
- J. Suresh Kumar, PhD., St. Joseph University, India
- Mauricio Lima Narváez, PhD., Universidad Técnica del Norte, Ecuador
- Héctor Luis López López, PhD., Universidad Autónoma de Sinaloa, México
- Samuel Helena Tumbula, PhD., Universidad Católica de Angola, Angola
- Carlos Bolivar Sarmiento Chugcho, PhD., Universidad Técnica de Machala, Ecuador
- Savier Fernando Acosta Faneite, PhD., Universidad del Zulia, Venezuela
- Mirian Alexandra Valeriano Meneses, PhD., Instituto Superior Tecnológico Liceo Aduanero, Ecuador
- Sivabalan Settu, PhD., CSE SoCI Vignan University Guntur, India
- Lorena Elizabeth Casanova Imbaquingo, MSc., Instituto Universitario Cotacachi, Ecuador
- Gladys Magdalena Paredes, MSc., Ministerio de Educación, Ecuador
- Henri Emmanuel López Gómez, MSc., Universidad Peruana Los Andes, Perú



El contenido del texto y sus datos en su forma, corrección y confiabilidad son de exclusiva responsabilidad del autor y no representan necesariamente la posición oficial de Alumni Editora. Se permite descargar la obra y compartirla siempre que se den los créditos al autor, pero sin posibilidad de alterarla de ninguna forma ni utilizarla con fines comerciales.



## Revisión de Pares

Este libro ha sido evaluado mediante un proceso de revisión por pares externos bajo el formato de doble ciego. En consecuencia, la investigación presentada en esta obra cuenta con el respaldo de expertos en la materia, quienes han emitido un juicio imparcial basado en criterios científicos, garantizando así la solidez académica del contenido.

## Peer Review

This book has undergone a peer review process by external academics using a double-blind system. Consequently, the research presented in this work has the endorsement of subject matter experts, who have provided an impartial assessment based on scientific criteria, ensuring the academic rigor of the content.



## **Declaración del Editor**

### **Alumni Editora declara para todos los efectos legales, que:**

Esta publicación implica únicamente una cesión temporal de los derechos de autor y de publicación, sin que ello constituya responsabilidad solidaria en la creación de los manuscritos publicados en conformidad con la Ley de Propiedad Intelectual y las normativas legales aplicables.

Autoriza y fomenta que los autores firmen acuerdos con repositorios institucionales con el fin exclusivo de difundir la obra, siempre que se reconozca adecuadamente la autoría y la edición, y que no existan fines comerciales involucrados.

Todos los libros electrónicos publicados son de acceso abierto y, por lo tanto, no se venden en el sitio web de Alumni Editora, ni en plataformas asociadas, de comercio electrónico u otros medios virtuales o físicos, eximiéndose de la transferencia de derechos de autor a los autores.

Todos los miembros del consejo editorial cuentan con el grado académico de cuarto nivel y están vinculados a instituciones de educación superior, conforme a las recomendaciones de las entidades de evaluación académica nacionales e internacionales para la obtención de estándares de calidad editorial.

Alumni Editora no transfiere, comercializa, ni autoriza el uso de los nombres, correos electrónicos u otros datos personales de los autores para fines distintos a la difusión de esta obra.

## **Declaración del Autor**

El autor de la obra declara: 1. no poseer ningún interés comercial que pueda representar un conflicto de interés en relación con el presente documento publicado; 2. Asegura haber participado activamente en la elaboración del manuscrito, específicamente en la concepción del estudio, la obtención de datos y/o su análisis e interpretación; la redacción o revisión del documento para garantizar su relevancia intelectual y la aprobación final del manuscrito antes de su envío; 3. Certifica que el contenido publicado está libre de datos o resultados fraudulentos; 4. Confirma que todas las citas y referencias de datos e interpretaciones de investigaciones previas son correctas; 5. Reconoce haber declarado todas las fuentes de financiamiento recibidas para la investigación; 6. Autoriza la publicación de la obra, que incluye su inclusión en catálogos, asignación de ISBN, DOI, otros índices, diseño visual, portada, maquetación interior, y su posterior difusión según lo dispuesto por Alumni Editora.

## Prólogo

La transformación digital ha impulsado cambios significativos en la manera en que las organizaciones gestionan proyectos, colaboran en equipo y entregan valor a sus clientes. En este escenario dinámico, la adopción de metodologías ágiles ha permitido mejorar la eficiencia, fomentar la adaptabilidad y optimizar los procesos de trabajo.

SCRUM, como marco de trabajo ágil, se ha consolidado como una de las herramientas más efectivas para la gestión de proyectos, facilitando la comunicación entre equipos, promoviendo la entrega iterativa de productos y garantizando la mejora continua. Sin embargo, para aprovechar su potencial, es esencial comprender sus principios, roles y dinámicas de implementación.

Este libro, "SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio", ha sido concebido con un enfoque pedagógico que combina teoría y práctica, ofreciendo a estudiantes, docentes y profesionales una herramienta clara y accesible para la enseñanza y aplicación de SCRUM. A través de una estructura progresiva, los lectores podrán conocer desde los fundamentos de la metodología hasta su implementación en escenarios reales mediante actividades interactivas y prácticas de laboratorio.

El propósito de esta obra es no solo transmitir conocimientos sobre SCRUM, sino también fomentar el desarrollo de habilidades clave como el trabajo en equipo, la comunicación efectiva y la resolución de problemas en entornos dinámicos. La educación y la formación profesional deben evolucionar al ritmo de las exigencias del mundo actual, y este libro busca ser un puente entre el aprendizaje académico y la realidad del trabajo ágil.

Esperamos que esta guía sea de gran utilidad para todos aquellos que deseen comprender y aplicar SCRUM en distintos contextos, convirtiéndose en una referencia fundamental en su camino hacia una gestión más eficiente y colaborativa.

**Los Autores**

## Tabla de contenido

Índice de Figuras .....	10
Índice de Tablas .....	11
Introducción .....	12
Capítulo I: Caracterización de la asignatura.....	15
Introducción .....	16
Antecedentes.....	16
Descripción General de la Asignatura .....	18
Objetivos de la Asignatura.....	18
Objetivo general.....	18
Objetivos específicos.....	18
Resultados de aprendizaje.....	19
Competencias Genéricas .....	20
Perfil de Egreso del futuro profesional .....	20
Capítulo II: Guía didáctica .....	21
Unidad 1 - Enfoques ágiles.....	22
Objetivos de aprendizaje .....	23
Contenidos .....	23
a. Contextos tradicionales y contextos ágiles .....	23
b. El Manifiesto ágil.....	26
Actividades.....	30
Actividad 1: Sopa de Letras .....	30
Actividad 2: Una lo correcto.....	31
Estrategias de enseñanza .....	32
Actividad en Clase: Makerspace.....	32
Duración: .....	32
Modalidad: .....	32
Actividad en Clase: Simulación.....	34
Resultados Esperados: .....	35
Evaluación.....	36
Evaluación: Retroalimentación.....	36
Duración: 30 minutos. ....	36
Tipo: Prueba escrita o en línea.....	36

# Tabla de Contenido

Formato: Preguntas de selección múltiple, verdadero/falso y desarrollo breve.	36
Evaluación de aprendizaje.....	38
Duración: 45 minutos.....	38
Formato: Preguntas de selección múltiple, verdadero/falso y desarrollo breve.	38
UNIDAD 2 – Introducción a SCRUM.....	40
Objetivos de aprendizaje .....	40
Contenidos .....	40
Historia .....	40
Influencia de Nonaka y Takeuchi en el desarrollo de SCRUM .....	41
Definición.....	42
Elementos .....	43
Fundamentos, Pilares y valores .....	43
Actividades.....	45
a. Actividad 1.....	45
Objetivo:.....	45
Instrucciones: .....	45
b. Actividad 2 .....	46
Simulación de un Daily Scrum (Reunión diaria) .....	46
Objetivo:.....	46
Estrategias de enseñanza .....	47
Actividad en Clase: Aprendizaje basado en problemas (ABP).....	47
Objetivo:.....	47
Descripción: .....	47
Pasos de implementación:.....	47
Actividad en clase 2: Gamificación y simulaciones interactivas.....	48
Objetivo:.....	48
Descripción .....	48
Pasos de implementación:.....	48
Evaluación.....	49
Evaluación: retroalimentación.....	50
Objetivo:.....	50
Descripción: .....	50
Estructura del examen: .....	50
Evaluación de aprendizaje: Proyecto práctico de implementación de SCRUM..	51

Objetivo:.....	51
Descripción: .....	51
Pasos del proyecto: .....	51
UNIDAD 3 – Miembros del equipo SCRUM .....	52
Introducción de miembros del equipo SCRUM .....	52
Objetivos de aprendizaje .....	52
Contenidos .....	53
Scrum Team .....	53
Roles y responsabilidades .....	53
b.1. Descripción de cada rol.....	53
Scrum Master: .....	53
Product Owner:.....	54
Equipo de Desarrollo o Developers: .....	54
b.2. Responsabilidades de cada rol .....	55
a. Interacción y Colaboración Scrum Team .....	56
Colaboración del Equipo de Desarrollo .....	57
Interacción con los Stakeholders .....	57
Flexibilidad y Adaptación a Cambios.....	58
Mejora en la Satisfacción del Cliente .....	59
Incremento en la Productividad del Equipo.....	59
Control de Calidad y Reducción de Tiempos de Desarrollo.....	59
Actividades.....	59
a. Actividad 1 .....	59
Estrategias de enseñanza .....	60
a. Actividad en clase: Foro de Discusión .....	60
Objetivo:.....	60
Evaluación.....	61
Evaluación 1 .....	61
Evaluación 2.....	62
UNIDAD 4 – Estructura del trabajo SCRUM.....	64
Objetivos de aprendizaje .....	64
Contenidos .....	64
Actividades.....	70
Actividad 1.....	70

# Tabla de Contenido

Objetivo:.....	70
Preparación:.....	70
Desarrollo: .....	70
Resultado de aprendizaje: .....	72
Actividad 2 .....	72
Objetivo:.....	72
Preparación:.....	72
Desarrollo: .....	72
Estrategias de enseñanza .....	73
Actividad en clase: Asociación Visual de Eventos de Scrum .....	73
Actividad en clase 2 .....	73
Evaluación.....	73
Evaluación 1 .....	73
Evaluación 2.....	74
Capítulo III: Desarrollo de un Caso Práctico.....	76
Aplicando la Metodología Ágil .....	77
Fundamentación.....	77
Objetivos .....	78
Objetivo General.....	78
Objetivos Específicos .....	78
Preparación Previa .....	78
Procedimiento.....	80
Materiales Necesarios.....	80
Paso 1: Análisis de Requerimientos Funcionales .....	81
Paso 2: Elaboración de Historias de Usuario .....	82
Paso 3: Introducción a SCRUM y Jira .....	82
Paso 4: Creación del Proyecto en Jira.....	83
Resultados de la Práctica.....	84
Evaluación del Aprendizaje .....	86
Actividades Complementarias .....	89
Capítulo IV: Gestión de Proyectos con Jira definiendo el Sprint Backlog .....	90
Fundamentación.....	91
Objetivos .....	92
General:.....	92

# Tabla de Contenido

Específicos: .....	92
Preparación Previa: .....	92
Procedimiento: .....	94
Resultados Esperados: .....	98
Evaluación de Aprendizaje: .....	98
Actividades Complementarias: .....	100
Referencias.....	102

## Índice de Figuras

Figura 1.....	26
Figura 2 .....	27
Figura 3 .....	27
Figura 4 .....	28
Figura 5 .....	31
Figura 6 .....	32
Figura 7 .....	56
Figura 8.....	60
Figura 9 .....	64
Figura 10 .....	69
Figura 11.....	70
Figura 12.....	72
Figura 13.....	83
Figura 14 .....	83
Figura 15.....	84
Figura 16 .....	84
Figura 17.....	95
Figura 18 .....	96
Figura 19 .....	96
Figura 20.....	97
Figura 21.....	97

## Índice de Tablas

Tabla 1 .....	19
Tabla 2.....	20
Tabla 3.....	26
Tabla 4.....	58
Tabla 5.....	81
Tabla 6.....	82
Tabla 7.....	86
Tabla 8.....	99

## Introducción

El libro SCRUM: Guía didáctica - práctica para el Aula y el Laboratorio tiene como propósito proporcionar una comprensión profunda y accesible de la metodología Scrum, de una manera didáctica y práctica tanto en el contexto educativo como en el laboratorio. El libro tiene como objetivo enseñar a los estudiantes y profesionales a dominar los principios, roles, artefactos y ceremonias de Scrum a través de ejemplos, actividades y prácticas de laboratorio. Además, busca fomentar la adopción de esta metodología ágil, promoviendo la colaboración y la entrega de valor en proyectos, mientras se refuerzan habilidades clave como la gestión de equipos y la resolución de problemas en situaciones dinámicas y cambiantes.

A lo largo de sus capítulos, el contenido está estructurado de manera clara y progresiva para facilitar la comprensión y aplicación de la metodología Scrum en entornos profesionales a estudiantes de la carrera de desarrollo de software. A continuación, se describe la organización de sus capítulos:

El capítulo uno trata sobre la caracterización de la asignatura dedicada a Scrum, detallando su objetivo principal, que es enseñar a los estudiantes los fundamentos y la aplicación práctica de la metodología ágil, se especifican los resultados de aprendizaje esperados, como la capacidad de gestionar proyectos de software utilizando Scrum en entornos colaborativos, así como también se destacan las competencias genéricas que los estudiantes deben desarrollar, como el trabajo en equipo y la resolución de problemas. Finalmente, se describe el perfil de egreso, que prepara a los estudiantes para enfrentar desafíos en el desarrollo de software de manera ágil.

En el capítulo II se presenta la guía didáctica dirigido a los estudiantes o profesionales que desean aprender Scrum, mismo que está dividido en cuatro unidades, cada uno iniciando con la propuesta del objetivo para luego continuar con el marco teórico correspondiente a la unidad en la que el estudiante conocer de forma progresiva desde el conocimiento general de SCRUM, continuando con los roles, valores, artefactos, eventos asegurando que los estudiantes puedan aplicar Scrum de manera práctica y exitosa.

Los capítulos tres y cuatro está dedicado a las prácticas de laboratorio en la que los estudiantes pueden poner en práctica los conceptos aprendidos en los capítulos anteriores, mediante una serie de actividades diseñadas

específicamente para el laboratorio donde tendrán la oportunidad de trabajar en equipo, gestionar un proyecto utilizando Scrum y experimentar con las fases del ciclo de vida de un sprint. Las prácticas buscan fortalecer la comprensión teórica mediante la aplicación directa de los principios ágiles en un escenario controlado y guiado.

Con la lectura del libro, los estudiantes desarrollaran habilidades para gestionar proyectos de una manera más eficiente y efectiva, utilizando las herramientas ágiles de Scrum y tener productos de software con mejor calidad en el menor tiempo posible.

La presente publicación nace de la intención de ofrecer una guía clara y estructurada que permita a los lectores no solo aprender los conceptos teóricos de Scrum, sino también implementarlos en situaciones prácticas. La combinación de una base conceptual sólida con actividades y prácticas de laboratorio asegura que los estudiantes no solo comprendan la dinámica de Scrum, sino que también puedan aplicar sus principios en proyectos de software reales, desarrollando competencias como el trabajo en equipo, la resolución de problemas y la adaptación al cambio.

Además, el libro busca ser una herramienta que ayude a la falta de recursos accesibles para aprender este tema, por lo que aquí se presenta ejemplos, prácticas de laboratorio y guías paso a paso que permita facilitar la enseñanza de Scrum y fomentar la cultura ágil entre las nuevas generaciones de profesionales.

El presente libro está dirigido principalmente a estudiantes, docentes y profesionales en el contexto del desarrollo de software que buscan conocer y/o profundizar sus conocimientos sobre la metodología Scrum. Su enfoque didáctico permitirá a los participantes, comprender y aplicar de manera práctica los principios ágiles en la gestión de proyectos de software en entornos empresariales, educativos o cualquier área en las que se requieran soluciones informáticas.

La estructura del libro permite al lector no solo conocer los conceptos fundamentales de Scrum, sino también poner en práctica de manera gradual, logrando una integración efectiva de la teoría con la práctica en el aula y/o laboratorio.

Para usar este libro de manera práctica, comience con el Capítulo uno, que presenta la caracterización de la asignatura que trata a Scrum como una

metodología ágil, incluyendo su objetivo, los resultados de aprendizaje esperados y las competencias genéricas a desarrollar, además, se define el perfil de egreso que informa a los estudiantes sobre cómo será su proceso de aprendizaje de SCRUM enfocado al desarrollo de software. Luego, avanza al Capítulo 2, donde se presenta la guía didáctica para utilizarlo en el aula, para que con la realización de las Prácticas 1 y 2 (Capítulos 3 y 4), los estudiantes puedan aplicar los conceptos en proyectos reales. Finalmente, se invita a repasar sobre los resultados de cada práctica para mejorar su comprensión, aprovechando las actividades para adaptar Scrum a diferentes contextos.

Para los docentes, el libro puede ser utilizado como material de apoyo para planificar clases y actividades en el aula, así como también en los proyectos de aula que puedan surgir del conocimiento de SCRUM, que permita consolidar los conocimientos mediante la aplicación prácticas de laboratorio en escenarios reales.

# CAPÍTULO I

## Caracterización de la asignatura



## **Introducción**

La asignatura que incluye el conocimiento de Scrum está diseñada para proporcionar a los estudiantes un conocimiento profundo de las herramientas y prácticas que conforman esta metodología ágil, permitiendo aplicar en el desarrollo de software. Con un enfoque colaborativo, los estudiantes aprenderán a gestionar proyectos de software de manera eficiente, asegurando que se entreguen productos de alta calidad en el menor tiempo posible. Cuando el estudiante conozca y domine los principios de Scrum, como la planificación de sprints, la asignación de roles y la gestión de tareas podrán enfrentar situaciones complejas y dinámicas, optimizando recursos y tiempos de desarrollo mientras garantizan la satisfacción del cliente y la mejora continua del producto.

## **Antecedentes**

El contexto académico y formativo en el que se desarrolla la temática de Scrum como asignatura se enmarca en programas relacionados con la ingeniería de software, desarrollo de aplicaciones y gestión de proyectos tecnológicos, por lo que se busca preparar a los estudiantes en entornos dinámicos y cambiantes para afrontar los desafíos del desarrollo de software.

La asignatura se ubica en el plan de estudios en un nivel intermedio o avanzado de la carrera, dependiendo del enfoque del programa académico, una vez que los estudiantes han adquirido los conocimientos básicos en programación, desarrollo de software y gestión de proyectos, cabe destacar que la asignatura proporciona habilidades directamente aplicables al ámbito laboral en el desarrollo de software y además, contribuye significativamente al desarrollo del perfil profesional de los estudiantes al dotarlos de habilidades necesarias para aplicar los principios ágiles en un entorno profesional dinámico.

Se debe incluir una descripción general del contexto académico y formativo en el que se desarrolla la temática. Esto implica explicar la ubicación de la asignatura en el plan de estudios, especificando el nivel en el que se imparte y su papel dentro de la unidad curricular profesional o básica. Además, es importante destacar cómo esta asignatura contribuye al desarrollo del perfil profesional de los estudiantes, permitiéndoles adquirir competencias clave para su campo de especialización.

Para aprovechar de los contenidos del libro, los estudiantes deben haber adquirido conocimientos previos en varias asignaturas relacionadas con el desarrollo de software y la gestión de proyectos, como, por ejemplo: Fundamentos de Programación, Gestión de proyectos básicos, Modelamiento de software, entre otras.

La comprensión de SCRUM requiere de un estudio profundo, didáctica y práctica, este proceso formativo radica en la creciente demanda de metodologías ágiles en la industria del software.

Otro aspecto fundamental es mencionar los conocimientos previos que los estudiantes deben haber adquirido en asignaturas anteriores para comprender y aplicar los contenidos del libro. Esto incluye habilidades específicas, como programación, manejo de dispositivos electrónicos, diseño de redes, o cualquier otra competencia técnica o teórica directamente relacionada con el tema del libro.

Esta información ayuda a contextualizar el material y a justificar su diseño en función del nivel de preparación de los estudiantes.

También es relevante establecer la conexión entre los contenidos del libro y otras asignaturas o áreas de conocimiento, resaltando cómo los temas abordados se integran con el resto del plan de estudios. Esto promueve una formación integral, mostrando cómo los conocimientos adquiridos en este ámbito específico complementan y enriquecen las competencias generales de los estudiantes.

En este sentido, el libro se enmarca en los programas educativos relacionados con el desarrollo de software, gestión de proyectos tecnológicos e ingeniería de sistemas, en los cuales los estudiantes ya han adquirido conocimientos básicos sobre programación, desarrollo de aplicaciones y gestión tradicional de proyectos y sirve como un puente entre los conocimientos teóricos previos y las demandas actuales del mercado laboral, proporcionando a los estudiantes una metodología ágil y flexible que se ha consolidado como estándar en la industria del software.

## **Descripción General de la Asignatura**

La asignatura enseña a los estudiantes los principios, herramientas y prácticas fundamentales de Scrum, aplicada al desarrollo de software. Presentando una base teórica bien fundamentada y con prácticas de laboratorio objetivas que permitirán a los estudiantes aprender a gestionar proyectos de manera ágil, organizando el trabajo en ciclos cortos (sprints), mejorando la colaboración en equipo y enfocándose en la entrega continua de valor al cliente.

Durante el curso, los estudiantes conocerán los roles clave dentro de Scrum, como el Product Owner, el Scrum Master y el equipo de desarrollo, y aprenderán a implementar los eventos y artefactos necesarios para una ejecución eficiente de la metodología. Al final del curso, los alumnos estarán preparados para aplicar Scrum en entornos profesionales, contribuyendo a la creación de soluciones de software ágiles, eficientes y orientadas al cliente.

## **Objetivos de la Asignatura**

### **Objetivo general**

Conocer los aspectos y procesos necesarios de la metodología Scrum en el desarrollo de software, para su aplicación en la gestión ágil de proyectos de software con el fin de generar productos de calidad en el menor tiempo posible.

### **Objetivos específicos**

- Proporcionar el conocimiento general de las metodologías ágiles, sus principios y beneficios, aplicado al desarrollo de software.
- Introducir a los estudiantes en la metodología Scrum, su historia, los roles fundamentales y cómo Scrum mejora la gestión de proyectos de software en entornos ágiles.
- Enseñar a los estudiantes los principales componentes de Scrum, incluyendo los artefactos, roles y ceremonias, y su aplicación práctica en el desarrollo de software.
- Capacitar a los estudiantes en la estructura y dinámica de trabajo de Scrum, tanto en la planificación de sprints, como la ejecución de tareas y la evaluación continua del desarrollo de los proyectos de software.

# Resultados de aprendizaje

**Tabla 1**  
Resultados de aprendizaje

<b>Objetivo Específico</b>	<b>El/a estudiante al final del curso:</b>
<b>1</b>	<b>Cognitivo:</b> Comprende los principios y beneficios de las metodologías ágiles.
	<b>Procedimental:</b> Analiza diferentes metodologías ágiles y cómo se aplican en proyectos de software.
	<b>Actitudinal:</b> Valora la importancia de la agilidad en el desarrollo de software y su impacto en el trabajo colaborativo.
<b>2</b>	<b>Cognitivo:</b> Conoce la historia, los principios fundamentales y los beneficios de Scrum como metodología ágil.
	<b>Procedimental:</b> Identifica y describe los roles y ceremonias clave dentro de Scrum.
	<b>Actitudinal:</b> Fomenta una actitud abierta hacia el trabajo ágil y la flexibilidad en los proyectos de software.
<b>3</b>	<b>Cognitivo:</b> Reconoce los componentes principales de Scrum, tales como roles, artefactos y eventos.
	<b>Procedimental:</b> Aplica los artefactos y eventos de Scrum en la gestión de proyectos de software.
	<b>Actitudinal:</b> Desarrolla una actitud proactiva hacia la colaboración y la responsabilidad dentro de un equipo Scrum.
<b>4</b>	<b>Cognitivo:</b> Comprende cómo se estructura el trabajo en Scrum, incluidos los sprints y las tareas de estos.
	<b>Procedimental:</b> Planifica y ejecuta un sprint dentro de un proyecto de software utilizando Scrum.
	<b>Actitudinal:</b> Adopta una actitud de mejora continua y adaptabilidad a los cambios durante el desarrollo de un proyecto con Scrum.

## Competencias Genéricas

**Tabla 2**  
Competencias

<b>Competencia Genérica</b>	<b>Descripción</b>
<b>Trabajo en equipo</b>	Capacidad para colaborar eficazmente en grupos multidisciplinarios, promoviendo la cooperación y el respeto mutuo.
<b>Resolución de problemas</b>	Habilidad para identificar y proponer soluciones efectivas ante desafíos en proyectos de software, aplicando enfoques ágiles.
<b>Comunicación efectiva</b>	Capacidad para expresar ideas de manera clara y precisa, promoviendo una comunicación abierta dentro del equipo y con los stakeholders.
<b>Adaptabilidad y flexibilidad</b>	Capacidad para ajustarse a cambios y nuevas circunstancias, manteniendo un enfoque ágil en el desarrollo de software.

### Perfil de Egreso del futuro profesional

Al finalizar la carrera, el graduado será un profesional competente en el desarrollo de software, con una sólida comprensión de las metodologías ágiles, especialmente Scrum, estará en capacidad de gestionar proyectos de software de manera ágil, liderando equipos de trabajo multidisciplinarios que garanticen la entrega continua de valor al cliente, además, dispondrá de habilidades en la planificación, ejecución y evaluación de sprints, así como la posibilidad de adaptarse a procesos cambiantes del entorno.

Podrá identificar y resolver problemas complejos en el desarrollo de software, optimizando recursos y tiempos de entrega, mientras mantiene una comunicación efectiva y constante con todos los miembros del equipo, el profesional estará preparado para enfrentar la dinámica de proyectos tecnológicos, gestionando tanto los aspectos técnicos como humanos de los mismos, con un enfoque hacia la mejora continua y la innovación

# CAPÍTULO II

Guía didáctica



## Unidad 1 - Enfoques ágiles

El enfoque ágil surgió cuando muchos desarrolladores de software notaron que los ciclos de producción y los métodos de colaboración del modelo en cascada no daban los resultados esperados. Para comienzos de la década de los 90, este problema se había generalizado, ya que era común que transcurrieran varios años entre la aparición de una necesidad empresarial justificada y la distribución de una aplicación en funcionamiento. Eran tantos los cambios que podían sufrir las demandas empresariales y los mercados durante esos años, que posiblemente se cancelarían partes importantes de los proyectos de software antes de poder distribuirse.

El desperdicio de tiempo y recursos llevó a que varios desarrolladores de software buscaran una alternativa. Las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo pequeñas piezas de software en funcionamiento para aumentar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un enfoque sencillo de la documentación de software y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos [1].

Al dividir los proyectos complejos en tareas más pequeñas llamadas historias de usuario o características, los equipos pueden centrarse en ofrecer valor de forma temprana y frecuente. Esta metodología fomenta la comunicación regular entre las partes interesadas y los miembros del equipo, fomentando la transparencia y reduciendo los riesgos asociados a expectativas desalineadas. Además, la importancia de la metodología ágil radica en su capacidad para fomentar la innovación y gestionar la incertidumbre de forma eficiente. Permite a los equipos responder con prontitud a los cambios del mercado o a las oportunidades emergentes adaptando rápidamente sus procesos de desarrollo [2].

## Objetivos de aprendizaje

- Entregar resultados de forma ágil, frecuente y eficiente, simplificando y adaptando los procesos de forma dinámica, flexible y considerando las necesidades que se presentan por parte del cliente [3].
- Promover un ambiente donde los equipos trabajan hacia la consecución de un objetivo común, en un entorno de colaboración estrecha y comunicación frecuente [4].
- Permitir adaptarse al cambio, de acuerdo con las necesidades cambiantes del cliente [5].
- Mejorar la calidad de software, A través de prácticas como la integración continua, pruebas automatizadas y revisiones frecuentes [6].
- Promover la autoorganización y empoderamiento de los equipos, permitiendo a los equipos ágiles se organicen a sí mismos, fomentando la responsabilidad y la creatividad entre los miembros [7].

## Contenidos

### a. Contextos tradicionales y contextos ágiles

Los contextos tradicionales de desarrollo de software se caracterizan por su enfoque secuencial y estructurado, dividiendo el proyecto de desarrollo en fases discretas, como: requisitos, diseño, implementación, verificación y mantenimiento. Cada fase debe completarse antes de avanzar, lo que facilita la planificación, minimiza cambios y mantiene un proceso predecible.

#### **Las metrologías tradicionales se pueden mencionar:**

- **Modelo en cascada (Waterfall):** se trata de un enfoque en el que cada etapa del desarrollo de software (requisitos, diseño, implementación, pruebas y mantenimiento) se completa antes de pasar a la siguiente. Es rígido y no permite cambios, una vez que se ha avanzado a la siguiente fase.
- **Modelo en espiral:** combina elementos del modelo en cascada y de la metodología ágil, permitiendo la repetición del ciclo de desarrollo de software, para abordar los riesgos y las incertidumbres de manera iterativa.

- **Diseño rápido de aplicaciones (RAD):** El RAD es una metodología que se enfoca en la entrega rápida y la participación del cliente. Se apoya en el uso de técnicas como la modelización, la automatización y la reutilización, para acelerar el desarrollo del software.
- **Modelo de desarrollo en prototipos:** se centra en la construcción rápida de prototipos, para obtener retroalimentación temprana y así redefinir o mejorar la solución final.
- **Desarrollo incremental:** esta metodología permite que el software se desarrolle y entregue en partes incrementales y funcionales. Cada interacción agrega capacidades adicionales al producto, lo que permite lanzamientos tempranos y frecuentes [9].

El contexto ágil en desarrollo de software no solo es un conjunto de prácticas metodológicas, sino también un enfoque cultural que fomenta la colaboración, la adaptabilidad y la entrega continua de valor. En entornos empresariales de gran escala, su flexibilidad se convierte en un activo estratégico. Los equipos ágiles se centran en la interacción constante con los interesados, lo que permite ajustes rápidos a medida que se obtienen comentarios tempranos.

Este enfoque iterativo y de entrega incremental permite que las grandes empresas aborden la complejidad de los proyectos al dividirlos en módulos más manejables. Esto proporciona la capacidad de priorizar y abordar las áreas críticas primero, mitigando así los riesgos y permitiendo una adaptación más fluida a los cambios en los requisitos o las expectativas del cliente. Además, la transparencia y la comunicación frecuente en el desarrollo ágil fomentan una mayor confianza y alineación entre los equipos, lo que puede ser crucial en entornos donde la coordinación entre múltiples departamentos o unidades de negocio es esencial para el éxito del proyecto [10].

### **Las metodologías ágiles se pueden mencionar:**

- **Agile:** Agile impulsa un desarrollo dinámico y adaptativo, centrado en la colaboración y la capacidad de responder rápidamente a cambios. Utiliza ciclos cortos llamados sprints que evalúan y adaptan el progreso del proyecto en intervalos regulares, de hecho, según echometer, las empresas que adoptaron esta metodología obtuvieron los mejores resultados.

- **Scrum:** La metodología Scrum es una implementación específica de Agile que estructura el desarrollo en ciclos cortos y regulares (sprints) con roles específicos como Scrum Master y Product Owner, para optimizar la gestión y la eficiencia [8].
- **Waterfall (Cascada):** Es un enfoque en el desarrollo de software que se caracteriza por seguir un proceso lineal y secuencial. Waterfall divide el proyecto en distintas fases claramente definidas y ordenadas, como concepción, inicio, análisis, diseño, construcción, pruebas, y mantenimiento; las cuales deben completarse una tras otra [8].
- **Lean:** Es una metodología que se basa en identificar y eliminar cualquier actividad que no aporte valor al producto final, lo cual se conoce como “desperdicio”, esto pueden ser tareas duplicadas o cualquier actividad que no aporte directamente al cumplimiento de las expectativas del cliente.
- **Rapid Application Development (RAD):** Se centra en la rápida creación y refinamiento de prototipos de software con un ciclo de retroalimentación constante de los usuarios, permitiendo a los desarrolladores ajustar y mejorar el software de manera continua en respuesta a las necesidades y preferencias del usuario.
- **Incremental:** La metodología Incremental divide el proyecto de software en múltiples segmentos o incrementos funcionales, cada uno de los cuales agrega nuevas funcionalidades que se integran con las partes ya desarrolladas.
- **Prototipo:** La metodología Prototipo se centra en la creación de versiones preliminares del software, conocidas como prototipos, que no son completamente funcionales, pero permiten visualizar cómo funcionará el producto final.
- **Espiral:** Es un enfoque que combina elementos del diseño iterativo y la metodología Waterfall para aprovechar los beneficios de ambos: el desarrollo iterativo para la flexibilidad y la adaptación, y la planificación sistemática y secuencial de Waterfall para la estructura y la medición del progreso [8].

**Tabla 3**  
Contextos tradicionales vs ágiles

COMPARATIVA		
Característica	Contextos Tradicionales	Contextos Ágiles
Planificación	Contextos Tradicionales	Contextos Ágiles
Documentación	Extensa y formal	Ligera y suficiente
Cambios en requisitos	Difíciles de incorporar	Fáciles de adaptar
Entrega de valor	Al final del proyecto	Incremental y continua
Comunicación	Basada en documentos	Directa y colaborativa
Enfoque de equipo	Jerárquico y especializado	Autoorganizado y multifuncional

## b. El Manifiesto ágil

El Manifiesto Ágil es un documento que se centra en los 4 valores y 12 principios del desarrollo de software con metodologías ágiles. Lo publicaron, en febrero de 2001, 17 desarrolladores de software que necesitaban contar con una alternativa al proceso de desarrollo de productos más lineal y orientado a procesos [11].

### a.1. Los 4 valores de las metodologías ágiles

**Figura 1**  
Individuos e interacciones sobre procesos y herramientas



Valorar más a los individuos y sus interacciones que a los procesos y las herramientas. Los equipos autoorganizados que trabajan con metodologías ágiles valoran más la colaboración en equipo y trabajar juntos, que trabajar de manera independiente y hacer las cosas “al pie de la letra”. Esta regla trata de impulsar la comunicación interna entre las personas del equipo para que estos colaboren unos con otros y, de esta forma, se obtenga el mejor resultado posible.

**Figura 2**  
Software funcionando sobre documentación extensiva



Valorar más el software en funcionamiento que la documentación exhaustiva. El software que desarrollan los equipos ágiles debe funcionar. El trabajo extra, como la documentación, no es tan importante como desarrollar un buen software. Lo que prima en el Manifiesto ágil es entregar al consumidor el producto que quieren por encima de otros trabajos que no se consideran imprescindibles.

**Figura 3**  
Colaboración con el cliente sobre negociación contractual.



Valorar más la colaboración entre las partes interesadas y el cliente que la negociación contractual. La gestión de clientes es sumamente importante para las metodologías Agile. Los equipos que trabajan con estas metodologías dejan que los clientes marquen la dirección en la que se debe orientar el software. Por lo tanto, la colaboración con el cliente es más importante que los detalles más específicos de la negociación contractual.

**Figura 4**  
Respuesta ante el cambio sobre seguir un plan.



Valorar más la respuesta ante el cambio que seguir un plan. Uno de los principales beneficios de las metodologías ágiles de proyectos es que permiten que los equipos sean flexibles. Esta estructura ofrece la posibilidad de que los equipos cambien rápido de estrategia sin afectar al proyecto entero. En el manifiesto ágil el contrato con el cliente debe ser flexible ya que lo más importante es entregar un producto que aporte valor al cliente [11].

#### **a.2. Los 12 principios ágiles del manifiesto ágil**

Los 4 valores son los pilares de la metodología ágil. A partir de esos valores, se desarrolló 12 principios. Estos principios se pueden adaptar fácilmente para adecuarlos a las necesidades de cada equipo y son:

- Satisfacer a los clientes a través de la entrega temprana y continua. Cuando los clientes reciben actualizaciones con regularidad, es más probable que vean los cambios que desean dentro del producto y la satisfacción del cliente sea positiva. La consecuencia es que los clientes son más felices y están más satisfechos; además de que hay más recaudación recurrente. En el manifiesto ágil es clave deleitar al cliente.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del proyecto. El eje fundamental de los esquemas ágiles es la adaptabilidad. En procesos iterativos como los ágiles, la flexibilidad causa más bien que mal.

- Hacemos entregas valiosas con frecuencia. Este principio es similar al primero, efectuamos entregas valiosas a los clientes con frecuencia y por eso es menos probable que los perdamos.
- Rompemos con el aislamiento en los proyectos. La colaboración es la clave de la estrategia ágil. El objetivo es que las personas superen sus propios proyectos individuales y colaboren juntas con mayor frecuencia.
- Desarrollamos proyectos con personas motivadas. La metodología ágil funciona mejor cuando los integrantes de los equipos se sienten comprometidos con lo que hacen y trabajan activamente para alcanzar algún objetivo, lo que contribuye a la excelencia técnica del proyecto.
- El modo más eficiente de comunicar es con conversaciones cara a cara. Si bien es cierto que el trabajo ha cambiado radicalmente desde 2001 a la fecha, la percepción con respecto a esta declaración aún tiene vigencia. Si trabajas en un equipo descentralizado, dedica tiempo a comunicarte de forma más personalizada, fomentando la comunicación cara a cara, como las videollamadas por Zoom.
- El software en funcionamiento es el principal indicador del progreso. Lo más importante es que los equipos se esfuercen (con la aplicación de estos esquemas de trabajo ágiles) por desarrollar los productos. En el Manifiesto ágil se establece un objetivo prioritario: la entrega de software que funciona por sobre cualquier otra cosa.
- Mantenemos un ritmo de trabajo constante. Algunos aspectos de las metodologías ágiles pueden tener un ritmo acelerado, pero nada debería ser tan rápido como para que los miembros del equipo terminen padeciendo agotamiento. El objetivo es mantener la constancia a lo largo del proyecto.

- La excelencia continua favorece la agilidad. Si el equipo desarrolla un código excelente en un sprint, podrá basarse en ese mismo código para el siguiente desarrollo sostenible. La generación continua de trabajos de excelencia permite que los equipos avancen más rápido a medida que transcurre el tiempo.
- La simplicidad es esencial. A veces, la solución más simple es la mejor. Las metodologías ágiles apuntan a no complicar demasiado las cosas y a hallar respuestas simples para problemas complejos.
- La organización interna y autónoma de los equipos genera resultados más valiosos. En cierto modo, es similar al quinto principio. Los equipos en los que se trabaja con proactividad producen activos más valiosos para la empresa, ya que se esfuerzan por aportar ese valor.
- Con regularidad, el equipo reflexiona y adapta las formas de trabajo para favorecer la efectividad. Las reuniones diarias para análisis retrospectivo son muy comunes en las prácticas ágiles. Es tiempo dedicado a que los equipos miren hacia atrás, reflexionen sobre su desempeño y adapten los comportamientos para el futuro [11].

## **Actividades**

### **Actividad 1: Sopa de Letras**

Resolver la siguiente sopa de letras.

**Figura 5**  
 Juego: SOPA DE LETRAS

R	G	R	T	G	R	E	T	A	N	P	L	C	R	D	U	A	R	R
M	S	A	O	R	E	A	C	N	C	R	S	D	I	A	R	S	D	E
O	A	D	O	L	V	D	O	D	R	O	T	D	D	E	E	G	R	A
D	D	A	F	D	P	I	M	E	V	T	I	S	L	L	A	A	D	F
E	E	U	I	D	C	A	R	L	S	O	D	P	I	O	W	I	E	S
L	O	O	S	U	A	U	S	I	E	T	C	G	A	T	T	N	L	E
O	C	S	L	A	C	D	L	O	T	I	A	E	F	O	O	S	S	R
D	E	O	R	E	N	S	T	C	S	P	H	O	S	A	O	E	K	C
E	S	R	E	S	U	L	T	A	D	O	S	R	A	P	I	D	O	S
E	T	E	M	L	S	A	T	O	S	C	A	E	A	E	O	N	D	P
S	S	S	I	A	S	P	R	D	O	O	O	P	T	S	D	A	S	E
P	P	N	A	R	N	P	R	R	E	F	E	A	P	E	N	O	O	S
I	A	J	M	O	D	E	L	O	D	E	C	A	S	C	A	D	A	A
R	T	R	C	A	A	A	P	R	Y	R	L	Z	E	T	A	P	A	D
A	T	A	P	R	B	L	A	O	P	E	E	O	E	F	T	O	U	A
L	E	F	S	S	C	A	A	C	A	S	C	O	L	R	O	E	D	S
O	R	P	S	T	S	D	D	F	T	E	P	T	O	A	T	Y	I	D
S	N	P	R	R	A	A	E	A	N	T	E	D	O	A	F	D	I	B
N	N	R	A	I	N	A	C	O	U	E	M	Q	N	P	C	T	I	O

**Palabras a encontrar:**

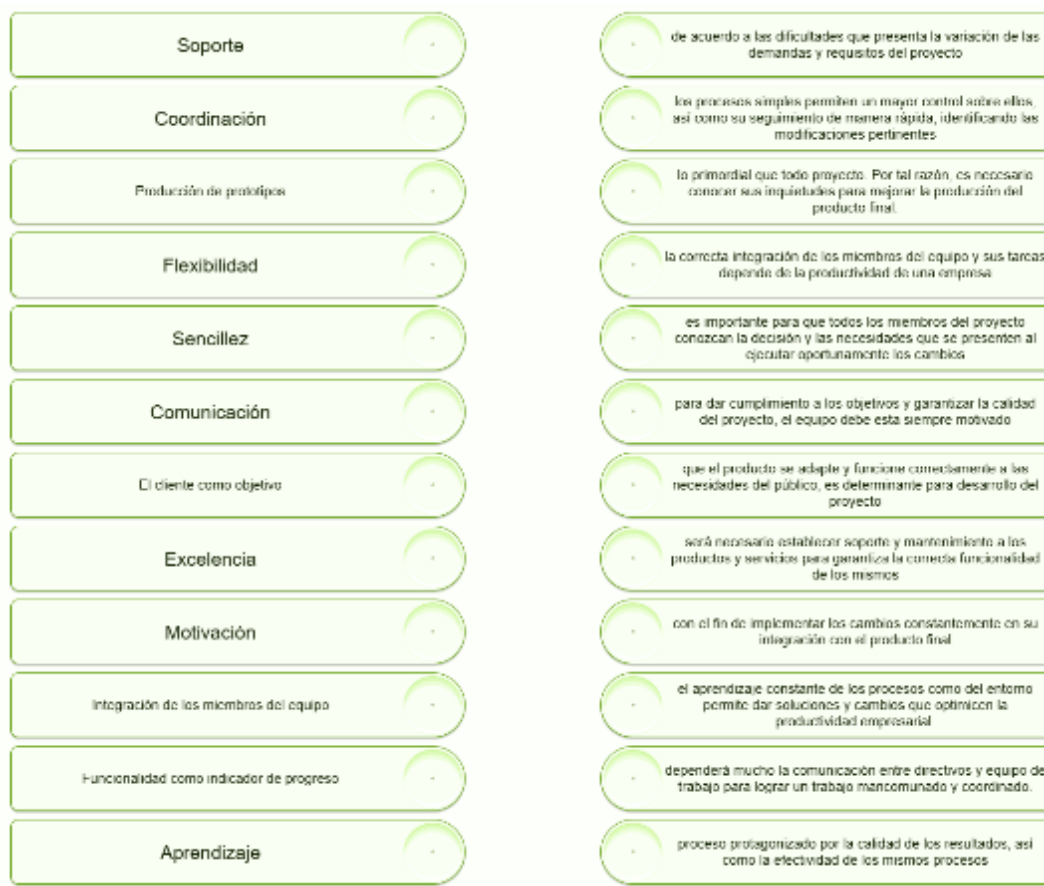
---

<b>AGILES</b>	<b>PROTOTIPO</b>	<b>PASOS</b>
<b>PESADAS</b>	<b>ETAPA</b>	<b>RESULTADOS RAPIDOS</b>
<b>MODELO DE CASCADA</b>	<b>PROYECTO</b>	<b>RAD</b>
<b>MODELO DE ESPIRAL</b>	<b>SOLUCION</b>	<b>SOFTWARE</b>

**Actividad 2: Una lo correcto**

Una la palabra con su respectiva definición de acuerdo con lo aprendido.

**Figura 6**  
Actividad 2



## Estrategias de enseñanza

### Actividad en Clase: Makerspace

Esta estrategia es de 4 horas, los estudiantes se sumergirán en el mundo de las metodologías ágiles a través de un enfoque teórico-práctico, utilizando el aprendizaje basado en proyectos. El objetivo principal es que los participantes no solo comprendan los conceptos teóricos detrás de las metodologías ágiles, sino que también los apliquen en un entorno simulado, desarrollando habilidades clave como el trabajo en equipo, la adaptación al cambio y la mejora continua.

**Duración:** 4 horas

**Modalidad:** Teórico-práctica (aprendizaje basado en proyectos)

**Objetivo General:** Que los estudiantes comprendan y apliquen las metodologías ágiles en la gestión de proyectos, desarrollando habilidades de trabajo en equipo, adaptación al cambio y mejora continua.

**Pasos:**

- Lluvia de Ideas Inicial (30 minutos):
- Se inicia con una pregunta abierta: "¿Qué entienden por 'ágil' en el contexto de proyectos?"
- Los estudiantes comparten sus ideas y percepciones, generando un diálogo inicial que permite identificar conocimientos previos y expectativas.
- Explicación Magistral (45 minutos):
- Introducción al Manifiesto Ágil, sus 4 valores y 12 principios.
- Breve descripción de metodologías como Scrum y Kanban.
- Ejemplos prácticos de cómo las metodologías ágiles se aplican en diferentes industrias.

**Dinámica Práctica:** "Construcción de un Avión de Papel" (1 hora 30 minutos):

**1. Enfoque Tradicional:**

Los equipos reciben instrucciones detalladas y rígidas para construir un avión de papel.

Se limita la comunicación y no se permite hacer cambios durante el proceso.

**2. Enfoque Ágil:**

Los equipos reciben retroalimentación después de la primera iteración.

Se fomenta la colaboración, la adaptación y la mejora continua.

Los equipos pueden ajustar su diseño basándose en los resultados y comentarios.

**Reflexión Grupal:**

Se comparan ambas iteraciones, destacando las diferencias en el proceso y los resultados.

Se analizan las ventajas del enfoque ágil: flexibilidad, colaboración y mejora continua.

Cierre (45 minutos):

Preguntas y discusión sobre cómo aplicar los principios ágiles en proyectos reales.

Los estudiantes comparten ideas sobre cómo podrían implementar estas metodologías en su vida académica o profesional.

El docente resume los conceptos clave y refuerza la importancia de la mentalidad ágil.

### **Resultados Esperados:**

Los estudiantes comprenderán los fundamentos de las metodologías ágiles y su aplicación práctica.

Habrán experimentado de manera vivencial las diferencias entre un enfoque tradicional y uno ágil.

Desarrollarán habilidades clave como trabajo en equipo, adaptabilidad y mejora continua.

### **Actividad en Clase: Simulación**

Esta sesión está diseñada para proporcionar a los estudiantes una comprensión profunda y práctica de las metodologías ágiles, con un enfoque en su aplicación en proyectos simulados o reales. A lo largo de 6 horas (divididas en dos bloques de 3 horas), los participantes explorarán los principios del Manifiesto Ágil, aprenderán sobre metodologías como Scrum, Kanban y XP, y aplicarán estos conocimientos en un proyecto práctico. El objetivo es desarrollar habilidades clave como el trabajo en equipo, la comunicación efectiva y la adaptabilidad, fomentando una mentalidad ágil para resolver problemas de manera iterativa e incremental.

**Duración:** 6 horas (divididas en dos bloques de 3 horas cada uno)

**Modalidad:** Teórico-práctica (aprendizaje basado en proyectos)

**Objetivo General:** Aplicar herramientas y técnicas ágiles en proyectos simulados o reales, desarrollando habilidades de trabajo en equipo, comunicación y adaptabilidad, mientras se fomenta una mentalidad ágil para resolver problemas de manera iterativa e incremental.

### **Pasos 1 (3 horas):**

- **Presentación de los principios del Manifiesto Ágil:**
  - Introducción a los 4 valores y 12 principios del Manifiesto Ágil
  - Explicación de metodologías ágiles (Scrum, Kanban, XP):
  - Descripción de cada metodología, sus roles, herramientas y procesos clave
- **Discusión de casos de éxito y fracaso:**
  - Análisis de ejemplos reales para entender los beneficios y desafíos de implementar metodologías ágiles.
- **Recursos:**
  - Videos explicativos, lecturas y dinámicas grupales para identificar problemas tradicionales que las metodologías ágiles buscan resolver.

### **Pasos 1 (3 horas):**

- **Formación de equipos de trabajo:**
  - Creación de equipos de 4-6 personas para trabajar en un proyecto simulado.
- **Asignación de un proyecto simulado:**
  - Ejemplos: desarrollar una aplicación, organizar un evento o mejorar un proceso.
- **Implementación de Scrum:**
  - Organización de sprints, definición de roles (Scrum Master, Product Owner, equipo de desarrollo) y uso de tableros Kanban.
  - Realización de reuniones diarias (stand-ups), revisiones de sprint y retrospectivas.
- **Recursos:**
  - Herramientas digitales como Trello o Jira para gestionar tareas.
  - Plantillas para definir user stories, backlog y criterios de aceptación.

### **Resultados Esperados:**

Comprensión profunda de los principios y metodologías ágiles.

Experiencia práctica en la aplicación de Scrum y Kanban en un proyecto simulado.

Desarrollo de habilidades clave como colaboración, comunicación y adaptabilidad.

Mentalidad ágil para resolver problemas de manera iterativa e incremental.

## **Evaluación**

### **Evaluación: Retroalimentación**

Esta evaluación está diseñada para medir la comprensión inicial de los estudiantes sobre los conceptos básicos de las metodologías ágiles. Incluye preguntas teóricas y prácticas que abarcan los principios del Manifiesto Ágil, las metodologías más comunes (Scrum, Kanban) y su aplicación en proyectos. La evaluación es ideal para un curso introductorio o como diagnóstico inicial.

**Duración:** 30 minutos.

**Tipo:** Prueba escrita o en línea.

**Formato:** Preguntas de selección múltiple, verdadero/falso y desarrollo breve.

- **Selección Múltiple (2 puntos cada una)**

1. ¿Cuál de los siguientes es uno de los valores del Manifiesto Ágil?

- a) Seguir un plan detallado sobre responder al cambio.
- b) Documentación exhaustiva sobre software funcionando.
- c) Negociación de contratos sobre colaboración con el cliente.
- d) Procesos y herramientas sobre individuos e interacciones.

Respuesta correcta: a) Seguir un plan detallado sobre responder al cambio.

2. ¿Qué metodología ágil utiliza sprints de tiempo fijo?

- a) Kanban.
- b) Scrum.
- c) XP (Extreme Programming).
- d) Lean.

Respuesta correcta: b) Scrum.

3. ¿En Kanban, ¿qué significa WIP (Work in Progress)?

- a) El número máximo de tareas que pueden estar en progreso al mismo tiempo.
- b) La lista de tareas completadas.
- c) El tiempo que toma completar un sprint.
- d) La reunión diaria del equipo.

Respuesta correcta: a) El número máximo de tareas que pueden estar en progreso al mismo tiempo.

4. ¿Cuál es el rol del Scrum Master en un equipo ágil?

- a) Definir los requisitos del producto.
- b) Facilitar el proceso Scrum y eliminar obstáculos.
- c) Desarrollar el código del producto.
- d) Gestionar el presupuesto del proyecto.

Respuesta correcta: b) Facilitar el proceso Scrum y eliminar obstáculos.

- **Verdadero/Falso (1 punto cada una)**

El Manifiesto Ágil prioriza la colaboración con el cliente sobre la negociación de contratos.

Respuesta: Verdadero.

En Scrum, el Product Owner es responsable de gestionar el backlog del producto.

Respuesta: Verdadero.

Kanban no utiliza sprints y permite un flujo continuo de trabajo.

Respuesta: Verdadero.

Las metodologías ágiles son útiles solo para proyectos de desarrollo de software.

Respuesta: Falso.

- **Desarrollo Breve (4 puntos cada una)**

**Explica brevemente qué es el Manifiesto Ágil y menciona uno de sus valores.**

Ejemplo de respuesta:

El Manifiesto Ágil es un conjunto de principios y valores que guían el desarrollo ágil de software. Uno de sus valores es "Individuos e interacciones sobre procesos y herramientas", lo que significa que se prioriza la comunicación y colaboración entre las personas sobre el uso de herramientas o procesos rígidos.

## **Describe la diferencia entre Scrum y Kanban en una oración.**

Ejemplo de respuesta:

Scrum utiliza sprints de tiempo fijo y roles definidos, mientras que Kanban se enfoca en un flujo continuo de trabajo con límites en el trabajo en progreso (WIP).

## **Evaluación de aprendizaje**

Esta evaluación está diseñada para medir la comprensión de los estudiantes sobre los conceptos fundamentales de las metodologías ágiles. Incluye preguntas teóricas y prácticas que abarcan los principios del Manifiesto Ágil, las metodologías más comunes (Scrum, Kanban, XP) y su aplicación en proyectos. La evaluación puede ser utilizada como un examen escrito, una actividad grupal o una autoevaluación.

**Duración:** 45 minutos.

**Tipo:** Prueba escrita o en línea.

**Formato:** Preguntas de selección múltiple, verdadero/falso y desarrollo breve.

### **• Selección Múltiple (2 puntos cada una)**

1. ¿Cuál de los siguientes es uno de los valores del Manifiesto Ágil?

- a) Seguir un plan detallado sobre responder al cambio.
- b) Documentación exhaustiva sobre software funcionando.
- c) Negociación de contratos sobre colaboración con el cliente.
- d) Procesos y herramientas sobre individuos e interacciones.

Respuesta correcta: a) Seguir un plan detallado sobre responder al cambio.

2. ¿En Scrum, ¿cuál es la duración máxima recomendada para un sprint?

- a) 1 semana.
- b) 2 semanas.
- c) 4 semanas.
- d) No hay duración máxima.

Respuesta correcta: c) 4 semanas.

3. ¿Cuál de las siguientes es una herramienta visual utilizada en Kanban?

- a) Tablero de tareas.
- b) Gráfico de Gantt.

- c) Diagrama de flujo.
- d) Matriz de priorización.

Respuesta correcta: a) Tablero de tareas.

4. ¿Qué rol en Scrum es responsable de maximizar el valor del producto?

- a) Scrum Master.
- b) Product Owner.
- c) Equipo de desarrollo.
- d) Stakeholder.

Respuesta correcta: b) Product Owner.

- **Verdadero/Falso (1 punto cada una)**

El Manifiesto Ágil prioriza la documentación exhaustiva sobre el software funcionando.

Respuesta: Falso.

En Kanban, el trabajo en progreso (WIP) debe limitarse para mejorar la eficiencia.

Respuesta: Verdadero.

Las reuniones diarias (stand-ups) en Scrum deben durar máximo 15 minutos.

Respuesta: Verdadero.

Extreme Programming (XP) se enfoca en la entrega rápida de software sin considerar la calidad.

Respuesta: Falso.

- **Desarrollo Breve (4 puntos cada una)**

**Explique dos principios del Manifiesto Ágil y cómo se aplican en un proyecto.**

Ejemplo de respuesta:

"Responder al cambio sobre seguir un plan": En un proyecto ágil, los equipos están preparados para adaptarse a cambios en los requisitos, incluso en etapas avanzadas del desarrollo.

"Individuos e interacciones sobre procesos y herramientas": Se prioriza la comunicación y colaboración entre los miembros del equipo sobre el uso de herramientas o procesos rígidos.

## **Describe las diferencias entre Scrum y Kanban.**

Ejemplo de respuesta:

**Scrum:** Se trabaja en sprints de tiempo fijo (2-4 semanas), con roles definidos (Scrum Master, Product Owner, equipo de desarrollo) y ceremonias como sprint planning, daily stand-ups y retrospectivas.

**Kanban:** No tiene sprints ni roles específicos; se enfoca en visualizar el flujo de trabajo en un tablero y limitar el trabajo en progreso (WIP) para mejorar la eficiencia.

## **UNIDAD 2 – Introducción a SCRUM**

### **Objetivos de aprendizaje**

Al final de esta sección, el lector será capaz de entender y explicar los principios fundamentales de SCRUM, incluyendo su historia, los elementos, los pilares y los valores. Además, podrá aplicar estos conceptos en un contexto práctico, demostrando cómo cada uno de estos componentes contribuye al éxito de un equipo de trabajo ágil y a la entrega continua de valor en proyectos de desarrollo de software.

### **Contenidos**

#### **Historia**

El término 'SCRUM' no fue elegido al azar. La palabra proviene del mundo del rugby, donde se utiliza para describir una jugada estratégica en la que los jugadores se alinean estrechamente, formando un bloque compacto que empuja con fuerza y coordinación para avanzar. Este tipo de esfuerzo colaborativo resonó profundamente con los principios que Jeff Sutherland y Ken Schwaber querían capturar en su marco de trabajo.

Así como en el rugby donde el éxito de la jugada depende de la sincronización y el esfuerzo conjunto de todos los jugadores, SCRUM en el mundo de la gestión de proyectos hace énfasis en la importancia de los equipos trabajando juntos de manera cohesiva. No es un esfuerzo individual sino colectivo, donde cada miembro tiene un papel crucial que desempeñar y el equipo avanza como una unidad hacia el objetivo.

El nombre no solo es un guiño a los orígenes del concepto, sino que también refleja los valores de SCRUM: la importancia de la colaboración, la flexibilidad, y el enfoque en los resultados a través del trabajo en equipo.

En la década de los 90, el mundo de la gestión de proyectos estaba marcado por enfoques rígidos y pesados. Los proyectos de software en particular enfrentaban enormes desafíos al seguir métodos tradicionales de planificación y ejecución, que no siempre lograban adaptarse a los cambios rápidos y las necesidades emergentes de los usuarios. Fue en este contexto donde surgió SCRUM, como una alternativa innovadora, diseñada para brindar flexibilidad, transparencia y eficiencia.

El marco SCRUM fue conceptualizado por Jeff Sutherland y Ken Schwaber a principios de los años 90, quienes reconocieron la necesidad de un enfoque ágil para enfrentar los problemas del desarrollo de software. Su inspiración provino de diversas fuentes, incluyendo el concepto de "empoderar a los equipos" y los principios de trabajo colaborativo que se utilizaban en la gestión de proyectos en Japón, como el modelo de lean y las ideas de productividad de equipos en el mundo del deporte [12].

## **Influencia de Nonaka y Takeuchi en el desarrollo de SCRUM**

Uno de los mayores referentes en la creación de SCRUM es el trabajo de Ikujiro Nonaka y Hirotaka Takeuchi, dos académicos japoneses cuyas ideas sobre la gestión del conocimiento y la creación de productos influyeron profundamente en Jeff Sutherland y Ken Schwaber.

En 1986, Nonaka y Takeuchi publicaron su artículo *The New New Product Development Game*, en el cual describieron cómo las empresas japonesas estaban adoptando un enfoque de trabajo más flexible y colaborativo, alejándose de las estructuras jerárquicas tradicionales. Según su investigación, los equipos de desarrollo más exitosos eran aquellos que se organizaban de forma autónoma, en ciclos rápidos de retroalimentación, y que trabajaban con un enfoque en el aprendizaje constante.

Los equipos, en lugar de ser solo un conjunto de personas con habilidades específicas, se convertían en unidades completamente integradas, con miembros de diferentes disciplinas que colaboraban de forma continua. Esta idea de

equipos multidisciplinarios que trabajaban de manera ágil y flexible es, sin duda, una de las piedras angulares de SCRUM [13].

Al igual que en SCRUM, los equipos que describen Nonaka y Takeuchi no solo eran responsables de sus tareas individuales, sino que tomaban decisiones colectivas para avanzar en el proyecto. La iteración y la constante mejora del proceso eran esenciales para el éxito. En este sentido, SCRUM toma muchas de estas ideas clave: la autogestión, la colaboración continua, y el enfoque en los resultados a través de ciclos repetitivos de trabajo. SCRUM no es solo una metodología para la gestión de proyectos, sino una filosofía de trabajo que encarna muchas de las enseñanzas de estos dos pioneros en el ámbito de la innovación empresarial.

La primera implementación de SCRUM se documentó formalmente en 1995 y a partir de ahí el marco creció en popularidad, influenciado por la creación del Manifiesto Ágil en 2001, que marcó un hito importante en la forma en que las organizaciones comenzaron a pensar sobre el desarrollo de software. Lo que comenzó como un modelo dirigido al desarrollo de software rápidamente se expandió a otros campos, demostrando su versatilidad y aplicabilidad en cualquier tipo de proyecto que requiera adaptabilidad y trabajo en equipo.

## **Definición**

SCRUM es un marco de trabajo ágil que permite a los equipos abordar proyectos complejos de manera flexible y eficiente. A diferencia de las metodologías tradicionales, SCRUM no dicta cómo se debe hacer cada tarea, sino que proporciona un conjunto de roles, eventos y artefactos que guían al equipo en la organización y ejecución del trabajo [14].

El marco se basa en ciclos de trabajo cortos llamados Sprints, que permiten entregar incrementos funcionales del producto de forma continua y mejorar la calidad del trabajo de manera iterativa. SCRUM fomenta un enfoque colaborativo entre los miembros del equipo, el Product Owner y otras partes interesadas, lo que garantiza que se mantenga un alineamiento constante con los objetivos del proyecto [14].

Uno de los aspectos más importantes de SCRUM es su énfasis en la transparencia, la inspección y la adaptación. Estos principios permiten que los

equipos puedan ajustarse rápidamente a los cambios, identificar problemas de forma temprana y mejorar continuamente sus procesos.

## **Elementos**

Los elementos fundamentales de SCRUM consisten en los roles, los eventos y los artefactos. Estos son los componentes clave que definen cómo funciona SCRUM y cómo los equipos organizan su trabajo para alcanzar sus objetivos.

El Product Owner, el Scrum Master y el Equipo de Desarrollo son los tres roles esenciales en SCRUM. Cada uno tiene responsabilidades claramente definidas que contribuyen al éxito del proyecto. Los eventos (Sprint, Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective) proporcionan una estructura que guía las actividades del equipo, asegurando que se mantenga el enfoque y la calidad.

Finalmente, los artefactos, como el Product Backlog y el Sprint Backlog, proporcionan la transparencia y el control necesario para gestionar el trabajo de manera efectiva [14].

## **Fundamentos, Pilares y valores**

### ***Fundamentos***

Los fundamentos de SCRUM se basan en un enfoque ágil, centrado en el empoderamiento del equipo, la entrega continua de valor y la adaptación a los cambios. SCRUM se apoya en la creencia de que los proyectos deben ser tratados de manera flexible, donde los equipos puedan responder a nuevas oportunidades y cambios sin perder el foco en el objetivo final.

- **Enfoque en la entrega continua de valor:** En SCRUM, el trabajo se organiza en sprints, lo que permite que el equipo entregue productos funcionales de manera continua y rápida, lo que maximiza el valor entregado al cliente en cada ciclo.
- **Colaboración continua:** SCRUM fomenta la colaboración estrecha entre el equipo de desarrollo, el Product Owner y otras partes interesadas. A través de los eventos (como el Daily Scrum y la Sprint Review), los equipos tienen la

oportunidad de alinear constantemente el trabajo con las expectativas del cliente.

- **Adaptación:** SCRUM no solo se adapta a los cambios del proyecto, sino que también fomenta la adaptación de procesos. Con la Sprint Retrospective, los equipos pueden reflexionar sobre el trabajo realizado y encontrar áreas de mejora.

## Pilares

Los pilares de SCRUM son transparencia, inspección y adaptación. Estos principios aseguran que el proceso de trabajo sea efectivo y que el equipo pueda entregar valor continuamente [15].

- **Transparencia:** La transparencia en SCRUM significa que todos los aspectos del proceso deben ser visibles para los participantes. Desde el trabajo que se realiza en el Sprint hasta el estado del Product Backlog, todo debe ser accesible para el equipo, el Product Owner y los interesados. Esto permite que las decisiones se tomen con base en información clara y actualizada.
- **Inspección:** Los equipos SCRUM deben inspeccionar de manera regular y frecuente el progreso del trabajo. Las reuniones diarias, las Sprint Reviews y las retrospectivas son todas oportunidades para evaluar si se están alcanzando los objetivos y si el equipo se mantiene enfocado en la entrega de valor.
- **Adaptación:** Los equipos SCRUM deben ser capaces de adaptarse rápidamente cuando detecten que algo no está funcionando como se esperaba. Este pilar está estrechamente relacionado con la flexibilidad inherente a SCRUM, lo que permite que los equipos ajusten su enfoque o incluso sus prioridades para mejorar el proceso y cumplir con las expectativas del cliente.

## Valores

- Los valores de SCRUM son los principios que guían la forma en que los equipos deben trabajar juntos, colaborar y enfrentarse a los desafíos. Son cruciales para garantizar que SCRUM sea implementado de manera efectiva y con éxito [16].
- **Compromiso:** Los miembros del equipo SCRUM se comprometen a trabajar juntos para lograr el objetivo del Sprint y entregar valor continuamente. Esto significa que cada miembro debe estar comprometido con el éxito del equipo y el proyecto, y debe ser responsable de sus tareas.

- **Coraje:** El coraje es esencial para que los equipos puedan abordar desafíos, tomar decisiones difíciles y proponer ideas innovadoras. En SCRUM, se necesita coraje para probar nuevas formas de trabajo, aceptar el cambio y hablar abiertamente sobre los problemas.
- **Foco:** El equipo SCRUM debe estar enfocado en el trabajo del Sprint, y no debe distraerse con tareas que no están alineadas con el objetivo. El enfoque permite que el equipo sea más productivo y eficiente al trabajar en lo que realmente importa.
- **Apertura:** Los miembros del equipo SCRUM deben ser abiertos a nuevas ideas, retroalimentación y sugerencias. La transparencia y la comunicación abierta son fundamentales para que el equipo pueda mejorar y adaptarse.
- **Respeto:** El respeto mutuo entre los miembros del equipo es clave para el éxito de SCRUM. Esto implica valorar las habilidades y contribuciones de todos los miembros del equipo y trabajar juntos hacia un objetivo común.

## Actividades

### a. Actividad 1

#### Objetivo:

Fomentar el análisis crítico de cómo los elementos de SCRUM se aplican en situaciones reales, aplicando los fundamentos, pilares y valores de SCRUM a un caso práctico real, ayudando a los estudiantes a comprender cómo estos principios impulsan el éxito de un equipo ágil.

#### Instrucciones:

1. Revisar el caso de estudio 1 del anexo 1.
2. Analicen cómo el equipo en el caso los fundamentos, los pilares y los valores de SCRUM.
3. Dividirse en grupos pequeños y discutan lo siguiente:  
Fundamentos: ¿Cómo se está garantizando la entrega continua de valor? ¿Se están adaptando a los cambios y colaborando de manera efectiva?

- Pilares: ¿Cómo se están utilizando la transparencia, la inspección y la adaptación en el caso? ¿Qué ejemplos prácticos se pueden identificar en el proceso del equipo?
- Valores: ¿Cómo se reflejan los valores de SCRUM (compromiso, coraje, foco, apertura y respeto) en la forma en que el equipo trabaja?

Al final, cada grupo debe presentar sus conclusiones y ejemplos específicos sobre cómo el equipo en el caso está aplicando estos conceptos.

## **b. Actividad 2**

### **Simulación de un Daily Scrum (Reunión diaria)**

#### **Objetivo:**

Experimentar el funcionamiento de uno de los eventos más importantes de SCRUM, el Daily Scrum, poniendo énfasis en los valores y pilares de la metodología. dándoles a los estudiantes una experiencia práctica de cómo se lo lleva a cabo, para que interioricen la importancia de la transparencia, inspección, adaptación y los valores clave de SCRUM en la interacción diaria del equipo.

#### **Instrucciones:**

1. Organizarse en pequeños grupos de 3 a 5 personas.
2. Realizar una simulación de un Daily Scrum, donde cada miembro del equipo deberá compartir lo siguiente:
  - Lo que hizo ayer para avanzar con las tareas del Sprint.
  - Lo que hará hoy para seguir avanzando.
  - Obstáculos que podrían impedirles completar las tareas.
3. Tomar en cuenta lo siguiente:
  - Mantengan transparencia sobre su trabajo y compartan abiertamente sus progresos y dificultades.
  - Inspeccionen el avance de las tareas del Sprint asegurándose de que todo está alineado con el objetivo.
  - Se adapten a las posibles dificultades, sugiriendo soluciones o cambios si es necesario.

4. Responder a las siguientes preguntas: ¿Cómo cada miembro mostró compromiso al compartir avances? ¿Qué ejemplos de coraje surgieron cuando se hablaron de obstáculos?
5. Reflexione sobre cómo los participantes pueden aplicar los mismos principios en sus propios entornos de trabajo.

## **Estrategias de enseñanza**

### **Actividad en Clase: Aprendizaje basado en problemas (ABP)**

#### **Objetivo:**

Fomentar el pensamiento crítico y la resolución de problemas reales utilizando los principios de SCRUM.

#### **Descripción:**

El aprendizaje basado en problemas (ABP) se enfoca en presentar a los estudiantes un desafío real o simulado que deben resolver aplicando los conceptos que han aprendido. En el contexto de SCRUM, esta estrategia puede ayudar a los participantes a integrar los fundamentos, pilares y valores de SCRUM al enfrentarse a situaciones prácticas.

#### **Pasos de implementación:**

- 1. Plantear un problema:** Presenta a los participantes un caso de estudio que simule una situación donde un equipo esté implementando SCRUM en un proyecto de desarrollo de software, por ejemplo, un proyecto que está retrasado y necesita reorganizar su enfoque para cumplir con las expectativas del cliente.
- 2. Formar grupos pequeños:** Divide a los participantes en grupos pequeños y asígnales el desafío de identificar los problemas del proyecto, con un enfoque específico en cómo los principios de SCRUM pueden ayudar a resolverlos.
- 3. Discusión en grupo:**
  - **Fundamentos:** ¿Cómo puede el equipo entregar valor continuamente mientras se adapta a los cambios en las prioridades del proyecto?

- **Pilares:** ¿Qué aspectos del proceso necesitan ser más transparentes? ¿Cómo pueden los miembros del equipo inspeccionar su progreso de manera efectiva?
  - **Valores:** ¿Qué valores de SCRUM (compromiso, coraje, foco, apertura, respeto) pueden mejorar la dinámica del equipo y ayudar a superar obstáculos?
- 4. Presentación y discusión:** Los grupos presentan sus soluciones y justifican cómo los principios de SCRUM pueden resolver los problemas planteados. Después de las presentaciones, se realiza una discusión de grupo sobre las diferentes estrategias y enfoques, resaltando la aplicabilidad de los pilares y valores de SCRUM en cada propuesta.

## **Actividad en clase 2: Gamificación y simulaciones interactivas**

### **Objetivo:**

Aprender de manera práctica y dinámica sobre los fundamentos, pilares y valores de SCRUM a través de una experiencia interactiva y lúdica.

### **Descripción:**

La gamificación consiste en aplicar elementos y dinámicas de juego en el proceso de enseñanza para hacer el aprendizaje más atractivo y motivador. Utilizar simulaciones o juegos relacionados con SCRUM puede ser una manera divertida y efectiva de enseñar cómo los principios de SCRUM funcionan en la práctica.

### **Pasos de implementación:**

- 1. Simulación de proyecto SCRUM:** Organiza una simulación de proyecto en la que los participantes deben trabajar como un equipo SCRUM. Este puede ser un proyecto ficticio, como el desarrollo de una nueva aplicación o la creación de una campaña de marketing digital, por ejemplo.
- 2. Asignación de roles:** Asigna a cada participante uno de los roles de SCRUM y proporciona un conjunto de tareas que deben completar en un tiempo

determinado puede ser una serie de Sprints. A lo largo de la simulación, los participantes deben:

3. Aplicar los fundamentos de SCRUM para entregar valor continuamente, adaptarse a los cambios y colaborar eficazmente.
  - Mantener la transparencia sobre el avance y los obstáculos del proyecto.
  - **Inspeccionar** y **adaptar** el trabajo de manera frecuente a través de eventos como el Daily Scrum, Sprint Review y Sprint Retrospective.
  - Demostrar los valores de SCRUM, como el compromiso, el coraje, el foco, la apertura y el respeto, al interactuar con los demás miembros del equipo.
4. **Elementos de gamificación:**
  - **Puntos y recompensas:** Cada vez que los participantes demuestren un comportamiento positivo relacionado con los valores o pilares de SCRUM, por ejemplo, comprometerse a completar tareas, mostrar apertura en la retroalimentación, etc., pueden ganar puntos o recompensas simbólicas.
  - **Desafíos y obstáculos:** Introduce obstáculos aleatorios, como cambios en las prioridades del proyecto o recursos limitados, para que los participantes deban adaptarse rápidamente y aplicar los principios de SCRUM para superarlos.
5. **Reflexión final:** Después de completar la simulación, realiza una discusión de grupo sobre cómo se aplicaron los principios de SCRUM. Pregunta a los participantes:
  - ¿Cómo ayudó la transparencia en la toma de decisiones durante el proyecto?
  - ¿Qué desafíos se presentaron al tratar de aplicar los valores de SCRUM y cómo los superaron?
  - ¿Cómo la inspección y adaptación contribuyeron al éxito del equipo en la simulación?

## Evaluación

## **Evaluación: retroalimentación**

### **Objetivo:**

Evaluar la comprensión teórica y conceptual de los participantes sobre los principios, pilares y valores de SCRUM.

### **Descripción:**

Un examen de respuesta escrita puede ser útil para evaluar si los participantes han comprendido los conceptos clave de SCRUM. Este examen incluirá preguntas tanto de opción múltiple como preguntas abiertas para evaluar la comprensión profunda de los temas tratados.

### **Estructura del examen:**

#### **1. Preguntas de opción múltiple:**

¿Cuál de los siguientes es un pilar de SCRUM?

- a) Transparencia
- b) Compromiso
- c) Colaboración
- d) Rendimiento

¿Qué valor de SCRUM se refleja cuando un miembro del equipo señala un obstáculo, aunque sea incómodo?

- a) Respeto
- b) Coraje
- c) Compromiso
- d) Foco

#### **2. Preguntas de respuesta corta:**

- Explica brevemente qué significa la inspección dentro del marco SCRUM y por qué es importante.
- ¿Cómo se aplican los valores de SCRUM en una reunión diaria (Daily Scrum)? Proporcione un ejemplo práctico.

#### **3. Preguntas abiertas:**

- Describe cómo un equipo puede utilizar la transparencia y la adaptación en un proyecto que enfrenta cambios frecuentes en los requisitos del cliente.
- ¿Qué desafíos puede enfrentar un equipo al tratar de implementar el valor de apertura en un Sprint Retrospective? ¿Cómo podría superarlos?

## **Evaluación de aprendizaje: Proyecto práctico de implementación de SCRUM**

### **Objetivo:**

Evaluar la capacidad de los participantes para aplicar los principios de SCRUM en un contexto práctico, trabajando en equipo y enfrentando escenarios reales.

### **Descripción:**

Un proyecto práctico es una excelente forma de evaluar cómo los estudiantes son capaces de aplicar los conceptos de SCRUM a un caso de la vida real. Este tipo de evaluación fomenta el trabajo en equipo, la resolución de problemas y la aplicación directa de los principios, pilares y valores de SCRUM.

### **Pasos del proyecto:**

- 1. División en equipos:** Organiza a los estudiantes en pequeños grupos y asigna a cada grupo un proyecto de desarrollo (por ejemplo, la creación de un sitio web o una aplicación móvil ficticia). Cada equipo debe simular que está utilizando SCRUM para gestionar su proyecto.
- 2. Asignación de roles:** Cada miembro del equipo asume un rol específico dentro del marco SCRUM.
- 3. Aplicación de SCRUM:** A lo largo de un período determinado, por ejemplo, durante varias semanas, los equipos deben:
  - Crear y priorizar un Product Backlog.
  - Planificar y ejecutar al menos un Sprint, idealmente dos o tres, con sus respectivas Daily Scrums, Sprint Review y Sprint Retrospective.
  - Demostrar cómo usan los fundamentos de SCRUM para entregar valor continuo, colaborar de manera efectiva y adaptarse a los cambios.

- Implementar los pilares de SCRUM en su proceso de trabajo (transparencia, inspección y adaptación) y reflejar los valores en sus interacciones diarias.
- 4. Informe de evaluación:** Al final del proyecto, cada equipo debe presentar un informe que incluya:
- Un resumen de cómo aplicaron cada uno de los fundamentos, pilares y valores de SCRUM.
  - Reflexiones sobre los desafíos que enfrentaron y cómo los resolvieron.
  - Un análisis de cómo la inspección y adaptación ayudaron a mejorar el proceso del equipo durante el proyecto.
- 5. Evaluación final:**
- Evalúa cómo cada equipo aplicó SCRUM en la práctica cumplimiento de roles, eventos, artefactos, y resolución de problemas.
  - Proporciona retroalimentación sobre la calidad del trabajo del equipo, la integración de los valores y la efectividad de los eventos SCRUM.

## **UNIDAD 3 – Miembros del equipo SCRUM**

### **Introducción de miembros del equipo SCRUM**

El elemento fundamental de Scrum es un equipo pequeño de personas, conocido como “Scrum Team”. Está conformado por profesionales enfocados en el objetivo del proyecto. Entorno al equipo SCRUM es importante identificar los roles y responsabilidades de cada integrante, así como las interacciones que estos tienen con el proceso del proyecto.

### **Objetivos de aprendizaje**

- Reconocer las características de un equipo de trabajo efectivo en SCRUM, adoptándolas como referencia en habilidades de colaboración y la resolución de conflictos.
- Identificar las principales responsabilidades de los roles en el equipo SCRUM (Scrum Master, Product Owner, Developers), mediante el análisis de las tareas asignadas.

- Conocer la interacción entre roles y cómo la colaboración y la comunicación entre ellos contribuyen al éxito de un proyecto.
- Identificar los beneficios que SCRUM aporta al desarrollo de software en contraste con metodologías tradicionales, centrándose en la productividad y flexibilidad.

## Contenidos

### Scrum Team

El Scrum Team es lo suficientemente pequeño como para seguir siendo ágil y lo suficientemente grande como para completar un trabajo significativo dentro de un Sprint, generalmente 10 personas o menos [16]. Sin embargo, según [16] los equipos más pequeños se comunican mejor y son más productivos.

El Scrum Team es responsable de todas las actividades relacionadas con el producto, desde la colaboración de los interesados, la verificación, el mantenimiento, la operación, la experimentación, la investigación y el desarrollo, y cualquier otra cosa que pueda ser necesaria [16].

En general el Scrum Team debe ser auto organizado y multi disciplinario trabajando en conjunto, de forma que les permita cumplir con el objetivo común, con cada miembro asumiendo roles y responsabilidades específicas, las cuales se detallan en la sección Roles.

### Roles y responsabilidades

#### b.1. Descripción de cada rol

En el marco de desarrollo Scrum, aplicado al desarrollo de software, existen **tres roles** fundamentales: el **Scrum Master**, el **Product Owner** y el Equipo de Desarrollo o **Developers**. Cada rol tiene responsabilidades específicas que aportan al éxito de un proyecto, mediante el trabajo colaborativo y en equipo, A continuación, se detalla una descripción de cada rol:

#### **Scrum Master:**

Es un facilitador y un guía del proceso, esta persona se encarga de ayudar a resolver cualquier asunto que pudiera obstaculizar el proyecto. Además, es la

persona encargada de facilitar la comunicación cliente-equipo y fomentar la autogestión del grupo de trabajo [17].

Es el Líder Servicial del Equipo Scrum quién modera y facilita las interacciones del equipo como entrenador del equipo y motivador. El Scrum Master es responsable de asegurarse que el equipo tenga un ambiente de trabajo productivo mediante la protección del equipo de las influencias externas, la eliminación de todos los obstáculos, y de confirmar que se cumplan los principios, aspectos y procesos de Scrum [18]. Para ejercer este rol se recomienda poseer habilidades de liderazgo, comunicación, y resolución de conflictos.

### **Product Owner:**

Es responsable de maximizar el valor del producto y gestionar el backlog (lista priorizada de todos los requisitos) del producto. Este rol implica priorizar las tareas y asegurar que el equipo de desarrollo entienda claramente los requisitos [19]. El Product Owner es una persona, no un comité que puede representar las necesidades de muchos interesados en el producto backlog. Aquellos que quieran cambiar el backlog pueden hacerlo intentando convencer al Product Owner[16]. El Product Owner siempre debe mantener una visión dual. Él/ella debe entender y apoyar las necesidades e intereses de todos los stakeholders, mientras que comprenden las necesidades y el funcionamiento del Equipo Scrum. Debido a que el Product Owner debe entender las necesidades y prioridades de los stakeholders, incluyendo los Clientes y los usuarios, este papel se conoce comúnmente como la Voz del Cliente [18]. Para ejercer este rol se requiere habilidades de gestión de proyectos.

### **Equipo de Desarrollo o Developers:**

Compuesto por profesionales técnicos que transforman las ideas del Product Owner en resultados tangibles. Son responsables de la entrega del producto [20]. Estas personas se encargan de desarrollar el producto y entregar incrementos en cada sprint (período de tiempo fijo). Los integrantes de este rol deben poseer habilidades técnicas y capacidad para trabajar de manera colaborativa.

En este mismo contexto, aunque no es parte del equipo SCRUM, un actor muy importante en la interacción y proceso del proyecto son los “**Stakeholders**”, los cuales pueden ser internos (SCRUM TEAM) como externos (clientes, usuarios

finales), el aporte de los stakeholders radica en que con su ayuda u opinión se pueda cumplir con los requisitos técnicos y satisfacer las expectativas y necesidades de los interesados.

## **b.2. Responsabilidades de cada rol**

Una vez conocido las características de cada rol, se analiza a continuación las principales responsabilidades o funciones de cada rol:

Según [16], los **Scrum Master** sirven al **Scrum Team** de varias maneras, que incluyen:

Guiar a los miembros del equipo en ser autogestionados y multifuncionales;  
Ayudar al Scrum Team a enfocarse en crear Incrementos de alto valor que cumplan con la Definición de Terminado;  
Procurar (Promover) la eliminación de impedimentos para el progreso del Scrum Team.

Asegurarse de que todos los eventos de Scrum se lleven a cabo y sean positivos, productivos y se mantengan dentro de los límites de tiempo recomendados.

Según [16], los **Scrum Master** sirven al **Product Owner** de varias maneras, que incluyen:

Ayudar a encontrar técnicas para una definición efectiva de objetivos del producto y la gestión del Product Backlog;  
Ayudar al Scrum Team a comprender la necesidad de tener elementos del Product Backlog claros y concisos;  
Facilitar la colaboración de los interesados según se solicite o necesite.

Entre las **responsabilidades** específicas de los **Product Owner**, en cuanto a la gestión del Product Backlog (lista priorizada de requisitos), según [16] se manifiesta las siguientes:

Desarrollar y comunicar explícitamente el objetivo del producto;  
Crear y comunicar claramente los elementos del Product Backlog;  
Ordenar los elementos del Product Backlog;

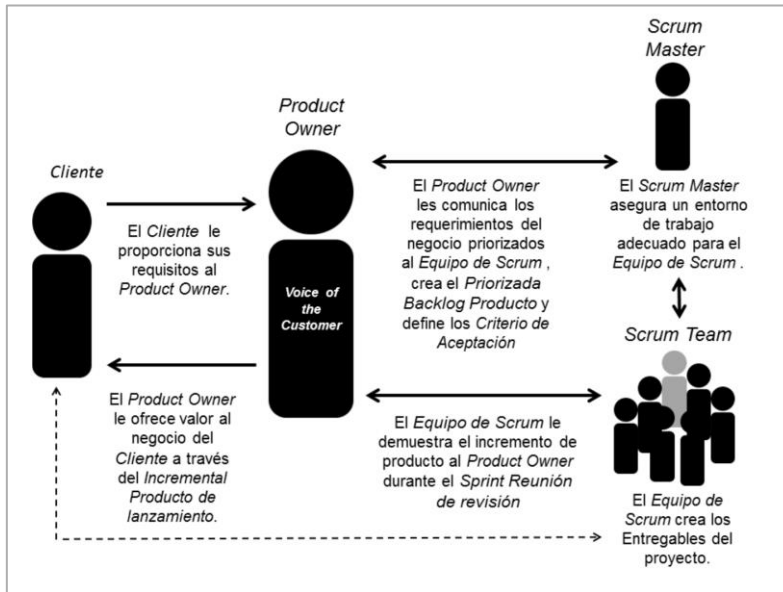
Asegurarse que el Product Backlog sea transparente, visible y se entienda.

Con respecto a las **responsabilidades** específicas de los **Developers**, vienen relacionadas con sus habilidades, según [16], se analiza las siguientes responsabilidades:

Crea un plan para el Sprint, el Sprint Backlog;

Inculcar calidad al adherirse a una definición de Terminado;  
Adaptar su plan cada día hacia el objetivo del Sprint.

**Figura 7**  
Descripción Roles de SCRUM según [18]



### a. Interacción y Colaboración Scrum Team

La interacción entre los diferentes roles en el marco de trabajo SCRUM es fundamental para tener éxito en un proyecto, alcanzando los objetivos propuestos. Este marco de trabajo se caracteriza por su enfoque en la flexibilidad y la colaboración entre los diferentes roles.

En este contexto, además de las interacciones detalladas en la sección anterior donde se describió las responsabilidades de cada rol, se analiza a continuación varias interacciones entre los roles, basadas en la comunicación, colaboración y retroalimentación continua para la resolución de problemas que puedan suscitarse en el proceso:

#### Interacción entre el Scrum Master y el Equipo de Desarrollo

El Scrum Master actúa como un líder, facilitando el funcionamiento del equipo y eliminando obstáculos que puedan surgir durante el desarrollo del proyecto. Su papel es crucial para maximizar la efectividad del equipo, promoviendo prácticas ágiles y asegurando que el equipo siga los principios de Scrum [21] [22].

#### Rol del Product Owner y su Interacción con los Stakeholders

Las competencias clave del Product Owner incluyen la gestión de relaciones y el soporte al usuario, lo cual es esencial para la satisfacción de los stakeholders y la efectividad del equipo [23].

### **Colaboración del Equipo de Desarrollo**

sus miembros deben colaborar estrechamente para cumplir con los objetivos del sprint. La interacción entre el equipo de desarrollo y el Scrum Master es vital para mantener un flujo de trabajo eficiente y adaptarse a los cambios que puedan surgir durante el desarrollo del proyecto [24] [25]. Además, el equipo debe estar en constante comunicación con el Product Owner para asegurar que el producto desarrollado esté alineado con las expectativas de los stakeholders [26].

### **Interacción con los Stakeholders**

La interacción con los stakeholders es un aspecto crucial en Scrum, ya que estos son quienes definen los requisitos y expectativas del producto final. El Product Owner actúa como el principal enlace entre el equipo de desarrollo y los stakeholders, asegurando que las prioridades del producto estén claras y que cualquier cambio en los requisitos sea comunicado de manera efectiva [23]. Esta comunicación constante permite que el equipo de desarrollo ajuste su trabajo según las necesidades del cliente, lo que resulta en un producto que cumple con las expectativas del usuario final [25].

En conclusión, el Scrum Master facilita el trabajo del Scrum Team, el Product Owner gestiona las expectativas de los stakeholders y los developers colaboran para entregar un producto final de calidad. Esta interacción entre roles y stakeholders asegura que el proyecto se desarrolle de manera eficiente y efectiva; adaptándose a los cambios y maximizando el valor del producto final.

## **b. Comparativa entre SCRUM y gestión de proyectos tradicional**

Con base al análisis realizado por [18], se detalla a continuación una tabla que proporciona una visión de las diferencias entre SCRUM y metodologías tradicionales, tomando como referencia sus principales enfoques.

**Tabla 4**  
Comparativo scrum vs metodología tradicional

Enfoque	SCRUM	Metodologías Tradicionales
<b>Planificación</b>	Iterativa y flexible; planificación en Sprints.	Exhaustiva y secuencial; planificación al inicio.
<b>Adaptabilidad al Cambio</b>	Alta; permite ajustes rápidos según la retroalimentación.	Baja; cambios complicados y costosos de implementar.
<b>Colaboración</b>	Fomenta la colaboración y comunicación abierta.	Estructura jerárquica; comunicación de arriba hacia abajo.
<b>Medición del Éxito</b>	Continua; basado en la entrega de incrementos de producto.	Al final del proyecto; entrega del producto final.
<b>Participación del Cliente</b>	Alta; el Product Owner representa la voz del cliente.	Variable; puede ser limitada durante el proceso.
<b>Estructura del Equipo</b>	Auto-organizado; todos los miembros son responsables.	Roles definidos y jerárquicos; es responsabilidad individual.
<b>Ciclo de Retroalimentación</b>	Frecuente; reuniones diarias y revisión de Sprints.	Limitada; retroalimentación al final de cada fase.

En este contexto, se puede resumir los beneficios de SCRUM en sus principales aspectos y enfoques, a continuación:

### **Flexibilidad y Adaptación a Cambios**

SCRUM, se caracteriza por su enfoque incremental e iterativo, lo que permite una mayor adaptabilidad a los cambios en los requisitos del proyecto. A diferencia del modelo tradicional en cascada, que sigue un enfoque lineal y rígido,

SCRUM facilita la incorporación de cambios a través de sprints cortos y revisiones continuas con el cliente. Esto es especialmente beneficioso en proyectos donde los requisitos no están claramente definidos desde el inicio [27].

### **Mejora en la Satisfacción del Cliente**

La satisfacción del cliente es un objetivo primordial en el desarrollo de software, y SCRUM contribuye a alcanzarlo mediante la entrega continua de incrementos de producto que son revisados y aprobados por el cliente. Este enfoque iterativo permite que los clientes vean avances tangibles y proporcionen retroalimentación constante, lo que asegura que el producto final cumpla con sus expectativas y necesidades cambiantes [28].

### **Incremento en la Productividad del Equipo**

SCRUM fomenta la autoorganización y la colaboración dentro de los equipos, lo que puede llevar a un aumento en la productividad. Las reuniones diarias y la clara definición de roles y responsabilidades ayudan a mantener al equipo enfocado y alineado con los objetivos del proyecto. Además, SCRUM promueve la mejora continua de las habilidades del equipo a través de la participación en diversas actividades de desarrollo [29] [30].

### **Control de Calidad y Reducción de Tiempos de Desarrollo**

La implementación de SCRUM ha demostrado mejorar el control de calidad y reducir significativamente los tiempos de desarrollo en comparación con las metodologías tradicionales. Un estudio de caso mostró que el tiempo de desarrollo se redujo en un 75% al utilizar SCRUM, lo que permitió a la organización entregar un sistema de alta calidad en un tiempo mucho menor [27] [31].

## **Actividades**

### **a. Actividad 1**

Resuelva el siguiente crucigrama, tomando como referencia la tabla de comparación entre SCRUM y método tradicional:

**Figura 8**  
Crucigrama

			1				2		
				3					
4						5			
	6						7		
					8				
		9				10			

Pistas:

1. **Horizontal:** Método ágil que permite ajustes rápidos (5 letras).
2. **Vertical:** Planificación exhaustiva y secuencial (9 letras).
3. **Horizontal:** Representa la voz del cliente en SCRUM (12 letras).
4. **Vertical:** Estructura jerárquica en metodologías tradicionales (8 letras).
5. **Horizontal:** Permite a los equipos auto-organizarse (4 letras).
6. **Vertical:** Alta capacidad de respuesta al cambio (5 letras).
7. **Horizontal:** Medición continua en SCRUM (5 letras).
8. **Vertical:** Comunicación abierta y frecuente en SCRUM (10 letras).
9. **Horizontal:** Rol que elimina obstáculos en SCRUM (11 letras).
10. **Vertical:** Entrega del producto final al final del proyecto (6 letras).

## Estrategias de enseñanza

### a. Actividad en clase: Foro de Discusión

**Tema:**

Scrum Team, Roles y responsabilidades

**Objetivo:**

Fortalecer el conocimiento sobre marco de trabajo SCRUM, mediante el análisis de los roles dentro del Scrum Team (Product Owner, Scrum Master y Equipo de Desarrollo), sus responsabilidades y su impacto en la gestión de proyectos.

Instrucciones:

1. Investiga sobre los conceptos clave de los roles en Scrum, toma como referencia la guía SBOK (Conocimiento de Scrum), que se encuentra en el siguiente enlace: [https://www.academia.edu/38121650/Una\\_gu%C3%ADa\\_completa\\_para\\_la\\_entrega\\_de\\_proyectos\\_utilizando\\_Scrum\\_CONOCIMIENTO\\_DE\\_SCRUM\\_GU%C3%8DA\\_SBOK\\_2013\\_Edici%C3%B3n\\_A\\_Guide\\_to\\_the\\_SCRUM\\_BODY\\_OF\\_KNOWLEDGE\\_SBOK\\_Guide](https://www.academia.edu/38121650/Una_gu%C3%ADa_completa_para_la_entrega_de_proyectos_utilizando_Scrum_CONOCIMIENTO_DE_SCRUM_GU%C3%8DA_SBOK_2013_Edici%C3%B3n_A_Guide_to_the_SCRUM_BODY_OF_KNOWLEDGE_SBOK_Guide).
2. Reflexiona sobre la importancia de cada rol y su contribución al éxito de un proyecto Scrum.
3. Responde las siguientes preguntas:

Pregunta:

Desde tu perspectiva, ¿Cómo crees que la falta de comunicación entre el Scrum Master y el Scrum Team podría afectar el éxito de un proyecto?

En tu opinión, ¿cuál de los roles (Scrum Master, Product Owner o Developers) tiene el mayor impacto en la satisfacción del cliente y por qué?

¿Cómo crees que la correcta interacción entre estos roles contribuye al éxito de un Sprint?

Puedes apoyar tu argumento con ejemplos, experiencias personales o referencias a la guía SBOK de Scrum.

4. Comenta al menos dos participaciones de compañeros:

## Evaluación

### Evaluación 1

1. ¿Cuáles son los roles de un Scrum Team, según este marco de trabajo?
  - a) Scrum Manager, Product Owner y Development Team.
  - b) Scrum Tester, Product Owner y Development Team.
  - c) Scrum Project, Product Owner y Scrum Manager.

- d) Ninguno.
2. ¿Cuál es el rol principal del Scrum Master en un equipo SCRUM?
- a) Definir los requisitos del producto.
  - b) Facilitar la comunicación y eliminar obstáculos.
  - c) Desarrollar el producto.
  - d) Priorizar el backlog del product
3. ¿Quién tiene más autoridad ante el equipo de desarrollo, el Scrum Master o
- a) Product Owner?
  - b) Scrum Master.
  - c) Ambas.
  - d) Ninguno.
4. ¿Qué responsabilidad tiene el Product Owner?
- a) Maximizar el valor del producto y gestionar el backlog.
  - b) Dirigir el equipo.
  - c) Organizar las reuniones diarias.
  - d) Mantener las partes interesadas al tanto.
5. ¿Cuál de las siguientes afirmaciones describe mejor al Equipo de Desarrollo (Developers)?
- a) Son responsables de la gestión del backlog.
  - b) Transforman las ideas del Product Owner en resultados tangibles.
  - c) Actúan como intermediarios entre el cliente y el equipo
  - d) Se encargan de eliminar obstáculos del proyecto.

## **Evaluación 2**

1. ¿Cuál es una de las principales responsabilidades del Product Owner en relación con los stakeholders?
- a) Desarrollar el producto final.
  - b) Actuar como el principal enlace entre el equipo de desarrollo y los stakeholders.
  - c) Organizar las reuniones diarias del equipo.
  - d) Establecer las reglas del proceso SCRUM.de
2. Seleccione la respuesta correcta: Guiar al Development Team en ser auto-organizado y multifuncional es una responsabilidad de:

- a) El Gerente de Proyecto.
  - b) El CEO.
  - c) El Arquitecto del Software.
  - d) Ninguna de las anteriores.
3. Seleccione la respuesta correcta: Ayudar al Development Team a crear productos de alto valor es una responsabilidad del:
- a) Scrum Master
  - b) Product Owner.
  - c) Product Manager.
  - d) Coordinador del Proyecto.
4. ¿Cuál es una de las principales responsabilidades del Product Owner en relación con los stakeholders?
- a) Desarrollar el producto final.
  - b) Actuar como el principal enlace entre el equipo de desarrollo y los stakeholders.
  - c) Organizar las reuniones diarias del equipo.
  - d) Establecer las reglas del proceso SCRUM.
5. ¿Qué se espera de la colaboración entre los roles en un equipo SCRUM?
- a) Que cada rol trabaje de manera aislada para evitar conflictos.
  - b) Que haya comunicación constante y retroalimentación para resolver problemas.
  - c) Que el Scrum Master tome todas las decisiones importantes.
  - d) Que los Developers se enfoquen únicamente en la programación.

## UNIDAD 4 – Estructura del trabajo SCRUM

### Objetivos de aprendizaje

Comprender los distintos eventos y artefactos fundamentales en Scrum, para que el estudiante integre en la planificación, ejecución y seguimiento de proyectos ágiles en el contexto del desarrollo de software.

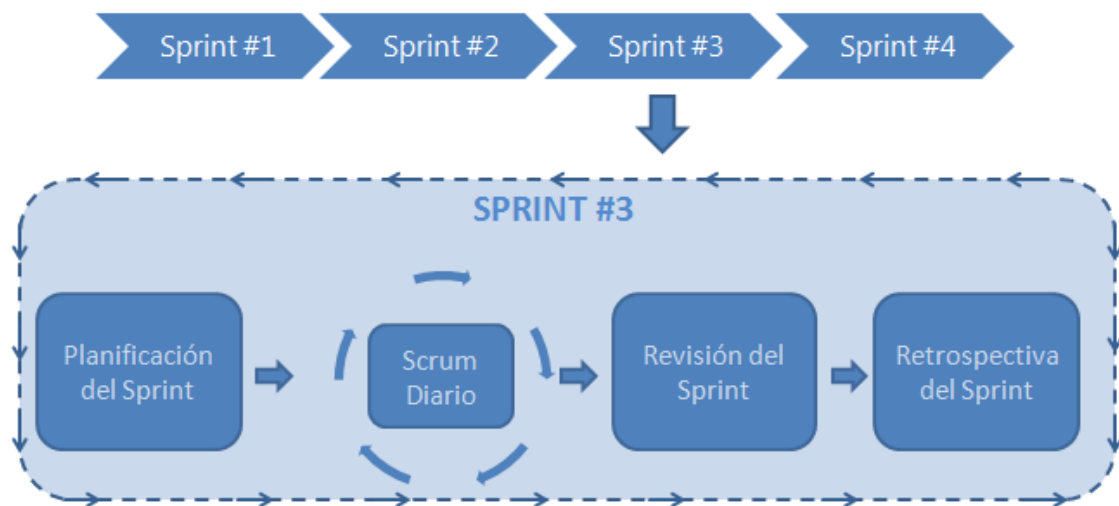
### Contenidos

#### a. Eventos

Un proyecto que sigue Scrum se gestiona en base a cinco eventos claves, que han sido diseñados para facilitar la comunicación, la colaboración y el seguimiento del progreso en la gestión ágil de proyectos. Estos eventos incluyen Sprint, Sprint Planning, Daily Scrum, Sprint Review y Retrospective, y cada uno tiene un propósito específico en el proceso de Scrum.

**Figura 9**

Eventos clave de Scrum mostrando su secuencia cíclica dentro de un Sprint en cuestión [32]



#### a.1. Sprint

Un Sprint es un evento que representa un período de tiempo fijo, generalmente de una a cuatro semanas, durante el cual se lleva a cabo un trabajo específico para crear un incremento del producto. Durante un Sprint, el equipo de desarrollo se enfoca en completar un conjunto de tareas seleccionadas del

Product Backlog que se consideran alcanzables dentro de ese período. El objetivo principal de un Sprint es entregar un resultado funcional y potencialmente entregable al final de este. Esto permite al equipo recibir retroalimentación temprana y frecuente, lo que a su vez ayuda a mejorar el producto de manera continua. Al finalizar cada Sprint, se realiza una revisión y una retrospectiva para evaluar el trabajo realizado y planificar mejoras para el próximo Sprint. Este enfoque ágil no solo promueve la colaboración entre los miembros del equipo, sino que también fomenta una cultura de adaptabilidad y respuesta rápida a los cambios en las necesidades del cliente [33].

### **a.2. Sprint Planning (Planificación del Sprint)**

La Planificación del Sprint es un evento que tiene lugar al inicio de cada Sprint. Su propósito es definir qué se va a trabajar durante el Sprint y cómo se llevará a cabo ese trabajo, asegurando que todos estén alineados y enfocados en los mismos objetivos. Este evento involucra a todo el equipo Scrum, incluyendo al Product Owner, el equipo de desarrollo y el Scrum Master. A continuación, se muestran las actividades a realizar durante la planificación [34].

- **Definición del objetivo del Sprint:** El equipo establece un objetivo claro que guiará el trabajo durante el Sprint. Este objetivo debe ser alcanzable y alineado con las necesidades del cliente.
- **Selección de ítems del Product Backlog:** El Product Owner presenta los ítems más prioritarios del Product Backlog. El equipo de desarrollo discute y selecciona los ítems que considera que puede completar durante el Sprint, teniendo en cuenta su capacidad y la complejidad de las tareas.
- **Desglose de tareas:** Una vez seleccionados los ítems, el equipo descompone cada uno en tareas más pequeñas y manejables. Esto ayuda a tener una mejor comprensión del trabajo que se debe realizar y facilita el seguimiento del progreso.
- **Estimación del esfuerzo:** El equipo estima el esfuerzo necesario para completar cada tarea, lo que permite una mejor planificación y gestión del tiempo durante el Sprint.
- **Compromiso del equipo:** Al finalizar la planificación, el equipo se compromete a alcanzar el objetivo del Sprint y completar los ítems seleccionados.

### a.3. Daily Scrum (Scrum Diario)

El Daily Scrum, también conocido como Scrum Diario, es un evento breve y regular que se lleva a cabo todos los días durante un Sprint. Su principal objetivo es facilitar la comunicación y la colaboración dentro del equipo de desarrollo, asegurando que todos los miembros estén alineados y enfocados en el trabajo. Este es un componente esencial que ayuda a mantener la comunicación fluida, a fomentar la colaboración entre los miembros del equipo y a facilita la identificación de obstáculos, contribuyendo así al éxito del Sprint [35].

#### **Propósito y Estructura del Daily Scrum:**

- **Duración y Frecuencia:** El Daily Scrum suele durar entre 15 minutos y 30 minutos y se realiza a la misma hora y lugar todos los días. Esto ayuda a establecer una rutina y asegura la participación constante del equipo.
- **Formato:** Durante el Daily Scrum, cada miembro del equipo responde a tres preguntas clave:
  - ¿Qué hice ayer que ayudó al equipo a alcanzar el objetivo del Sprint?
  - ¿Qué haré hoy para ayudar al equipo a alcanzar el objetivo del Sprint?
  - ¿Hay algún impedimento que me impida avanzar?
- **Enfoque en la Colaboración:** Este evento no es una reunión de estatus para el Scrum Master o el Product Owner, sino una oportunidad para que el equipo de desarrollo se coordine y comparta información relevante. Fomenta la autoorganización y la responsabilidad compartida.
- **Identificación de Impedimentos:** El Daily Scrum permite identificar rápidamente cualquier obstáculo o impedimento que pueda estar afectando el progreso del equipo. Esto permite que el equipo y el Scrum Master tomen medidas para resolver esos problemas de manera oportuna.
- **Mejora Continua:** Al realizar el Daily Scrum, el equipo tiene la oportunidad de reflexionar sobre su progreso y ajustar su enfoque si es necesario. Esto promueve una cultura de mejora continua y adaptabilidad.

### a.4. Sprint Review (Revisión del Sprint)

El Sprint Review es un evento fundamental que se lleva a cabo al final de cada Sprint. Su propósito principal es revisar el trabajo completado durante el

Sprint y obtener retroalimentación de los interesados sobre el incremento del producto que se ha creado. Este evento fomenta la colaboración y la transparencia, permitiendo que todos los evaluar el progreso, ajustar el rumbo del proyecto y asegurar que se está trabajando en las prioridades correctas, contribuyendo así al éxito general del desarrollo ágil [36].

Estructura del Sprint Review [37]:

- **Presentación del Incremento:** Durante el Sprint Review, el equipo de desarrollo presenta el trabajo completado, mostrando el incremento del producto. Esto puede incluir demostraciones del software, revisión de funcionalidades y cualquier otro aspecto relevante que se haya desarrollado.
- **Retroalimentación de los Interesados:** Los interesados, que pueden incluir al Product Owner, clientes y otros miembros relevantes, tienen la oportunidad de proporcionar retroalimentación sobre el incremento presentado. Esta retroalimentación es crucial para entender si el equipo está alineado con las expectativas y necesidades del cliente.
- **Revisión del Product Backlog:** El equipo y el Product Owner revisan el Product Backlog, discutiendo los ítems completados y los que aún están pendientes. Esto ayuda a priorizar el trabajo futuro y a ajustar el enfoque según las necesidades cambiantes del cliente.
- **Discusión sobre el Futuro:** Se discuten las próximas etapas del proyecto, incluyendo la planificación de los próximos Sprints y la identificación de nuevas oportunidades o desafíos que puedan surgir.

#### **a.6. Sprint Retrospective (Retrospectiva del Sprint)**

La Retrospectiva del Sprint es un evento clave que se lleva a cabo al final de cada Sprint. Su propósito principal es permitir que el equipo reflexione sobre el trabajo realizado durante el Sprint, identifique lo que funcionó bien, lo que no funcionó y cómo se pueden realizar mejoras en el futuro. Este evento fomenta una cultura de aprendizaje y adaptación continua dentro del equipo, fortalece la colaboración entre los miembros del equipo a través de la comunicación abierta y permite identificar riesgos potenciales al discutir desafíos y obstáculos, contribuyendo así a la eficacia y éxito de los futuros Sprints [38].

Estructura de la Retrospectiva del Sprint [39].

- **Revisión del Sprint:** El equipo comienza revisando el trabajo completado durante el Sprint, incluyendo los logros y los desafíos enfrentados.
- **Identificación de Aspectos Positivos:** Se anima a los miembros del equipo a compartir lo que consideran que salió bien, lo que ayuda a reforzar las prácticas efectivas y a reconocer el esfuerzo de todos.
- **Discusión de Desafíos:** El equipo también identifica los obstáculos y problemas que surgieron durante el Sprint. Este es un momento crucial para abordar cualquier impedimento que haya afectado el rendimiento del equipo.
- **Generación de Acciones de Mejora:** A partir de la discusión sobre los aspectos positivos y los desafíos, el equipo propone acciones concretas que pueden implementarse en el próximo Sprint para mejorar el proceso, la colaboración y los resultados.
- **Compromiso con las Mejoras:** Al finalizar la Retrospectiva, el equipo se compromete a llevar a cabo las acciones de mejora acordadas, asegurando que el aprendizaje se traduzca en cambios prácticos.

#### **a.7. Backlog Refinement (Refinamiento del Backlog)**

El Refinamiento del Backlog es un proceso continuo en el marco de Scrum que tiene como objetivo mejorar y mantener el Product Backlog, asegurando que esté alineado con las necesidades del cliente y las prioridades del proyecto. Aunque no se considera un evento formal como los otros en Scrum, es una práctica esencial que contribuye a la efectividad del equipo y del proceso ágil. Al contar con un Backlog bien refinado, el equipo puede planificar los Sprints de forma más eficiente, optimizando el uso del tiempo y los recursos disponibles. Asimismo, esta práctica promueve la adaptabilidad del equipo frente a cambios en las necesidades del cliente o del mercado, asegurando la relevancia y el valor del producto final [40].

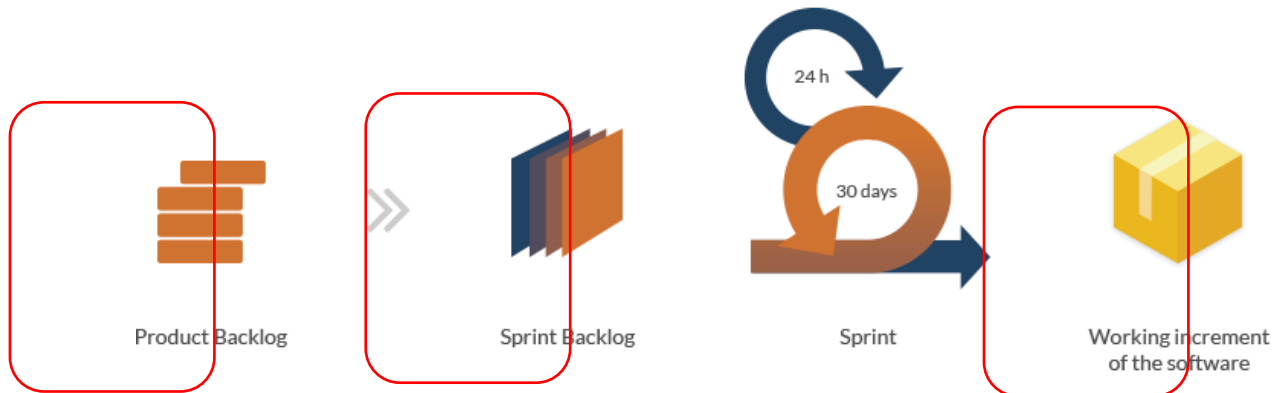
#### **b. Artefactos**

En el marco de Scrum, los artefactos representan el trabajo o valor tangible dentro del proceso ágil. Están diseñados con el propósito de maximizar la transparencia de la información, permitiendo que todos los involucrados tengan una base común para la adaptación y toma de decisiones informadas. Cada

artefacto contiene un compromiso definido para garantizar que se proporcione información en relación con los objetivos medibles [41].

**Figura 10**

Artefactos de Scrum resaltados por su función fundamental en la definición y ejecución de cada Sprint [42]



**b.1. Product Backlog (Lista de Productos)**

Es la lista de todos los requisitos y características que se desean en el producto final. El Product Owner es responsable de su gestión y priorización. Los ítems en el Product Backlog pueden incluir historias de usuario, mejoras, correcciones de errores y cualquier otro trabajo que deba realizarse. Este artefacto es dinámico y evoluciona a medida que se desarrollan nuevas ideas y se obtienen retroalimentaciones [41].

**b.2. Sprint Backlog (Lista del Sprint)**

Es un subconjunto del Product Backlog que contiene los ítems seleccionados para ser completados durante un Sprint específico, junto con un plan para entregar el incremento del producto. El equipo de desarrollo crea y gestiona el Sprint Backlog, asegurándose de que todos los miembros estén alineados con las tareas a realizar. Este artefacto refleja el trabajo que se ha comprometido a completar durante el Sprint [41].

**b.3. Incremento**

Representa la suma de todos los ítems del Product Backlog completados durante un Sprint y los Sprints anteriores. El Incremento debe ser un resultado funcional y potencialmente entregable, lo que significa que debe cumplir con los

criterios de aceptación y estar en condiciones de ser presentado a los interesados. Este artefacto proporciona una visión clara del progreso del proyecto y del valor entregado al cliente [41].

## **Actividades**

### **Actividad 1**

#### **Objetivo:**

Reconocer de manera efectiva los eventos fundamentales que tienen lugar durante un Sprint en el marco de trabajo de Scrum.

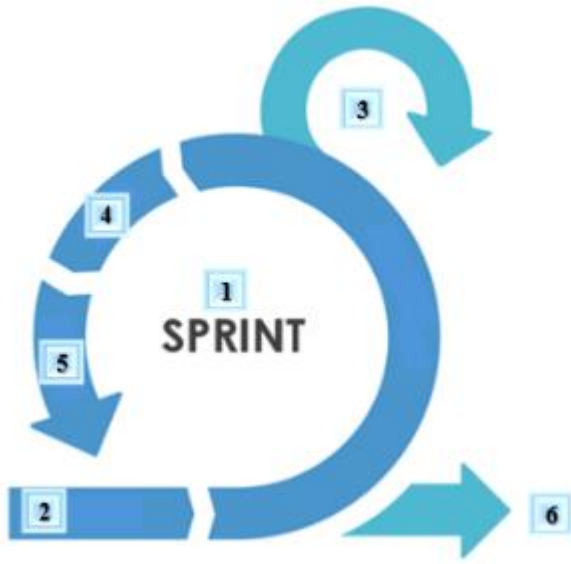
#### **Preparación:**

- Enumera y repasa los eventos fundamentales de Scrum.
- Recordar el esquema gráfico que representa un Sprint dentro del marco Scrum.
- Asociar cada número del esquema con el evento correspondiente.

#### **Desarrollo:**

Identifica todos los eventos clave de Scrum revisados previamente y descríbelos según la numeración del esquema a continuación:

#### **Figura 11** Eventos Clave



1. Sprint

2.

---

3.

---

4.

---

5.

---

6.

---

## Resultado de aprendizaje:

Adquirir conocimiento sólido sobre los eventos clave en un Sprint dentro del marco de trabajo Scrum, así como su importancia para su aplicación de manera efectiva en un entorno ágil e impulsar el éxito del proyecto.

## Actividad 2

### Objetivo:

Fomentar la comprensión detallada y la capacidad de describir con precisión los artefactos clave en Scrum, enfatizando su función, importancia y aplicación dentro del marco ágil.

### Preparación:

- Analizar los conceptos de cada uno de los artefactos revisados en este documento.
- Identificar en qué momento dentro del marco Scrum se determinan.

### Desarrollo:

Identifica el artefacto de Scrum, describe cuál es su función dentro del marco de trabajo y especifica en qué momento se define.

**Figura 12**  
Identificación de Artefactos



**Resultado de aprendizaje:** Profundiza en la comprensión de los artefactos de Scrum y permite describirlos de manera efectiva, para la aplicación de este conocimiento en proyectos ágiles de forma eficaz.

## **Estrategias de enseñanza**

### **Actividad en clase: Asociación Visual de Eventos de Scrum**

Esta estrategia implica presentar a los estudiantes un esquema gráfico numerado que representa cada evento de Scrum en el desarrollo de un Sprint. Los estudiantes deben asociar cada número con el evento correspondiente, describiendo su función y relación con otros eventos. Esta estrategia fomenta la participación, el aprendizaje colaborativo y la comprensión efectiva de la secuencia de eventos clave en el marco de trabajo de Scrum, promoviendo una mejor aplicación práctica en proyectos ágiles.

### **Actividad en clase 2**

La estrategia de enseñanza sumerge a los estudiantes en el estudio detallado de cada artefacto de Scrum, enfatizando su función, importancia y momento de definición dentro del marco ágil. A través de la identificación y descripción precisa de cada artefacto, los estudiantes fortalecen su comprensión práctica y teórica, promoviendo una aplicación efectiva de estos conceptos en proyectos ágiles.

## **Evaluación**

### **Evaluación 1**

Resalta la respuesta correcta de cada pregunta en el siguiente cuestionario:

1. ¿Qué evento de Scrum tiene como objetivo inspeccionar el trabajo completado durante el Sprint y adaptar el Plan de Sprint si es necesario?
  - a) Sprint Review
  - b) Sprint Planning
  - c) Daily Scrum

- d) Sprint Retrospective
2. ¿Cuál de los siguientes eventos tiene una duración máxima recomendada de 15 minutos?
    - a) Sprint Review
    - b) Sprint Planning
    - c) Daily Scrum
    - d) print Retrospective
  3. ¿En qué evento de Scrum los miembros del Equipo de Desarrollo inspeccionan el trabajo desde el último Daily Scrum y planifican el trabajo para las próximas 24 horas?
    - a) Sprint Retrospective
    - b) Sprint Review
    - c) Daily Scrum
    - d) Sprint Planning
  4. ¿Cuál de los siguientes eventos se centra en la inspección y adaptación del producto resultante del Sprint?
    - a) Sprint Review
    - b) Sprint Retrospective
    - c) Daily Scrum
    - d) Sprint Planning
  5. ¿En qué evento se revisa el trabajo realizado y se colabora en posibles mejoras para futuros Sprints?
    - a) Sprint Retrospective
    - b) Sprint Planning
    - c) Daily Scrum
    - d) Sprint Review

## **Evaluación 2**

Resalta la respuesta correcta de cada pregunta en el siguiente cuestionario:

1. ¿Qué artefacto de Scrum representa el trabajo o el valor a entregar al final del Sprint?
  - a) Product Backlog
  - b) Increment
  - c) Sprint Backlog

2. ¿Qué artefacto define la lista ordenada de todo lo que se necesita en el producto y es la única fuente de requerimientos para cualquier cambio?
  - a) Product Backlog
  - b) Increment
  - c) Sprint Backlog
3. ¿Qué artefacto contiene los elementos seleccionados del Product Backlog que el equipo se compromete a completar durante el Sprint?
  - a) Product Backlog
  - b) Increment
  - c) Sprint Backlog

# CAPÍTULO III

## Desarrollo de un Caso Práctico



# Aplicando la Metodología Ágil

## Fundamentación

El desarrollo de software ha evolucionado significativamente en las últimas décadas, adoptando enfoques más flexibles y centrados en la colaboración para responder a las necesidades cambiantes del mercado y los usuarios finales. En este contexto, las metodologías ágiles y puntualmente el Marco de Trabajo SCRUM, han demostrado ser herramientas clave para garantizar la entrega de productos de calidad, maximizando el valor entregado al cliente y promoviendo la adaptabilidad frente a los cambios.

SCRUM es un marco ágil ampliamente utilizado que permite a los equipos trabajar de manera iterativa e incremental, organizando el trabajo en ciclos cortos llamados sprints. Este enfoque fomenta la transparencia, la inspección y la adaptación, lo que resulta en una mejora continua del producto y del proceso de desarrollo. Además, SCRUM se enfoca en la colaboración entre roles claramente definidos (Product Owner, Scrum Master y Equipo de Desarrollo), asegurando que cada miembro del equipo contribuya al éxito del proyecto.

La práctica titulada "Desarrollo de un Caso Práctico Aplicando la Metodología Ágil" tiene como objetivo introducir a los estudiantes en los conceptos fundamentales de SCRUM mediante la aplicación práctica en un entorno realista. En este caso, el proyecto se centra en el desarrollo de una aplicación para la gestión de información de una agencia de empleo. Este contexto, permite a los estudiantes experimentar con problemas reales relacionados con la planificación, priorización y gestión del trabajo en equipo.

Durante esta práctica, se abordarán actividades esenciales como el análisis de requerimientos funcionales, la creación de historias de usuario y épicas, así como el uso de Jira como herramienta para gestionar incidencias y organizar el backlog del producto. Estas actividades no solo permitirán a los estudiantes comprender los principios teóricos de SCRUM, sino también desarrollar habilidades prácticas para su implementación efectiva.

El uso de Jira es particularmente relevante porque es una herramienta ampliamente utilizada en la industria para gestionar proyectos ágiles. Su integración con SCRUM facilita la planificación, seguimiento y ejecución del

trabajo, proporcionando visibilidad al equipo sobre el progreso del proyecto y ayudando a identificar áreas que necesitan atención.

En conclusión, esta práctica fundamenta su importancia en la necesidad de formar profesionales que no solo comprendan las bases conceptuales de SCRUM, sino que también sean capaces de aplicarlas en entornos reales utilizando herramientas modernas como Jira. Esto asegura que los estudiantes estén mejor preparados para enfrentar los desafíos del desarrollo ágil en sus futuras carreras profesionales.

## **Objetivos**

### **Objetivo General**

Desarrollar competencias en la aplicación de la metodología ágil SCRUM a través de la creación y gestión de un proyecto práctico en Jira, enfocado en el desarrollo de una aplicación para la gestión de información de una agencia de empleo.

### **Objetivos Específicos**

1. Analizar los requerimientos funcionales del sistema propuesto, identificando las necesidades de los diferentes roles involucrados (Administrador, Empresa y Postulante)
2. Introducir a los estudiantes en el uso de Jira, proporcionando una comprensión práctica de cómo esta herramienta puede ser utilizada para gestionar proyectos ágiles
3. Crear épicas del backlog del producto, organizando las historias de usuario en categorías que permitan una mejor planificación y ejecución del trabajo durante los sprints.
4. Fomentar el trabajo colaborativo entre los estudiantes, promoviendo la comunicación efectiva y la resolución conjunta de problemas
5. Establecer un marco para la gestión del backlog, que incluya la priorización y estimación inicial de tareas

## **Preparación Previa**

Para garantizar el éxito en la práctica "Desarrollo de un Caso Práctico Aplicando la Metodología Ágil", los estudiantes deben revisar los siguientes conceptos teóricos y herramientas prácticas:

### **1. Fundamentos de SCRUM**

**Ciclo de vida de un sprint:** Comprender las etapas de planificación, ejecución, revisión y retrospectiva, enfocándose en la entrega iterativa de valor [43]

**Roles clave:** Repasar las responsabilidades del Product Owner (gestión del backlog), Scrum Master (facilitación) y Equipo de Desarrollo (ejecución técnica [43] [44]

**Ceremonias ágiles:** Revisar el propósito de las reuniones diarias (daily scrum), planificación de sprints y retrospectivas [44]

### **2. Introducción a Jira**

**Configuración de proyectos:** Familiarizarse con la creación de tableros SCRUM en Jira, incluyendo backlogs y sprints [45].

**Gestión de incidencias:** Aprender a crear y priorizar historias de usuario, épicas y tareas utilizando plantillas predefinidas [45].

**Automatización básica:** Explorar reglas automatizadas para mover tareas entre sprints o notificar actualizaciones [39]

### **3. Elaboración de Historias de Usuario**

**Estructura (INVEST):** Asegurar que las historias sean Independientes, Negociables, Valiosas, Estimables, Pequeñas y Testeables [46]

**Criterios de aceptación:** Definir condiciones claras para validar el cumplimiento funcional [46]

### **4. Planificación de Sprints**

**Estimación de esfuerzo:** Repasar técnicas como Scrum Poker y conceptos como velocidad del equipo para asignar trabajo realista [39]

**Evitar sobrecarga:** Comprender los riesgos de incluir tareas con dependencias externas o alto grado de incertidumbre.

## 5. Recursos Recomendados

Guía Oficial de SCRUM: Actualización 2020 sobre roles y eventos. Disponible en:

<https://scrumguides.org/>

Documentación de Jira: Tutoriales para configuración de sprints y automatización<sup>3</sup>. Disponible en:

<https://www.atlassian.com/es/software/jira/guides>

Agile Manifesto: Principios fundacionales del desarrollo ágil<sup>1</sup>. Disponible en:

<https://agilemanifesto.org/>

## Procedimiento

Para la práctica propuesta vamos a suponer el desarrollo de un aplicativo de **Gestión de Información de una Agencia de Empleo** con el siguiente detalle inicial de los roles que intervienen:

1. **Administrador:** gestión de usuarios, validación de datos de la empresa, gestión de catálogos y visualización de reportes
2. **Empresa:** gestión de datos de la empresa, oferta de empleos y visualización de reportes
3. **Postulante:** registro de hoja de vida, listar ofertas, búsqueda de ofertas y postulación

## Materiales Necesarios

### Hardware:

Computadoras con acceso a Internet.

Proyector (para presentaciones grupales).

### Software:

Cuenta en Jira (puede ser gratuita para equipos pequeños).

Herramienta de documentación (Google Docs).

Herramienta de videoconferencia (Google Meet) para reuniones de forma remota.

### Documentación:

Guía oficial de SCRUM.

Tutoriales de Jira.

Material Adicional:

Pizarras físicas o digitales para lluvia de ideas.

Post-it o herramientas digitales para la creación de historias de usuario.

## **Paso 1: Análisis de Requerimientos Funcionales**

### **Reunión Inicial:**

Organizar una reunión con todos los miembros del equipo para discutir los requerimientos funcionales del sistema.

Utilizar técnicas de lluvia de ideas para identificar las necesidades de cada rol (Administrador, Empresa, Postulante).

Documentar Requerimientos:

Registrar los requerimientos identificados en un documento compartido.

Asegurarse de que cada requerimiento esté alineado con las historias de usuario que se crearán posteriormente.

A continuación, se detallan algunos ejemplos de requerimientos funcionales, el listado completo se encuentra en el anexo 2:

**Tabla 5**  
Ejemplos de requerimientos funcionales

<b>ID</b>	<b>Rol</b>	<b>Requerimiento Funcional</b>	<b>Prioridad</b>
<b>RF1</b>	Administrador	Gestionar usuarios del sistema (crear, editar, eliminar).	Alta
<b>RF5</b>	Empresa	Gestionar los datos de la empresa (información de contacto, descripción, etc.).	Alta
<b>RF8</b>	Postulante	Registrar y gestionar la hoja de vida (datos personales, experiencia laboral, educación, habilidades, etc.).	Alta
<b>RF13</b>	Todos	Autenticación segura de usuarios (inicio de sesión, cierre de sesión,	Alta

---

recuperación de  
contraseña).

---

## Paso 2: Elaboración de Historias de Usuario

### 1. Definir Historias:

- Utilizando el formato estándar ("Como [rol], quiero [acción] para [beneficio]"), redactar al menos 5 historias por rol.

A continuación, se detalla un ejemplo con un formato sugerido:

**Tabla 6**

Historia de usuario

<b>Historia de Usuario</b>	
<b>Numero: 1</b>	Usuario: Cliente
Nombre de historia: Gestión de usuarios del sistema	
<b>Prioridad en negocio: Alta</b>	Riesgo en desarrollo: Bajo
<b>Puntos estimados: 2</b>	Iteración asignada: 1
<b>Programador Responsable: Juan Pérez</b>	
<b>Descripción:</b>	
Como [Administrador] Quiero [Gestionar la información de los usuarios] Para [asegurar que cada usuario tiene la información correcta y un rol con permisos asignados según las necesidades del negocio]	
<b>Validación:</b>	
<ul style="list-style-type: none"><li>• El Administrador puede crear, editar y eliminar información de los usuarios del sistema</li><li>• El administrador asigna un rol a cada usuario según las necesidades del negocio</li></ul>	

## Paso 3: Introducción a SCRUM y Jira

Presentación Teórica:

Realizar una presentación sobre los fundamentos de SCRUM y su importancia en el desarrollo ágil.

Discutir los roles, eventos y artefactos en SCRUM.

Introducir Jira como herramienta para la gestión ágil de proyectos.

Demostración Práctica:

Mostrar cómo crear una cuenta en Jira con credenciales del correo institucional (@ist17dejulio.edu.ec)

Crear un sitio en el entorno de Jira, darle un nombre significativo al sitio (por ejemplo, "proyectos ágiles")

Explorar la interfaz inicial

## **Paso 4: Creación del Proyecto en Jira**

### **Crear un Nuevo Proyecto:**

Iniciar sesión en Jira.

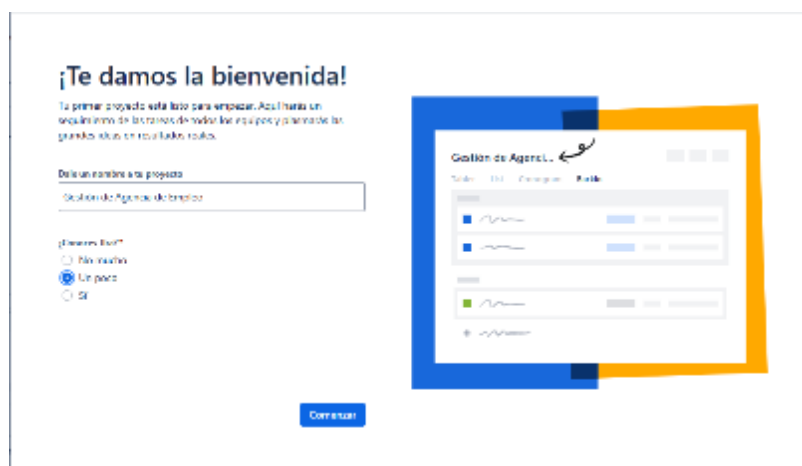
Seleccionar "Crear proyecto" y elegir "Scrum" como tipo de proyecto.

Nombrar el proyecto (por ejemplo, "Gestión de Agencia de Empleo"). Puede ver la guía oficial en video en el siguiente enlace:

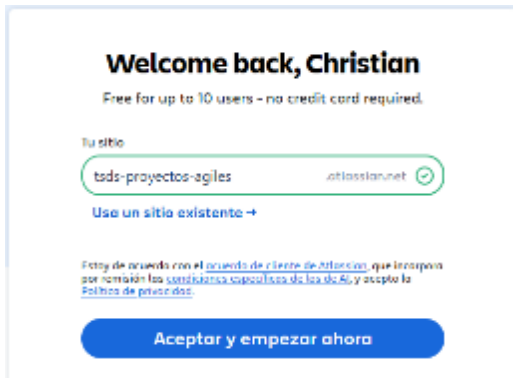
<https://www.youtube.com/watch?v=TV6JWJWAglk>

### **Figura 13**

Creación del proyecto



**Figura 14**  
Creación del sitio



**Figura 15**  
Selección del tipo de Proyecto

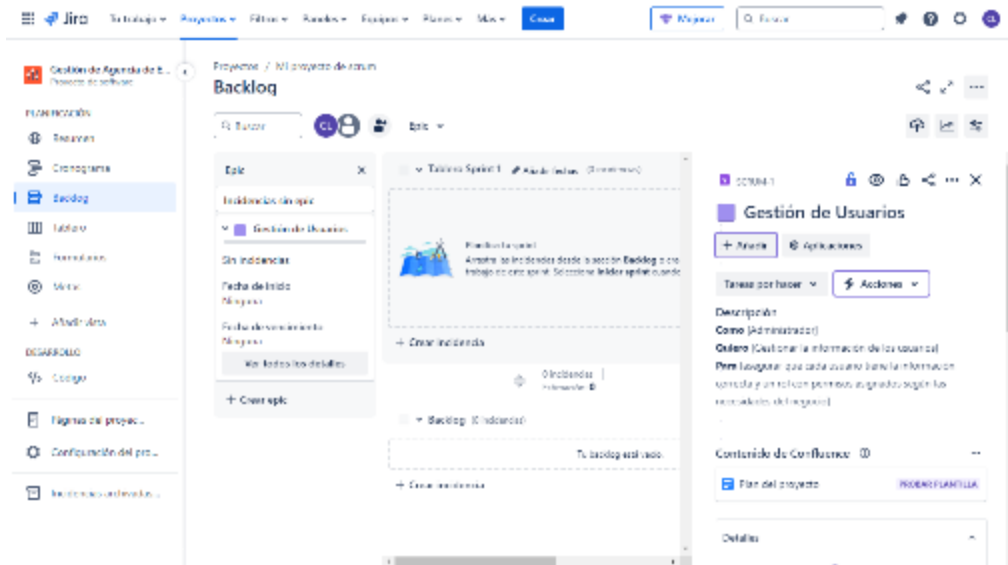


### Configurar el Backlog:

Acceder a la sección "Backlog" del proyecto.

Crear épicas que representen las funcionalidades principales del sistema (por ejemplo, "Gestión de Usuarios", "Gestión de Ofertas", "Postulación").

**Figura 16**  
Épica en Jira



### Resultados de la Práctica

Al completar las actividades detalladas en la práctica "Desarrollo de un Caso Práctico Aplicando la Metodología Ágil", se espera que los estudiantes logren los siguientes resultados de aprendizaje:

### **Comprensión de los Fundamentos de SCRUM:**

Los estudiantes demostrarán una comprensión sólida de los principios, roles, eventos y artefactos de SCRUM.

Serán capaces de explicar cómo SCRUM facilita el desarrollo ágil y la entrega iterativa de valor.

### **Habilidad para Configurar Proyectos en Jira:**

Los estudiantes podrán crear y configurar un proyecto SCRUM en Jira, incluyendo la definición de tipos de incidencias y flujos de trabajo básicos.

Serán capaces de navegar por la interfaz de Jira y utilizar sus funcionalidades principales para la gestión de proyectos ágiles.

### **Capacidad para Analizar Requerimientos Funcionales:**

Los estudiantes podrán identificar y documentar los requerimientos funcionales de un sistema desde la perspectiva de diferentes roles de usuario.

Serán capaces de utilizar técnicas de elicitación de requerimientos, como lluvia de ideas, para obtener información relevante y completa.

### **Destreza en la Elaboración de Historias de Usuario:**

Los estudiantes podrán redactar historias de usuario claras, concisas y centradas en el valor que aportan al usuario final.

Serán capaces de aplicar el formato estándar de historia de usuario y definir criterios de aceptación medibles y verificables.

### **Familiarización con la Gestión del Backlog:**

Los estudiantes comprenderán la importancia de un backlog bien gestionado y cómo este facilita la planificación y el seguimiento del proyecto.

Serán capaces de crear épicas para organizar las historias de usuario y priorizar los elementos del backlog en función del valor y el riesgo.

Estos resultados de aprendizaje están alineados con los objetivos de la práctica y buscan proporcionar a los estudiantes las habilidades y conocimientos necesarios para aplicar SCRUM de manera efectiva en proyectos de desarrollo de software.

## Evaluación del Aprendizaje

Los resultados de aprendizaje se evaluarán según la siguiente rúbrica:

**Tabla 7**  
Rúbrica de evaluación

<b>Criterio de Evaluación</b>	<b>Excelente (4 puntos)</b>	<b>Bueno (3 puntos)</b>	<b>Satisfactorio (2 puntos)</b>	<b>Insuficiente (1 punto)</b>
<b>Comprensión de los Fundamentos de SCRUM</b>	Demuestra un entendimiento profundo y completo de los principios, roles, eventos y artefactos de SCRUM, explicando claramente cómo SCRUM facilita el desarrollo ágil.	Muestra un buen entendimiento de los principios, roles, eventos y artefactos de SCRUM, explicando cómo SCRUM facilita el desarrollo ágil.	Demuestra un entendimiento básico de los principios, roles, eventos y artefactos de SCRUM, aunque la explicación de cómo SCRUM facilita el desarrollo ágil es limitada.	Muestra un entendimiento superficial de los principios, roles, eventos y artefactos de SCRUM, con poca o ninguna explicación de cómo SCRUM facilita el desarrollo ágil.
<b>Habilidad para Configurar Proyectos en Jira</b>	Crea y configura un proyecto SCRUM en Jira de manera eficiente y efectiva,	Crea y configura un proyecto SCRUM en Jira, definiendo tipos de incidencias y	Crea un proyecto SCRUM en Jira, pero la configuración de tipos de incidencias y flujos de	Intenta crear un proyecto SCRUM en Jira, pero la configuración es incompleta o incorrecta,

	definiendo tipos de incidencias y flujos de trabajo personalizados y relevantes para el proyecto.	flujos de trabajo básicos.	trabajos limitados o incorrectos.	demonstrando falta de comprensión de las funcionalidades básicas.
<b>Capacidad para Analizar Requerimientos Funcionales</b>	Identifica y documenta los requerimientos funcionales de manera exhaustiva y precisa, desde la perspectiva de todos los roles de usuario, utilizando técnicas de elicitación efectivas.	Identifica y documenta los requerimientos funcionales de manera adecuada, desde la perspectiva de la mayoría de los roles de usuario, utilizando técnicas de elicitación básicas.	Identifica y documenta los requerimientos funcionales limitados, desde la perspectiva de algunos roles de usuario, sin utilizar técnicas de elicitación claras.	No logra identificar ni documentar los requerimientos funcionales de manera adecuada, demostrando falta de comprensión de las necesidades de los usuarios.
<b>Destreza en la Elaboración de Historias de Usuario</b>	Redacta historias de usuario claras, concisas y centradas en	Redacta historias de usuario claras y centradas en el valor que	Redacta historias de usuario que son comprensibles, pero la	No logra redactar historias de usuario claras, concisas ni

	el valor que aportan al usuario final, aplicando correctamente el formato estándar y definiendo criterios de aceptación medibles y verificables.	aportan al usuario final, aplicando el formato estándar y definiendo criterios de aceptación adecuados.	claridad y el enfoque en el valor son limitados, y los criterios de aceptación son vagos o inexistentes.	centradas en el valor, y los criterios de aceptación son inexistentes o irrelevantes.
<b>Familiarización con la Gestión del Backlog</b>	Gestiona el backlog de manera eficiente, creando épicas relevantes y priorizando los elementos del backlog en función del valor, el riesgo y la dependencia, justificando claramente las decisiones tomadas.	Gestiona el backlog de manera adecuada, creando épicas y priorizando los elementos del backlog en función del valor y el riesgo.	Gestiona el backlog de manera básica, creando algunas épicas y priorizando los elementos del backlog de forma superficial, sin justificar las decisiones tomadas.	No logra gestionar el backlog de manera efectiva, demostrando falta de comprensión de su importancia y cómo priorizar los elementos.

## Actividades Complementarias

Tomando como consideración los requerimientos funcionales descritos en el Anexo 2. Requerimientos funcionales realizar las siguientes actividades:

1. En el grupo de trabajo analizar los requerimientos propuestos en el Anexo 2, pueden modificar los requerimientos funcionales según el criterio del grupo.
2. Para cada rol crear al menos 5 historias de usuario
3. Nombrar, de entre los miembros del equipo de trabajo, un Product Owner quien será el responsable de crear las épicas en Jira y un Scrum Master para que colabore con el Product Owner en la definición de las Épicas
4. El Product Owner debe crear el Sitio y el Proyecto en Jira
5. El Product Owner debe crear las Épicas que se estimen necesarias, considerando las recomendaciones del Scrum Master
6. Discutir con equipo de trabajo las épicas creadas.

# CAPÍTULO IV

## Gestión de Proyectos con Jira definiendo el Sprint Backlog



## Fundamentación

La gestión efectiva de proyectos ágiles con Jira implica utilizar esta herramienta para organizar visualmente el trabajo del equipo, planificar Sprints y colaborar entre los miembros del equipo. Jira permite a los equipos ágiles crear un **Product Backlog**, donde se registran las tareas necesarias para alcanzar los objetivos del proyecto. Además, facilita la colaboración al proporcionar una plataforma centralizada para gestionar Sprints, historias de usuario y tareas técnicas.

En Scrum, el **Sprint Backlog** es un artefacto que resulta directamente de la reunión de planificación del sprint. Este conjunto incluye tanto las historias o elementos seleccionados como el plan detallado para entregarlos durante ese ciclo específico. La gestión efectiva implica descomponer las historias grandes en subtareas manejables y ajustar continuamente según sea necesario a través de Daily Scrums.

La práctica titulada "**Gestión de Proyectos con Jira, definiendo el Sprint Backlog**" tiene como objetivo introducir a los estudiantes en la creación de uno de los artefactos más críticos y fundamentales en la gestión de SCRUM. En este caso, la práctica se centra en el mismo caso anterior la gestión de información de una agencia de empleo, pero enfocándose en el objetivo del Sprint. Este contexto, permite a los estudiantes experimentar con problemas reales relacionados con la priorización, estimación y planificación del trabajo en equipo.

Durante esta práctica, se utilizará como base el trabajo generado en el capítulo anterior con el objetivo de complementarlo mediante actividades como el análisis de tareas técnicas, la gestión de épicas, así como el uso de Scrum Poker como herramienta para estimación de incidencias y tareas técnicas. Estas actividades no solo permitirán a los estudiantes comprender los principios teóricos de SCRUM, sino también desarrollar habilidades prácticas para su implementación efectiva.

El uso de Scrum Poker también llamado Planning Poker es una técnica de estimación ágil utilizada para asignar puntos de complejidad a tareas o historias de usuario en proyectos Scrum, parte por involucrar a todos los miembros del equipo en la discusión de la complejidad del trabajo mediante estimaciones anónimas iniciales donde cada miembro del equipo recibe un conjunto de cartas

con valores basados en la secuencia de Fibonacci modificada (ejemplo: 0, 1/2, 1, 2, 3, 5, 8, 13), para continuar con una especie de análisis grupal lo que reduce la influencia de opiniones dominantes con el objetivo de lograr un acuerdo colectivo.

La gestión de proyectos ágiles con Jira, centrada en la creación del Sprint Backlog, proporciona a los estudiantes una experiencia práctica invaluable en la implementación de metodologías ágiles ya que, al utilizar Jira como herramienta central, los estudiantes aprenden a visualizar, organizar y priorizar tareas, a colaborar de manera efectiva en equipos y aplicar técnicas de estimación como el Scrum Poker. Esta práctica no solo fortalece los conocimientos teóricos sobre Scrum, sino que también desarrolla habilidades prácticas esenciales para el éxito en proyectos ágiles, preparando a los estudiantes para enfrentar desafíos reales en entornos de trabajo colaborativos y dinámicos.

## **Objetivos**

### **General:**

Desarrollar la capacidad de los estudiantes en la aplicación del Marco de Desarrollo Scrum en el Desarrollo de Software utilizando la herramienta de Gestión de Proyectos Jira.

### **Específicos:**

- Planificar un Sprint mediante la generación de incidencias y tareas técnicas.
- Estimar los puntos de historia de una incidencia o tarea técnica utilizando la herramienta Scrum Poker.
- Generar el artefacto Sprint Backlog mediante la planificación y ejecución del Sprint

## **Preparación Previa:**

### **1. El Sprint**

Es un componente esencial del marco de trabajo Scrum, que es ampliamente adoptado en la industria del desarrollo de software ágil. Los Sprints

son intervalos cortos y fijos durante los cuales se desarrollan incrementos de producto en este caso pueden ejecutarse de 1 a 4 semanas [47], permitiendo una entrega continua y adaptativa de software [48]. La planificación del Sprint es crucial, ya que establece los objetivos de la próxima entrega y traza los planes para alcanzarlos [49].

## **2. Las ceremonias de Scrum**

Son fundamentales para la gestión ágil de proyectos y se componen de varias etapas clave: planificación del sprint, reuniones diarias, revisión del sprint y retrospectiva del sprint. La planificación del sprint es crucial para definir los objetivos y tareas del sprint, asegurando que el equipo esté alineado con las metas del proyecto [50] mientras que las reuniones diarias fomentan la comunicación y el seguimiento del progreso, permitiendo a los equipos ajustar sus estrategias según sea necesario [50], la revisión del sprint ofrece una oportunidad para evaluar el trabajo completado y recibir retroalimentación, mientras que la retrospectiva del sprint se centra en la mejora continua del equipo y del proceso, utilizando análisis estadísticos para mejorar la calidad del producto y la productividad del equipo [51].

## **3. La priorización**

En Scrum es un proceso crítico que involucra la gestión del Product Backlog por parte del Product Owner, quien debe maximizar el valor del producto mediante la selección adecuada de las historias de usuario [52]; este proceso puede ser desafiante debido a la complejidad de los requisitos y la necesidad de equilibrar múltiples factores, como el valor de negocio, la priorización del equipo Scrum y el costo del retraso [53].

Otro artefacto importante es el Sprint Backlog, se trata de una lista de tareas seleccionadas del Product Backlog que el equipo se compromete a completar durante un Sprint (tiempo de Iteración), una técnica ampliamente utilizada para estimar y planificar el Sprint Backlog es el Planning Poker, ya que involucra a todos los miembros del equipo en la estimación de las historias de usuario, asegurando una planificación más precisa y efectiva [54].

#### 4. **La estimación de puntos de historia**

Es esencial para planificar de manera efectiva los Sprints en Scrum ya que contribuye en la priorización de elementos de alto valor y a optimizar el tiempo de trabajo desde el inicio del proyecto. Aunque métodos como Planning Poker y el juicio de expertos son comunes, pueden ser subjetivos e inexactos [55]. Recientemente, se han desarrollado enfoques basados en inteligencia artificial, como GPT2SP, que utilizan modelos de lenguaje preentrenados para mejorar la precisión de las estimaciones de puntos de historia, siendo más transferibles e interpretables que los métodos tradicionales [55]. Estos avances subrayan la importancia de la IA en la mejora de la estimación de esfuerzos en entornos ágiles [55].

#### 5. **Visualización del progreso del Sprint (gráficos de burndown, tablero Kanban)**

Tanto los gráficos de burndown como los tableros Kanban son herramientas valiosas para la visualización del progreso en proyectos ágiles. Los gráficos de burndown ayudan a los equipos a seguir el progreso y prever la entrega de valor, mientras que los tableros Kanban mejoran la eficiencia del flujo de trabajo al visualizar y gestionar el trabajo en progreso. Ambas herramientas son fundamentales para mejorar la comunicación y la gestión de proyectos en entornos ágiles y vienen embebidos dentro del entorno de trabajo Jira [49].

### **Procedimiento:**

#### 1. **Materiales Requeridos**

**Acceso a Jira:** Cuenta en Atlassian (versión Cloud gratuita o licencia educativa).

**Equipo Scrum:**

**Scrum Master:** Facilitador del proceso.

**Product Owner:** Responsable del Product Backlog.

**Developers:** 3-5 miembros encargados de desarrollar las tareas.

**Historias de Usuario:** Documento con requisitos funcionales de los roles: Administrador, Empresa y Postulante.

**Plantillas:** Plantilla de proyecto Scrum en Jira.

**Herramienta de Estimación:** Scrum Poker Online.

## Paso 1: Configuración del Proyecto en Jira

- 1) Crear proyecto: Seleccionar "Crear proyecto" → "Scrum" → Nombrar proyecto: "Gestión de Agencia de Empleo".
- 2) Invitar al equipo: Agregar miembros (Scrum Master, Product Owner, Developers) mediante sus correos electrónicos.
- 3) Asignar roles en "Configuración del proyecto" → "Permisos".

## Paso 2: Definición del Product Backlog

- 1) Ir a "Backlog" → "Crear épica" → Agregar descripción y prioridad.
- 2) Desglosar en historias de usuario.

Ejemplo para Épica 1:

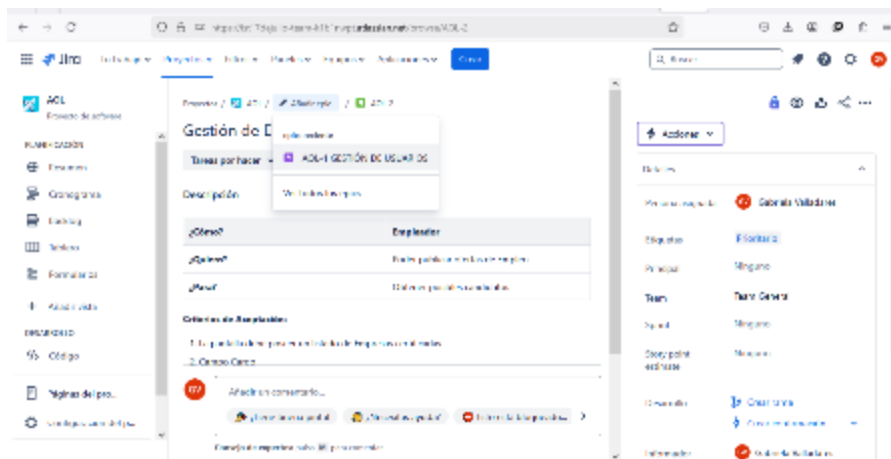
HU-01: "Validar datos de empresas registradas".

HU-02: "Gestionar catálogos de usuarios".

Usar la opción "Crear historia" y vincularla a la épica correspondiente.

### Figura 17

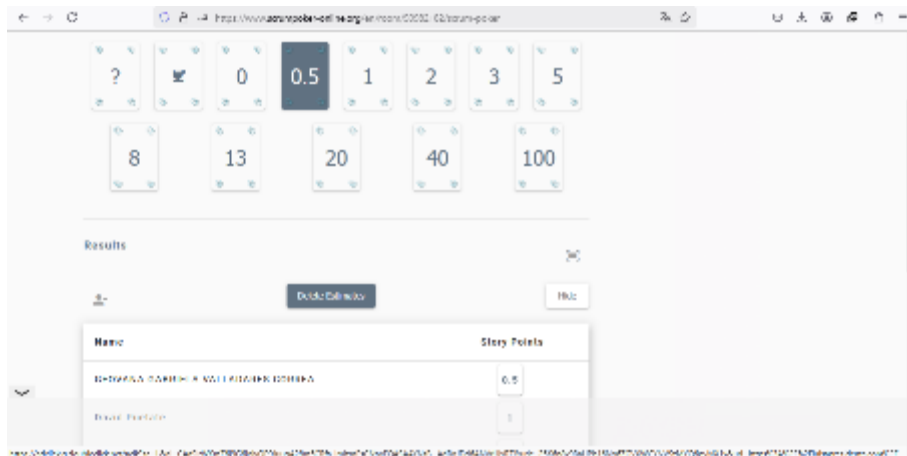
Asignación de épicas a la Historia de Usuario



## Paso 3: Estimación de Complejidad con Scrum Poker

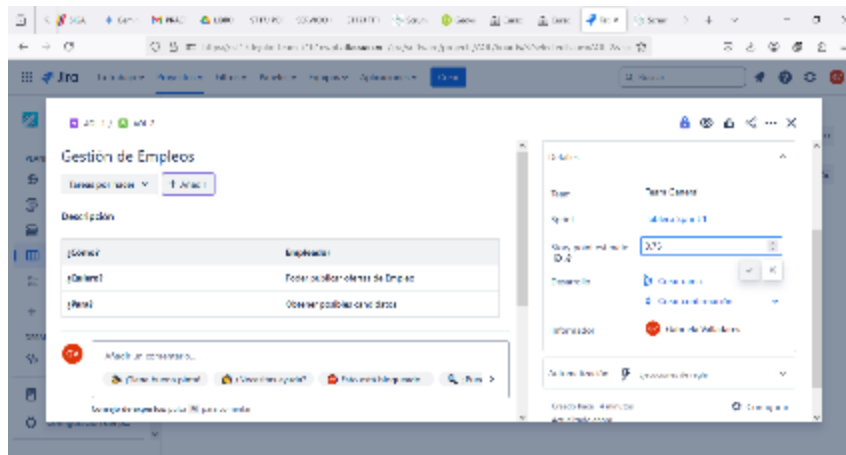
- 1) Reunión de estimación: El Scrum Master presenta cada historia de usuario al equipo.
- 2) Usar Scrum Poker Online para votar puntos de complejidad. Equipo reunido con tarjetas físicas o pantalla compartida.

**Figura 18**  
Estimación Grupal Historia de Usuario



3) Registrar estimaciones en Jira: Promediar los valores de la estimación realizada y asignar dicho valor en la sección "Story point estimate".

**Figura 19**  
Asignación de valor de la estimación en Jira



#### **Paso 4: Planificación del Sprint**

1) Seleccionar Sprint Backlog: Arrastrar historias desde el "Product Backlog" al "Sprint Backlog".

2) Definir objetivo del Sprint : "Implementar módulo de gestión de usuarios".

3) Dividir en tareas técnicas:

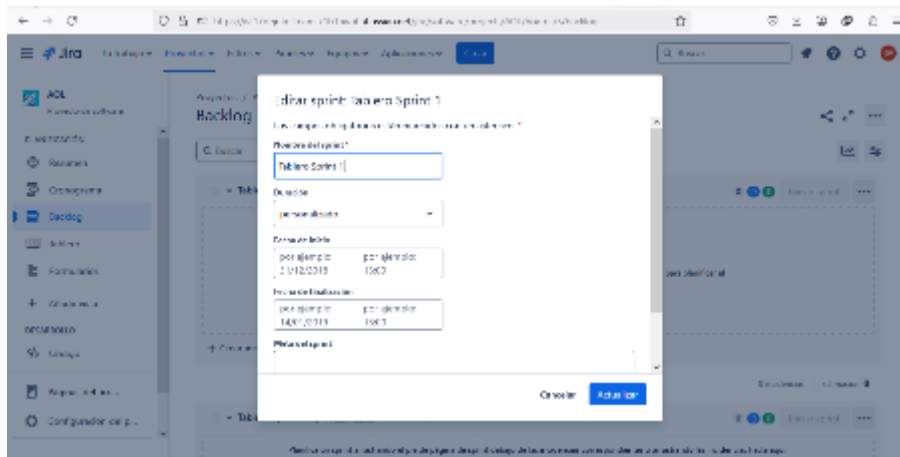
Ejemplo para HU-01:

Tarea 1: "Diseñar formulario de validación de datos".

Tarea 2: "Implementar API de verificación".

Usar "Crear subtarea" en Jira.

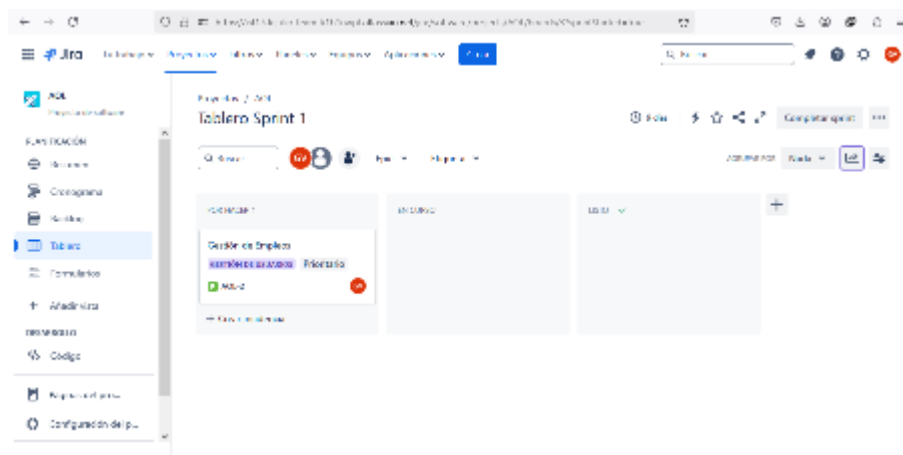
**Figura 20**  
Creación del Sprint



### Paso 5: Ejecución del Sprint

- 1) Configurar tablero de sprint: Arrastrar tareas a columnas según su estado.  
Columnas: "To Do", "In Progress", "Done".

**Figura 21**  
Table Sprint 1 gestionado por estado



### Recursos Recomendados

- Guía Oficial de SCRUM: Actualización 2020 sobre roles y eventos<sup>12</sup>.  
Disponible en: <https://scrumguides.org/>

- Documentación de Jira: Tutoriales para configuración de sprints y automatización<sup>3</sup>. Disponible en: <https://www.atlassian.com/es/software/jira/guides>
- Agile Manifiesto: Principios fundacionales del desarrollo ágil<sup>1</sup>. Disponible en: <https://agilemanifesto.org/>

## Resultados Esperados:

Al finalizar esta práctica, se espera que los estudiantes sean capaces de:

- **Crear y gestionar un Sprint Backlog efectivo:** Dominar el proceso de descomponer grandes tareas en subtareas manejables y priorizarlas según su valor.
- **Utilizar Jira como herramienta de colaboración:** Trabajar de manera colaborativa en un entorno virtual, utilizando las funcionalidades de Jira para asignar tareas, realizar seguimiento del progreso y gestionar cambios.
- **Aplicar técnicas de estimación ágil:** Utilizar el Scrum Poker para estimar la complejidad de las tareas y llegar a un consenso dentro del equipo.
- **Comprender los principios fundamentales de Scrum:** Aplicar los conceptos clave de Scrum, como el Sprint, el Product Backlog y el Daily Scrum, en un contexto práctico.
- **Desarrollar habilidades de planificación y organización:** Planificar y organizar el trabajo de manera eficiente, adaptándose a los cambios y priorizando las tareas.

## Evaluación de Aprendizaje:

Para la evaluación de la práctica se tomará en cuenta los Criterios de Evaluación descritos en la siguiente tabla:

**Tabla 8**

Resultados de aprendizaje

<b>Criterio de Evaluación</b>	<b>Excelente (4 puntos)</b>	<b>Bueno (3 puntos)</b>	<b>(3 Satisfactorio (2 puntos))</b>	<b>Insuficiente (1 punto)</b>
Comprensión de Scrum	Demuestra una comprensión limitada de los conceptos clave de Scrum.	Demuestra una comprensión básica de los conceptos clave de Scrum.	Demuestra una comprensión sólida de los conceptos clave de Scrum y su aplicación en la práctica.	Demuestra una comprensión profunda de Scrum y puede aplicar los conceptos en diferentes contextos.
Creación del Sprint Backlog	El Sprint Backlog está completo o es poco detallado.	El Sprint Backlog no es completo, pero carece de claridad o priorización.	El Sprint Backlog está bien estructurado y priorizado.	El Sprint Backlog es bien estructurado, priorizado y demuestra un análisis exhaustivo de las tareas.
Utilización de Jira	Utiliza Jira de forma básica y limitada.	Utiliza Jira de forma adecuada para gestionar las tareas y el progreso del proyecto.	Utiliza Jira de manera eficiente y personalizada para optimizar el flujo de trabajo del equipo.	Utiliza Jira de forma avanzada, explorando funcionalidades adicionales y creando informes personalizados.
Colaboración en equipo	Participa poco en las	Participa en las	Colabora de manera	Demuestra un liderazgo

	actividades del equipo.	actividades del equipo, pero siempre de manera activa.	efectiva con el equipo, compartiend o ideas y resolviendo conflictos.	positivo y fomenta la colaboración dentro del equipo.
Estimación de tareas	Tiene dificultades para estimar las tareas.	Estima las tareas de manera inconsistente .	Utiliza técnicas de estimación ágil (como Scrum Poker) de manera efectiva.	Justifica sus estimaciones y puede adaptarlas según sea necesario.
Adaptación a los cambios	Tiene dificultades para adaptarse a los cambios en el proyecto.	Se adapta a los cambios de manera reactiva.	Proactiva en la identificación de cambios y en la adaptación del plan.	Lidera el cambio y propone soluciones innovadoras.

## Actividades Complementarias:

- 1) Asignar Tareas Técnicas a cada Historia de Usuario planificada.
- 2) Realizar Daily Scrum: Reunión diaria de 15 minutos, cada Developer responde:
  - ¿Qué hice ayer?
  - ¿Qué haré hoy?
  - ¿Qué impedimentos tengo?
- 3) Sprint Review: Presentar incremento (ejemplo: "Formulario de registro de empresas"), documentar feedback del Product Owner.
- 4) Sprint Retrospective:
  - Identificar mejoras (ejemplo: "Reducir tiempos en Daily Scrum").
  - Registrar acuerdos en "Conclusión del Sprint".

- 5) Exportar métricas: Generar "Burndown Chart" (quema de trabajo) y "Velocity Chart" (velocidad del equipo).
- 6) Revisar progreso del proyecto y el Sprint mediante la perspectiva del tablero Kanban.

## Referencias

- [1] redhat.com, «<https://www.redhat.com>,» 19 07 2022. [En línea]. Available: <https://www.redhat.com/es/topics/devops/what-is-agile-methodology>. [Último acceso: 29 01 2025].
- [2] F. Lanfranchi, «[www.b2econsulting.com](http://www.b2econsulting.com),» [En línea]. Available: <https://www.b2econsulting.com/introduction-to-agile-methodology-and-its-importance/>. [Último acceso: 29 01 2025].
- [3] I. d. Souza, «<https://blog.zapsign.com.br>,» 14 03 2023. [En línea]. Available: <https://blog.zapsign.com.br/en/metodologia-agil/>. [Último acceso: 29 01 2025].
- [4] S. Domínguez, «<https://openwebinars.net>,» 03 10 2023. [En línea]. Available: <https://openwebinars.net/blog/agile-learning-la-metodologia-agil-aplicada-al-aprendizaje/>. [Último acceso: 29 01 2025].
- [5] . M. Bindrees , R. Pooley , I. Ibrahim y N. Taylor, «[agilemanifesto.org](http://agilemanifesto.org),» 13 02 2001. [En línea]. Available: <https://agilemanifesto.org/history.html>. [Último acceso: 29 01 2025].
- [6] D. Torgeir, «[www.researchgate.net](http://www.researchgate.net),» 06 2012. [En línea]. Available: [https://www.researchgate.net/publication/236211358\\_A\\_decade\\_of\\_agile\\_methodologies\\_Towards\\_explaining\\_agile\\_software\\_development](https://www.researchgate.net/publication/236211358_A_decade_of_agile_methodologies_Towards_explaining_agile_software_development). [Último acceso: 29 01 2025].
- [7] H. James, Agile Software Development Ecosystems, Addison-Wesley, 2022.
- [8] sqdm.com, «<https://sqdm.com>,» 28 11 2023. [En línea]. Available: <https://sqdm.com/es/metodologias-en-el-desarrollo-de-software/>. [Último acceso: 29 01 2025].
- [9] «<https://www.syloper.com>,» [En línea]. Available: <https://www.syloper.com/es/blog/desarrollo-agil-vs-desarrollo-tradicional-ventajas-y-consideraciones-esenciales/>. [Último acceso: 29 01 2025].
- [10] gurusis.com, «<https://gurusis.com>,» 8 10 2024. [En línea]. Available: <https://gurusis.com/metodologias-de-desarrollo-de-software/#:~:text=Las%20metodolog%C3%ADas%20tradicionales%20de>

%20desarrollo,%2C%20implementaci%C3%B3n%2C%20verificaci%C3%B3n%20y%20mantenimiento.. [Último acceso: 29 01 2025].

- [11] S. Laoyan, «asana.com,» 08 01 2025. [En línea]. Available: <https://asana.com/es/resources/agile-methodology>. [Último acceso: 29 01 2025].
- [12] J. L. Monte Galiano, *Implantar SCRUM con éxito*, Barcelona: UOC, 2017.
- [13] Harvard Business School Publishing, «Harvard Business Review,» Enero 1986. [En línea]. Available: <https://hbr.org/1986/01/the-new-new-product-development-game?language=es>. [Último acceso: 30 Enero 2024].
- [14] S. Sachdeva, «Scrum Methodology,» *International Journal Of Engineering And Computer Science*, vol. 5, n<sup>o</sup> 6, pp. 16792-16799, 6 Junio 2016.
- [15] S. Jean, V. Aurélien y S. Angel, *Scrum Un método ágil para sus proyectos*, Barcelona: Ediciones ENI, 2018.
- [16] S. Ken y S. Jeff, «La Guía Scrum,» 2020.
- [17] S. A. Hoc, «Los roles en Scrum, la importancia de los equipos Ágiles,» 2025. [En línea]. Available: <https://solvingadhoc.com/los-roles-scrum-la-importancia-los-equipos-agiles/>. [Último acceso: 10 02 2025].
- [18] J. Cortes, *A guide to the Scrum Body of knowledge (SBOK Guide)*, Phoenix, Arizona, 2013.
- [19] F. Fowler, «The Scrum Development Team. Navigating Hybrid Scrum Environments,» 2018. [En línea]. Available: [https://doi.org/10.1007/978-1-4842-4164-6\\_6](https://doi.org/10.1007/978-1-4842-4164-6_6). [Último acceso: 10 02 2024].
- [20] N. Holtzhausen, «A Systematic Review of SCRUM in Software Development.,» *JOIV : International Journal on Informatics Visualization.*, [En línea]. Available: <https://doi.org/10.30630/JOIV.3.2.167>. [Último acceso: 10 02 2025].
- [21] Karabiyik, T., Jaiswal, A., Thomas, P. y Magana, A., «Understanding the Interactions between the Scrum Master and the Development Team: A Game-Theoretic Approach. Mathematics.,» 2020. [En línea]. Available: <https://doi.org/10.3390/math8091553>.
- [22] Shastri, Y., Hoda, R. y Amor, R., «Spearheading agile: the role of the scrum master in agile projects. Empirical Software Engineering.,» 2021. [En línea].

Available: <https://doi.org/10.1007/s10664-020-09899-4>. [Último acceso: 10 02 2025].

- [23] Oomen, S., Waal, B., Albertin, A. y Ravesteyn, P., How can Scrum be Successful? Competences of the Scrum Product Owner., 2017.
- [24] Wonohardjo, E., Sunaryo, R. y Sudiyono, Y., «A Systematic Review of SCRUM in Software Development,» JOIV : International Journal on Informatics Visualization., 2019. [En línea]. Available: <https://doi.org/10.30630/JOIV.3.2.167>. [Último acceso: 10 02 2025].
- [25] Mahnic, V., «A Capstone Course on Agile Software Development Using Scrum.,» IEEE Transactions on Education., 2012. [En línea]. Available: <https://doi.org/10.1109/TE.2011.2142311> . [Último acceso: 10 02 2025].
- [26] Efendi, G., Mutawalli, L., Akbar, J., Rahmat, J., Mataram, P. y Lombok, K., & , T., «Perancangan Sistem Pelaporan Incident Hack Di Kominfo Lombok Tengah Menggunakan Agile Pendekatan Scrum.,» 2024. [En línea]. Available: <https://doi.org/10.62951/repeater.v2i3.104>. [Último acceso: 10 02 2025].
- [27] Azanha, A., Argoud, A., De Camargo, J. y Antonioli, P., «Agile project management with Scrum.,» International Journal of Managing Projects in Business, 2017. [En línea]. Available: <https://doi.org/10.1108/IJMPB-06-2016-0054>. [Último acceso: 10 02 2025].
- [28] Hidalgo, E., «Adapting the scrum framework for agile project management in science: case study of a distributed research initiative,» Heliyon. , 2019. [En línea]. Available: <https://doi.org/10.1016/j.heliyon.2019.e01447>. [Último acceso: 10 02 2025].
- [29] Morandini, M., Coleti, T., Oliveira, E. y Corrêa, P., «Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams.,» Comput. Sci. Rev., 2021. [En línea]. Available: <https://doi.org/10.1016/j.cosrev.2020.100314>. [Último acceso: 10 02 2025].
- [30] Malhotra, R. y Chug, A., «Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance.,» 3rd

International Conference on Computing for Sustainable Global Development (INDIACom), 2016. [En línea].

- [31] Mihailides, G. y Young, W., «Is Agile Scrum Improving IT Project Outcomes?», THE INTERNATIONAL JOURNAL OF MANAGEMENT SCIENCE AND BUSINESS ADMINISTRATION., 2024. [En línea]. Available: <https://doi.org/10.18775/ijmsba.1849-5664-5419.2014.102.1004>. [Último acceso: 10 02 2025].
- [32] M. E. Arevalo Lizardo, «Los 05 Eventos Scrum: Evento 01 El Sprint #scrum,» [En línea]. Available: <https://arevalomaria.wordpress.com/2017/12/18/los-05-eventos-scrum-evento-01-el-sprint-scrum/>. [Último acceso: 10 02 2025].
- [33] M. N. Aziz, «Increase productivity of team members in a Scrum project,» International Journal of Natural and Social Sciences, 2020. [En línea]. Available: <https://10.5281/zenodo.3957840>.. [Último acceso: 10 02 2025].
- [34] J. W. Liu, C. Y. Ho, J. Y. T. Chang y J. C. A. Tsai, «The role of Sprint planning and feedback in game development projects: Implications for game quality,» Journal of Systems and Software, 2019. [En línea]. Available: <https://10.1016/J.JSS.2019.04.057>.. [Último acceso: 10 02 2025].
- [35] M. Katic, «Effects of Daily Scrum Meeting on Software Quality - Open Questions,» SQAMIA, 2024. [En línea]. [Último acceso: 10 02 2025].
- [36] DoneTonic, «¿Qué es el Sprint Review?,» DoneTonic, 2025. [En línea]. Available: <https://donetonic.com/es/que-es-el-sprint-review/>. [Último acceso: 11 02 2025].
- [37] alaimolabs.com, «Qué es la Sprint Review - Alaimo Labs.,» alaimolabs, 2025. [En línea]. Available: [https://alaimolabs.com/es/self-learning/scrum/que-es-la-sprint-review?srsltid=AfmBOoo6txgMrIdUkX5J2NckkX\\_PwD4tCrb21uNXYaoWV5Do-2VME5Ut#tems-terminados](https://alaimolabs.com/es/self-learning/scrum/que-es-la-sprint-review?srsltid=AfmBOoo6txgMrIdUkX5J2NckkX_PwD4tCrb21uNXYaoWV5Do-2VME5Ut#tems-terminados). [Último acceso: 10 02 2025].
- [38] Scrum.org, «¿Qué es una retrospectiva de sprint?,» 2025. [En línea]. Available: <https://www-scrum-org.translate.google/resources/what-is-a-sprint->

retrospective?\_x\_tr\_sl=en&\_x\_tr\_tl=es&\_x\_tr\_hl=es&\_x\_tr\_pto=tc.

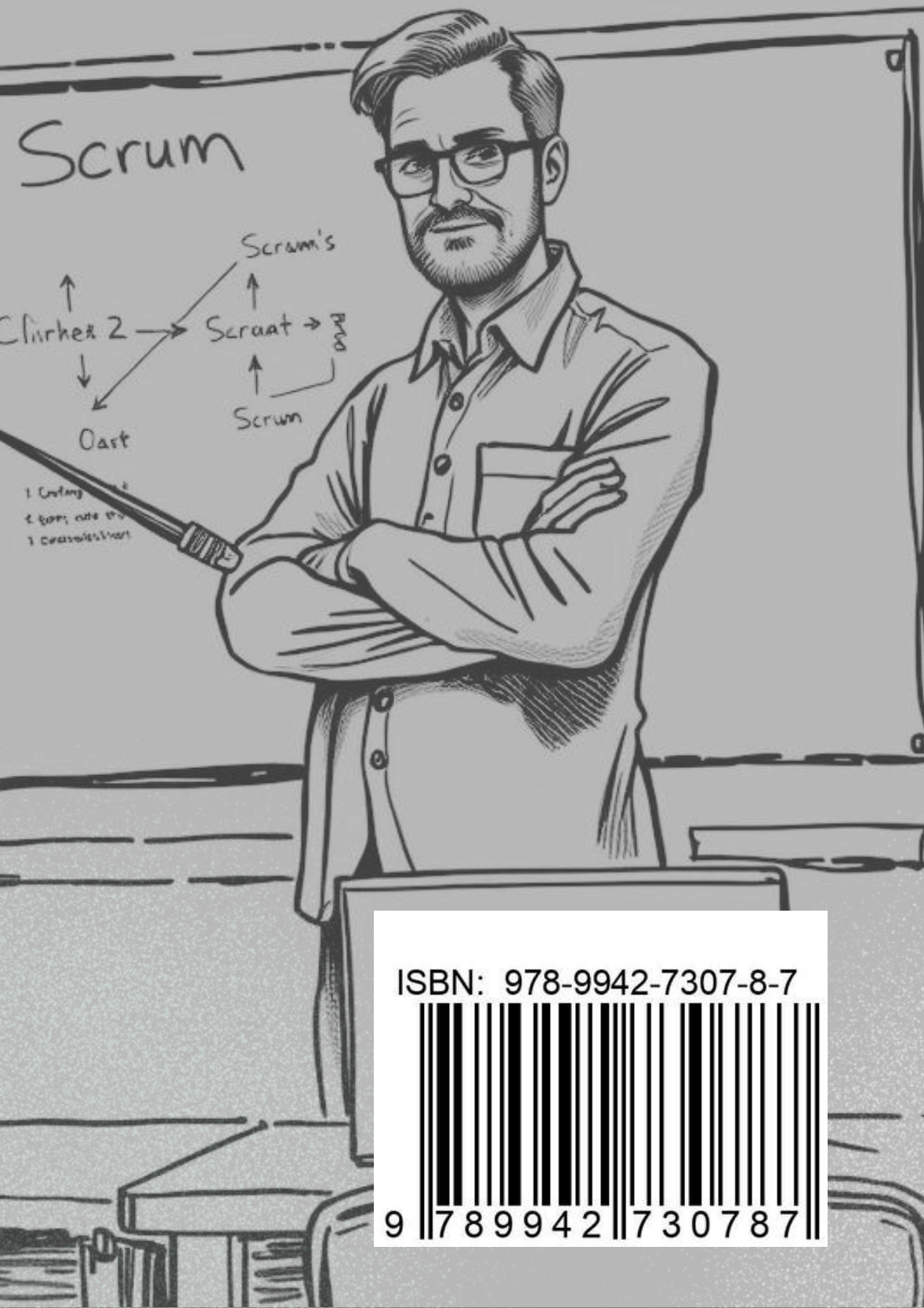
[Último acceso: 10 02 2025].

- [39] atlassian.com, «Retrospectiva de sprint: cómo celebrar una reunión eficaz.,» 2025. [En línea]. Available: <https://www.atlassian.com/es/team-playbook/plays/retrospective#instructions>. [Último acceso: 10 02 2025].
- [40] S. V. Rooden, «Product Backlog Refinement Explained,» 2024. [En línea]. [Último acceso: 10 02 2025].
- [41] miro.com, «Artefactos Scrum: ¿qué son? ¿Para qué sirven? Mira ejemplos.,» 2025. [En línea]. Available: <https://miro.com/es/agile/que-son-artefactos-scrum/>. [Último acceso: 10 02 2025].
- [42] softwarecamp.mx, «Administración Ágil de Proyectos ¿Qué es SCRUM y por qué funciona?,» 2025. [En línea]. Available: <http://softwarecamp.mx/capacitacion/proyectos-atrasados-o-fuera-de-presupuesto-porque-funciona-scrum/>. [Último acceso: 10 02 2025].
- [43] Meet Jeff Sutherland y Meet Ken Schwaber, «Scrum Guides,» [En línea]. Available: <https://scrumguides.org/>. [Último acceso: 10 02 2025].
- [44] Ken Schwaber y Jeff Sutherland, «2020-Scrum-Guide-Spanish-Latin-South-American,» 2020. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>. [Último acceso: 10 02 2025].
- [45] Ward Cunningham, «Manifiesto for Agile Software Development,» 2025. [En línea]. Available: <https://agilemanifesto.org/>. [Último acceso: 10 02 2025].
- [46] M. A. e. al., «Historias de Usuario Ingeniería de Requisitos Ágil Imagen de cubierta,» 2022. [En línea].
- [47] consensus.app, «Digital Collections Are a Sprint, Not a Marathon: Adapting Scrum Project Management Techniques to Library Digital Initiatives,» 2025. [En línea]. Available: <https://consensus.app/papers/digital-collections-are-a-sprint-not-a-marathon-adapting-dulock-long/880foae4649a52989583564365c8548b/>. [Último acceso: 10 02 2025].

- [48] Z. Wang, «Comparisons on Scrum Team Strategies: A multi-agent Simulation,» 2th International Conference on Computer Modeling and Simulation, 2020. [En línea]. Available: <http://10.1145/3408066.3408087>. [Último acceso: 10 02 2025].
- [49] consensus.app, «A Decision Support System for Sprint Planning in Scrum Practice - Consensus,» consensus.app, 2025. [En línea]. Available: <https://consensus.app/papers/a-decision-support-system-for-sprint-planning-in-scrum-alhazmi-huang/de15333bcc5550eb8279c27498bbf250/>. [Último acceso: 10 02 2025].
- [50] G. Rodríguez, I. Gasparini, A. Kemczinski y A. Veloso de Matos, «Students' Perception of Scrum in a Course Project,» IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 2021. [En línea]. Available: <https://10.1109/RITA.2021.3136436>.. [Último acceso: 10 02 2025].
- [51] Erdoğan, «More effective sprint retrospective with statistical analysis,» Journal of Software: Evolution and Process - Wiley Online Library, 2018. [En línea]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/smr.1933>. [Último acceso: 10 02 2025].
- [52] K. Model y G. Herzwurm, «Software-Supported Product Backlog Prioritization in Scrum Software Development Projects,» 2025. [En línea].
- [53] I. C. Publication, «A Deep Dive into Product Backlog Prioritization Technique in Scrum,» IEEE Conference Publication, 2025. [En línea]. Available: <https://ieeexplore.ieee.org/document/10430672>. [Último acceso: 10 02 2025].
- [54] A. E. Babiker, A. Mahmoud y A. Abdalrahman, «Sprint Backlog Estimating and Planning Using Planning Poker Technique in Agile Scrum Framework,» International Journal of Advanced Research in Computer Science and Software Engineering, 2018. [En línea]. Available: <https://10.23956/IJARCSSE.V8I5.686>.. [Último acceso: 10 02 2024].
- [55] M. Fu y C. Tantithamthavorn, «GPT2SP: A Transformer-Based Agile Story Point Estimation Approach,» IEEE Transactions on Software Engineering,

2023. [En línea]. Available: <http://10.1109/TSE.2022.3158252>.. [Último acceso: 10 02 2025].

# SCRUM: GUÍA DIDÁCTICA - PRÁCTICA PARA EL AULA Y EL LABORATORIO



 **ALUMNI  
EDITORIA  
2025**

**PRIMERA EDICIÓN**

ISBN: 978-9942-7307-8-7



9 789942 730787