



Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Doctorado

**Adaptación de un modelo de comunicación
al marco de trabajo Scrum
en entornos de Desarrollo de Software**

presentada por

M.T.I. Juan Miguel Hernández Bravo

como requisito para la obtención del grado de

Doctor en Ciencias de la Computación

Directora de tesis

Dra. Blanca Dina Valenzuela Robles

Cuernavaca, Morelos, México. Febrero de 2026.

 Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL		Código: CENIDET-AC-006-D20
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4		Revisión: 0
			Página 1 de 1

Cuernavaca, Mor., a 28 de enero de 2026

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
P R E S E N T E

AT'n: DRA. ANDREA MAGADÁN SALAZAR
PRESIDENTA DEL CLAUSTRO DOCTORAL DEL
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial del estudiante **M.C. JUAN MIGUEL HERNÁNDEZ BRAVO** manifiestan que después de haber revisado el documento de tesis titulado **"Adaptación de un modelo de comunicación al marco de trabajo scrum en entornos de desarrollo de software"**, realizado bajo la dirección de la **Dra. Blanca Dina Valenzuela Robles**, el trabajo se **ACEPTA** para proceder a su impresión.

A T E N T A M E N T E

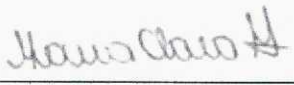
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"


DRA. BLANCA DINA VALENZUELA ROBLES
TECNM/CENIDET


DR. RENÉ SANTAOLAYA SALGADO
TECNM/CENIDET


DR. JUAN GABRIEL GONZÁLEZ SERNA
TECNM/CENIDET


DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
TECNM/CENIDET


DRA. MARÍA CLARA GÓMEZ ÁLVAREZ
UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN

c.c.p: C Verónica Sotelo Boyas/ Jefa del Departamento de Servicios Escolares
c.c.p: C Nôe Alejandro Castro Sánchez / Jefe del Departamento de Ciencias Computacionales
c.c.p: Expediente



Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca Mor, 29/enero/2026

Oficio No. SAC/034/2026

Asunto: Autorización de impresión de tesis

JUAN MIGUEL HERNÁNDEZ BRAVO
CANDIDATO AL GRADO DE DOCTOR EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Adaptación de un modelo de comunicación al marco de trabajo scrum en entornos de desarrollo de software"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

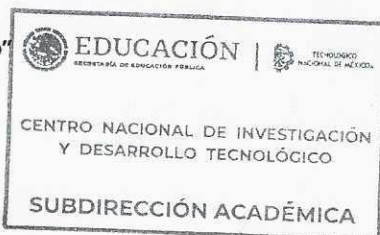
Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®

"Conocimiento y Tecnología al servicio de México"

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO



c.c.p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz



2026
año de
Margarita
Maza

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico



Dedicatoria

A mi esposa Ana Elsa, gracias por todo el tiempo que no pude dedicarte y por tu apoyo incondicional durante esta etapa del doctorado. Tu paciencia, comprensión y amor fueron fundamentales para que pudiera seguir adelante. Te amo.

A mis hijos Annel, Iván y Ángel, y a mis nietos Gael, Juanito e IvanBe, gracias por su cariño y apoyo incondicional. Los amo con todo mi corazón.

A mis padres, Paula (†) y Pedro (†), por su gran amor, fortaleza y sacrificio para sacar adelante a mis diez hermanos y a mí. Gracias por todas sus enseñanzas y valores, estaré eternamente agradecido. Este agradecimiento va hasta el cielo; los llevo siempre en mi mente y en mi corazón, y los seguiré amando por siempre. Cuán orgullosa estarías mamá.

A mis hermanos Alfredo, Hugo, Pedro, Marlene, Maribel, Elisabeth, Alejandra, Jorge, Steve y Esmir, gracias por su apoyo, confianza y por estar presentes incluso en los momentos más difíciles. Su respaldo ha sido muy importante para seguir adelante.

Agradecimientos

Agradezco primeramente a Dios por darme la fuerza, la paciencia y la perseverancia necesarias para concluir esta etapa profesional de mi vida. Su presencia ha sido constante en cada momento de mi vida.

A mis asesores, la Dra. Blanca Dina Valenzuela Robles, al Dr. René Santaolaya Salgado, al Dr. Juan Gabriel González Serna, al Dr. Noé Alejandro Castro Sánchez y a la Dra. María Clara Gómez Álvarez, por compartir sus conocimientos, orientación y experiencia, así como por su paciencia y compromiso durante el desarrollo de este trabajo. En particular, agradezco a la Dra. Blanca y al Dr. René por su acompañamiento constante a lo largo de todo el doctorado y por el ánimo y apoyo permanente que me brindaron en cada etapa de este proceso, estoy muy agradecido.

A todos los docentes que, con dedicación y vocación, aportaron de manera importante a mi formación profesional, compartiendo sus conocimientos y experiencia, y dejando una huella significativa en mi desarrollo académico y personal.

A mis compañeros y a los nuevos amigos del área de computación, por el apoyo, la convivencia y los aprendizajes compartidos a lo largo de este camino, que hicieron más llevadera esta etapa; en especial a Liz, por su amistad, apoyo y buenos consejos.

A la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), anteriormente Consejo Nacional de Ciencia y Tecnología (CONACYT) y Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), por la beca económica otorgada para la realización de estos estudios; al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por la formación académica brindada; y al Instituto Tecnológico de Acapulco, por el otorgamiento del permiso de beca comisión.

Resumen

Con la finalidad de optimizar la efectividad comunicativa en proyectos de desarrollo de software, se adaptó el modelo SMCR (*Source–Message–Channel–Receiver*) propuesto por David Berlo para que pudiera ser utilizado en el marco ágil Scrum. Esto aplica a los grupos colocalizados y a los grupos distribuidos (Desarrollo Global de Software, DGS). En particular en los entornos distribuidos, donde persisten retos como la falta de interacción cara a cara, las diferencias horarias y culturales. Se ha determinado que una comunicación deficiente es uno de los motivos más relevantes del incumplimiento de requisitos, la baja calidad y los retrasos en proyectos gestionados con Scrum.

El propósito principal de este proyecto de tesis es promover una comunicación efectiva entre los integrantes del proyecto Scrum, adaptando el modelo de Berlo para garantizar que los mensajes generados en cada ceremonia Scrum sean recibidos, comprendidos y retroalimentados. Los objetivos concretos incluyen: garantizar la recepción de mensajes por medio de herramientas digitales, verificar su entendimiento y proporcionar retroalimentación a tiempo, teniendo como indicador que al menos el 90 % de las historias de usuario sean cumplidas por cada Sprint.

La investigación se llevó a cabo siguiendo un enfoque que consta de tres etapas: marco analítico, desarrollo del modelo y evaluación. Tras revisar el estado del arte, se diseñó y puso en marcha un modelo de comunicación adaptado a Scrum, respaldado por la aplicación web Comunica-Scrum.

La evaluación se llevó a cabo a través de la validación por parte de expertos, la implementación con profesionales nacionales e internacionales y un estudio de caso real. Se demostró que el modelo es fiable y consistente. Además, mostró una mejora importante en la interpretación de los mensajes, una disminución de los malentendidos y una alineación del equipo con las metas del proyecto.

Los resultados corroboran que, al adaptar el modelo de Berlo al marco Scrum y añadirle una herramienta tecnológica concreta, se logra una solución efectiva, escalable y replicable para mejorar la comunicación en los proyectos de desarrollo de software ágil. Esto contribuye a que los equipos Scrum tengan éxito tanto en entornos presenciales como distribuidos.

Palabras clave: comunicación efectiva, Scrum, desarrollo global de software, modelo de Berlo, equipos ágiles distribuidos, equipos de desarrollo de software colocalizados.

Abstract

With the aim of optimizing communication effectiveness in software development projects, the SMCR (Source–Message–Channel–Receiver) model proposed by David Berlo was adapted so that it could be applied within the agile Scrum framework. This adaptation applies to both co-located teams and distributed teams operating in Global Software Development (GSD) environments. Particularly in distributed settings, where challenges such as the lack of face-to-face interaction and differences in time zones and cultures persist. It has been determined that poor communication is one of the most significant causes of requirements non-compliance, low quality, and delays in Scrum-managed projects.

The main purpose of this research is to promote effective communication among Scrum project members by adapting Berlo's model to ensure that messages generated in each Scrum ceremony are received, understood, and properly fed back. The specific objectives include ensuring message reception through digital tools, verifying their understanding, and providing timely feedback, with the indicator that at least 90% of user stories are completed in each Sprint.

The research was conducted following a three-stage approach: analytical framework, model development, and model evaluation. After reviewing the state of the art, a communication model adapted to Scrum was designed and implemented, supported by the web application Comunica-Scrum.

The evaluation was carried out through expert validation, implementation with national and international professionals, and a real-world case study. The results demonstrated that the model is reliable and consistent. In addition, they showed a significant improvement in message interpretation, a reduction in misunderstandings, and stronger team alignment with project goals.

The results confirm that adapting Berlo's model to the Scrum framework and complementing it with a dedicated technological tool provides an effective, scalable, and replicable solution to improve communication in agile software development projects. This contributes to the success of Scrum teams in both co-located and distributed environments.

Keywords: effective communication, Scrum, global software development, Berlo's model, distributed agile teams, co-located software development teams.

1 Tabla de Contenido

Índice de figuras.....	4
Índice de tablas.....	8
Capítulo 1 Introducción.....	10
1.1 Planteamiento del problema.....	12
1.2 Objetivos	12
1.3 Justificación.....	13
1.4 Alcances y limitaciones	14
Capítulo 2 Marco Teórico	16
2.1 Metodologías de Desarrollo de Software Ágiles.....	16
2.2 Marco de trabajo Scrum	18
2.1 Desarrollo de Software Distribuido o Global.....	20
2.2 Administración de Proyectos.....	22
2.3 Comunicación efectiva	23
2.4 Modelos de comunicación.....	25
2.5 Método Delphi	36
2.6 V de Aiken	38
2.7 Alfa de Cronbach	39
Capítulo 3 Estado del arte	41
3.1 Introducción.....	41
3.2 Problemas de comunicación en equipos colocizados (on-site)	42
3.3 Comunicación en el Desarrollo de Software, en Equipos Scrum Distribuidos	44

3.4	Desafíos en la Comunicación en el Desarrollo Global de Software y en Entornos Ágiles Distribuidos	47
3.5	Modelos y marcos existentes	48
3.6	Revisión de herramientas comerciales y gratuitas empleadas en la gestión de proyectos de software.....	50
Capítulo 4	Metodología de la solución.....	52
4.1	Análisis del modelo	55
4.2	Diseño del Modelo.....	60
4.3	Implementación del Modelo	66
4.4	Diseño y desarrollo de Comunica-Scrum.....	72
4.5	Evaluación del Modelo	75
4.5.1	Validación del instrumento mediante el método Delphi (por expertos) 75	
4.5.2	Aplicación del instrumento a profesionales en desarrollo de software bajo Scrum	77
4.5.3	Validación Empírica del Modelo.....	81
Capítulo 5	Formalización del Modelo.....	88
Capítulo 6	Resultados	93
Capítulo 7	Conclusiones, Aportaciones y trabajos futuros.....	96
Referencias		99
Capítulo 8	Anexos	109
	Anexo A Divulgación del Proyecto de Investigación	109
8.1	Presentación del Artículo en un Congreso Internacional.....	109
8.2	Artículo JCR (Emerging Sources Citation Index) y Revista CONACYT .	113

8.3 Ponencia en el Instituto Tecnológico de Zacatepec	116
Anexo B Adaptación del Modelo de Comunicación de los Eventos Scrum	117
Anexo C Definición de términos que deben conocer todos los involucrados del proyecto	131
Anexo D Aplicación Comunica-Scrum	134
<i>Gestión de proyectos Scrum</i>	134
<i>Ceremonias Scrum</i>	138
Anexo E Preguntas de la encuesta aplicada al panel de expertos.	144
Anexo F Clases del diseño de Comunica-Scrum.....	146

Índice de figuras

Figura 2.1. Marco de trabajo Scrum	21
Figura 2.2. Modelo Aristotélico	25
Figura 2.3. Modelo de Lasswell.....	26
Figura 2.4. Modelo de Shannon y Weaver	28
Figura 2.5. Modelo de Berlo	29
Figura 2.6. Modelo de Jakobson	30
Figura 2.7. Modelo de Schramm	32
Figura 2.8. Modelo de Braddock	32
Figura 2.9. Modelo de Maletzke	34
Figura 3.1. Esquema conceptual de las fallas en la comunicación	46
Figura 4.1. Metodología de Desarrollo	52
Figura 4.2. Problemas de un proyecto de desarrollo de software que afectan el cumplimiento de los requerimientos iniciales	57
Figura 4.3. Problemas en la comunicación en el desarrollo de Proyectos Scrum en un entorno distribuido, DGS	58
Figura 4.4. Reconocimiento de las ceremonias de Scrum donde interviene la comunicación	59
Figura 4.5. Adaptación del modelo comunicativo propuesto por Berlo.....	60
Figura 4.6. Propuesta de solución.....	64
Figura 4.7. Adaptación del modelo de comunicación (Reunión de Revisión del Sprint).....	64
Figura 4.8. Ejemplo práctico de una plantilla generada para la Ceremonia de Revisión del Sprint	68

Figura 4.9. Ejemplo de una plantilla de Retroalimentación de la Ceremonia de Revisión del Sprint	70
Figura 4.10. Gráfica Burndown, esfuerzo real VS planeado en el Sprint	72
Figura 4.11. Modelo de clases aplicado a la gestión de proyectos con Scrum	73
Figura 4.12. Diagrama de clases que representa la comunicación entre emisor y receptor	74
Figura 4.13. Valor V de Aiken de la redacción de las preguntas, 1ª vuelta	76
Figura 4.14. Valor V de Aiken de la pertinencia de las preguntas	77
Figura 4.15 Valor V de Aiken de la redacción de las preguntas, 2a vuelta	77
Figura 4.16. Alfa de Cronbach, equipo Scrum al excluir cada ítem individualmente	85
Figura 5.1. Diagrama de Máquina de Estados del flujo de un Historia de Usuario	89
Figura 5.2. ADF que representa el flujo del ciclo de vida de una HU en Scrum	91
Figura 8.1. Página de la Portada Memorias del Congreso	109
Figura 8.2. Página de índice de la Memoria del Congreso 1/2.....	110
Figura 8.3. Página de índice de la Memoria del Congreso 2/2.....	111
Figura 8.4. Página Inicial del Artículo	112
Figura 8.5. Portada de la Revista IJCOPI	114
Figura 8.6. Página de índice de la Revista IJCOPI Vol. 16 No. 4 (2025)	115
Figura 8.7. Página Inicial del Artículo IJCOPI.....	115
Figura 8.8. Ponencia en el IT Zacatepec.....	116
Figura 8.9. Adaptación del modelo comunicativo (Reunión de Refinamiento del Product Backlog)	117

Figura 8.10. Ejemplo Mensaje de la Reunión de Refinamiento del Product Backlog	118
Figura 8.11. Ejemplo de Mensaje de la Retroalimentación Reunión de Refinamiento del Product Backlog	119
Figura 8.12. Adaptación del modelo de comunicación (Reunión de planificación del Sprint).....	120
Figura 8.13. Ejemplo de una plantilla de la Ceremonia de Planeación del Sprint	121
Figura 8.14. Ejemplo de una plantilla de la Ceremonia de Planeación del Sprint, HU Divididas.....	121
Figura 8.15. Ejemplo de un Plantilla de Retroalimentación de la Ceremonia de Planeación del Sprint	123
Figura 8.16. Adaptación del modelo de comunicación (Reunión Diaria)	124
Figura 8.17. Ejemplo de una Plantilla de la Reunión Diaria	124
Figura 8.18. Adaptación del modelo de comunicación (Ejecución del Sprint)	125
Figura 8.19. Ejemplo de una Plantilla de la Ejecución del Sprint.....	126
Figura 8.20. Adaptación del modelo de comunicación (Retrospectiva del Sprint).	127
Figura 8.21. Ejemplo de una Plantilla de la Ceremonia de Retrospectiva del Sprint	128
Figura 8.22. Ejemplo de un Plantilla de Retroalimentación de la Ceremonia de Retrospectiva del Sprint	129
Figura 8.23. Pantalla de inicio se sesión	134
Figura 8.24. Formulario de registros de usuarios	134
Figura 8.25. Formulario de proyectos.....	135
Figura 8.26. Listado de proyectos	135

Figura 8.27. Registro de historia de usuario.....	136
Figura 8.28. Listado de historias de usuario.....	136
Figura 8.29. Creación de tarea para una historia de usuario	137
Figura 8.30. Registro de sprint	137
Figura 8.31. Listado de Sprints.....	138
Figura 8.32. Ceremonias	138
Figura 8.33. Reunión de planeación de sprint sin registros.....	139
Figura 8.34. Listado de Sprints para reunión de planeación del sprint.....	139
Figura 8.35. Formulario para agendar reunión de planeación de Sprint	139
Figura 8.36. Listado de reuniones de planeación del Sprint.....	140
Figura 8.37. Listado de asistentes para reunión de planeación del Sprint	140
Figura 8.38. Listado de historias de usuario para reunión de planeación del Sprint	140
Figura 8.39. Plantilla de reunión de planeación del sprint	141
Figura 8.40. Reunión de planeación del sprint con comentarios	142
Figura 8.41. Envío del mensaje para reunión de planeación del Sprint	142
Figura 8.42. Mensajes de retroalimentación para reunión de planeación del Sprint	143
Figura 8.43. Respuesta del Product Owner	143
Figura 8.44. Clases para a la gestión de proyectos con Scrum	146
Figura 8.45. Clases que representa la comunicación entre emisor y receptor ...	147

Índice de tablas

.....	
Tabla 2.1 Los principios de los métodos ágiles	18
Tabla 2.2 Características comparativas de los modelos de comunicación	35
Tabla 3.1 Comparativa de modelos y marcos de comunicación en equipos Scrum distribuidos	48
Tabla 3.2 Cuadro comparativo de herramientas similares	51
Tabla 4.1 Comparativo Modelo de Berlo y Modelo de Comunicación Adaptado (Reunión de Revisión del Sprint).....	65
Tabla 4.2 Correlación entre los Componentes de Berlo y las Ceremonias Scrum	66
Tabla 4.3 Descripción de la plantilla de la Ceremonia de Revisión del Sprint	69
Tabla 4.4 Descripción de la plantilla de la Retroalimentación de la Ceremonia de Revisión del Sprint	71
Tabla 4.5 Clases para la administración de proyectos Scrum.....	73
Tabla 4.6 Clases encargadas de gestionar la comunicación entre emisores y receptores	74
Tabla 4.7 Preguntas para Valorar la Implementación del Modelo de Comunicación en Entornos Scrum.....	79
Tabla 4.8 Estadísticos descriptivos de la encuesta a profesionales en desarrollo de software con Scrum.....	80
Tabla 4.9 Preguntas al equipo Scrum para validar el modelo de comunicación propuesto	83
Tabla 4.10 Estadísticos descriptivos de la encuesta al equipo Scrum	86
Tabla 5.1 Tabla de transiciones de un AFD	91
Tabla 5.2 Correspondencia entre los estados y el flujo de una HU	92

Tabla 8.1 Descripción del Mensaje de la Reunión de Refinamiento del Product Backlog	118
Tabla 8.2 Atributos del mensaje de retroalimentación generado en la ceremonia de refinamiento del Product Backlog.....	119
Tabla 8.3 Descripción de la Plantilla de la Ceremonia de Planeación del Sprint.	122
Tabla 8.4 Descripción de la Plantilla de la Retroalimentación de la Ceremonia de Planificación del Sprint.....	123
Tabla 8.5 Descripción del Plantilla de la Reunión Diaria	125
Tabla 8.6 Descripción de la Plantilla de la Ejecución del Sprint	126
Tabla 8.7 Descripción de la Plantilla de la Ceremonia de Retrospectiva del Sprint	128
Tabla 8.8 Descripción de la Plantilla de la Retroalimentación de la Ceremonia de Retrospectiva del Sprint	129

Capítulo 1 Introducción

La manera en la que los equipos de desarrollo de software colaboran y las dinámicas organizacionales se han modificado debido al progreso continuo de la tecnología digital y a la adopción cada vez mayor de modelos laborales flexibles. Las organizaciones han aumentado en los años recientes la contratación de talento especializado fuera de sus fronteras, utilizando la conectividad mundial para disminuir costos operativos, acelerar procesos y tener acceso a una mayor cantidad de capital humano. El uso y la constante actualización de herramientas y aplicaciones que permiten el trabajo en equipo también han sido estimulados por este contexto.

No obstante, estos progresos tecnológicos no erradican los retos propios de la comunicación entre los equipos de desarrollo. La complejidad en el intercambio de información aumenta debido a factores como la variedad de canales digitales, las diferencias culturales, la dispersión geográfica y la asincronía en las interacciones. A pesar de que las herramientas de comunicación contemporáneas hacen posible la disminución parcial de estas brechas, aún existen dificultades sin resolver que afectan la coordinación, la claridad en los mensajes y la eficacia en el desarrollo de proyectos implementados con metodologías ágiles como Scrum [1].

En este escenario, se vuelve indispensable estudiar y adaptar modelos de comunicación que fortalezcan los procesos colaborativos en Scrum, con el fin de mejorar la comprensión, la sincronización y el flujo de información entre los miembros del equipo en entornos de desarrollo de software cada vez más distribuidos y dinámicos.

La entrega tardía del software, el incumplimiento de requisitos y la baja calidad del producto son solo algunas señales comunes en proyectos de software que terminan en fracaso, estos síntomas son causados por problemas de comunicación entre desarrolladores y usuarios, arquitectos y programadores, incluso entre el mismo equipo de programadores. Las metodologías y enfoques ágiles, como Scrum, establecen reuniones breves y diarias entre los miembros del proyecto para favorecer la comunicación; sin embargo, en el desarrollo global de software resulta complejo alcanzar una interacción efectiva debido a la separación geográfica de los equipos [2], falta la interacción presencial entre los diferentes roles, aspecto señalado como esencial en el manifiesto ágil *“el método más efectivo de comunicar la información al equipo de desarrollo y entre sus miembros es la conversación*

cara a cara" [3].

La comunicación y la colaboración constituyen elementos esenciales para el éxito de los proyectos gestionados mediante enfoques ágiles; por ello, las organizaciones deben promover un entorno laboral que favorezca la interacción continua entre los integrantes del equipo [4].

La mala comunicación es una de las principales causas que afectan negativamente los proyectos de Tecnologías de la Información (TI). Según PRAXIS, una consultora mexicana de alcance transnacional especializada en soluciones tecnológicas, el 60% de los proyectos de TI en las organizaciones resultan impactados por este problema [5].

Actualmente, varias organizaciones no otorgan importancia a la comunicación, ni a contar con un personal calificado, lo cual es relevante para el buen desempeño de las empresas. Asimismo, la mayor parte de las dificultades internas, son causa de la mala comunicación de la información que se da entre las áreas/departamentos o el personal interno que deben de trabajar en equipo.

Entre las problemáticas más relevantes derivadas de la falta de una comunicación efectiva se encuentran: (1) el descontento del personal, originado por malentendidos, información incorrecta o inconsistente, lo que genera conflictos tanto entre los involucrados como en los proyectos; (2) la disminución de la productividad, pues los colaboradores se distraen y dedican tiempo a asuntos que no contribuyen al desarrollo del proyecto, y (3) la creación de un clima organizacional negativo debido a diversas inconsistencias que afectan el ambiente laboral y repercuten directamente en los proyectos de TI [5].

Una comunicación efectiva crea un vínculo entre las distintas partes interesadas de un proyecto, integrando entornos culturales y organizacionales diversos, distintos niveles de experiencia y una variedad de perspectivas e intereses relacionados con el desarrollo y los resultados del proyecto.

Es fundamental que durante la ejecución del proyecto se asegure un proceso de comunicación que permita recopilar y distribuir la información necesaria a las personas correctas, en el tiempo establecido y mediante los medios y formatos adecuados. Esto implica aplicar técnicas, conceptos y modelos provenientes de la teoría de la comunicación a las demandas específicas del proyecto, así como identificar las necesidades de información de las partes interesadas (*Stakeholders*): qué información se requiere, quién debe recibirla, de qué manera será presentada y en qué momento se necesitará. En

consecuencia, se requiere un alto nivel de dedicación y habilidad por parte del gerente del proyecto y del equipo, ya que deben reconocer que la comunicación influye directamente en el éxito global del proyecto [6].

1.1 Planteamiento del problema

En algunas ocasiones durante la gestión y desarrollo de proyectos de software, bajo el enfoque Scrum, **se presenta el problema cuando los requerimientos del proyecto no se cumplen completamente conforme a lo planteado inicialmente**. Una causa que origina este problema es la **falta de comunicación efectiva entre las personas involucradas y el equipo de desarrollo**.

Esta deficiencia comunicativa genera malentendidos en la interpretación de los requerimientos, retrasos en la identificación de impedimentos y una alineación insuficiente entre el *Product Owner*, los *Stakeholders* y el equipo de desarrollo, lo que puede derivar en entregas incompletas al final de los *Sprints*, acumulación de deuda técnica y, en última instancia, insatisfacción del cliente o fracaso parcial del proyecto.

1.2 Objetivos

Objetivo general

Favorecer la comunicación entre las personas involucradas en proyectos de software, bajo el marco de trabajo Scrum, **mediante la adaptación de un modelo de comunicación, contribuyendo en el cumplimiento de los requerimientos especificados al inicio del proyecto**.

Objetivos específicos

1. **Asegurar** la recepción efectiva de los mensajes generados en cada una de las ceremonias de Scrum por parte de los integrantes del proyecto, **mediante el uso de herramientas de comunicación**.
2. **Verificar** la correcta comprensión de los mensajes recibidos en cuanto a su estructura y contenido, **utilizando herramientas de comunicación y mecanismos de validación**.
3. **Proporcionar retroalimentación** sobre la comprensión de los mensajes analizados por los participantes involucrados, **mediante herramientas de comunicación**.

Indicador:

Cubrir al menos al 90% las historias de usuarios del proyecto planteado durante cada

Sprint.

1.3 Justificación

Este estudio, orientado a analizar la comunicación efectiva entre las personas y los procesos involucrados en proyectos de desarrollo de software dentro de equipos colocalizados y distribuidos que trabajan con el enfoque ágil Scrum, tiene como finalidad adaptar un modelo de comunicación que permita atender y solucionar problemáticas específicas en determinadas áreas y procesos. La versión adaptada del modelo busca optimizar la interacción y el intercambio de información entre los integrantes del equipo de desarrollo, el *Scrum Master*, el *Product Owner* y los distintos *Stakeholders*, apoyándose en la aplicación propuesta **Comunica-Scrum**.

La implementación de este proyecto generará beneficios significativos tanto para el personal involucrado en el desarrollo de software como para las organizaciones que dependen de aplicaciones para la gestión de proyectos. Asimismo, contribuirá a que los requerimientos establecidos se cumplan conforme a lo previsto desde la fase inicial, fortaleciendo la alineación entre los objetivos del proyecto, la colaboración entre los participantes y la calidad del producto final.

Con el presente trabajo se pretende mejorar la comunicación para que sea rápida, oportuna, fluida, consistente y accesible dentro del marco de trabajo ágil de Scrum. Después de la ejecución de cada una de las actividades o eventos que se manejan en el marco de trabajo ágil de Scrum, se propone una comunicación efectiva con retroalimentación a los participantes comprometidos e involucrados; el manifiesto ágil menciona la comunicación cara a cara, lo cual está bien, pero también debe considerarse que los acuerdos, decisiones y acciones tomadas durante la ejecución de cada uno de los *Sprints* y reuniones establecidas en Scrum debe quedar registrada y ser compartida con el personal involucrado en el proyecto de desarrollo de software, además de que en algún momento dado puede ser que algún integrante del equipo distribuido Scrum no esté disponible en las reuniones, pero sí deben ser partícipes de la información que se generan en éstas.

La información que se generará durante el desarrollo del proyecto, bajo el marco de trabajo Scrum se comunicará a todos los participantes, sin importar que estén en lugares distintos geográficamente y en un idioma determinado, dependiendo de su rol será la información que se les enviará, además la podrán consultar, mediante un módulo de comunicación

colaborativa, el cual será la principal herramienta de comunicación, también se ocuparán el correo electrónico (formal) y el WhatsApp (informal), estos dos últimos medios servirán como medios de comunicación adicionales para garantizar que les llegue la información a los participantes del proyecto, ya que el correo electrónico es una herramienta que se consulta diariamente, no interrumpe el trabajo, es rápido de consultar y es asíncrono; por otro lado el WhatsApp es una aplicación móvil que está instalada prácticamente en la mayoría de los celulares y la cual se consulta en todo momento.

1.4 Alcances y limitaciones

Alcances

- Adaptar un modelo de comunicación al marco ágil Scrum con el propósito de asegurar una interacción efectiva entre los desarrolladores, el *Scrum Master*, el *Product Owner* y los diversos *Stakeholders*.
- Desarrollar una aplicación tecnológica que integre las actividades, prácticas, roles y artefactos establecidos en Scrum, y que permita comunicar las acciones y decisiones del proyecto mediante un módulo de comunicación colaborativa, así como por medio de canales adicionales como correo electrónico y WhatsApp.
- Aplicar un estudio de caso mediante el cual se obtendrán los resultados necesarios para evaluar y validar el modelo de comunicación propuesto.
- Abordar y mitigar problemas específicos de comunicación tales como: falta de contacto frecuente cara a cara, deficiencias en la comunicación de requisitos y dificultades asociadas a diferencias de horarios, zonas geográficas e idiomas.

Limitaciones

- El modelo de comunicación propuesto, así como la aplicación desarrollada para su evaluación, se encuentran orientados exclusivamente a la gestión de proyectos bajo el marco ágil Scrum, sin abarcar otros enfoques o metodologías de desarrollo de software.
- La calidad de los productos entregados no depende únicamente de una comunicación efectiva; existen otros factores determinantes en la calidad del software. Por lo tanto, este proyecto se enfoca únicamente en la mejora desde la perspectiva comunicacional.

-
- Algunos problemas de comunicación no serán abordados por este estudio, tales como la falta de confianza, la falta de compromiso del equipo y las diferencias culturales entre los participantes.

Capítulo 2 Marco Teórico

En este capítulo se incorporan los conceptos y teorías que respaldan la investigación, iniciando con la historia y los principios de las metodologías ágiles, el marco de trabajo Scrum y su estructura basada en roles, eventos y artefactos; para luego hablar del contexto del Desarrollo Global de Software (DGS) y los desafíos que enfrentan los equipos distribuidos, como problemas de comunicación, coordinación y diferencias culturales y horarias, que impactan en el cumplimiento de los requisitos, la calidad y los plazos de entrega. A partir de ello, se define la comunicación efectiva y su importancia en proyectos Scrum distribuidos, mejorando la comprensión, la retroalimentación y el uso de canales y herramientas colaborativas.

Además, se revisan modelos clásicos de comunicación (Aristóteles, Lasswell, Shannon y Weaver, Berlo, Jakobson, Schramm, Braddock y Maletzke) para identificar elementos comunes y componentes ampliados (codificación/decodificación, canal, ruido, intención, retroalimentación) que son relevantes para analizarlos y adaptarlos en esta investigación. Finalmente, se exponen los métodos e índices estadísticos que permitirán evaluar los instrumentos: el método Delphi y la V de Aiken para validar por juicio de expertos el contenido del cuestionario y asegurar la pertinencia y claridad de sus ítems; y el Alfa de Cronbach para medir la confiabilidad de los cuestionarios y determinar la consistencia interna de las escalas utilizadas, garantizando así la validez y confiabilidad de los instrumentos de recolección de datos para respaldar los resultados de la investigación.

2.1 Metodologías de Desarrollo de Software Ágiles

En el año 2001, Kent Beck y otros dieciséis destacados desarrolladores de software, escritores y consultores —grupo conocido como la *Alianza Ágil*— firmaron el *Manifiesto por el Desarrollo Ágil de Software*. En dicho manifiesto se establece que:

“Estamos descubriendo mejores formas de desarrollar software haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar:

- A los individuos y sus interacciones por encima de los procesos y las herramientas.
- El software funcionando, por encima de la documentación exhaustiva.

-
- La colaboración con el cliente, por encima de la negociación contractual.
 - Responder al cambio, por encima de seguir un plan”.

Si bien los elementos situados en segundo término también poseen valor, el manifiesto enfatiza que se otorga mayor importancia a los elementos mencionados en primer lugar.

Aunque las ideas fundamentales que sustentan el manifiesto ya estaban presentes desde años previos entre los desarrolladores de software, fue en las últimas dos décadas cuando se consolidaron en un verdadero *movimiento*. Las metodologías ágiles surgieron con el propósito de superar debilidades reales y ampliamente reconocidas de los enfoques tradicionales de ingeniería de software. El desarrollo ágil ofrece ventajas significativas; sin embargo, no todos los proyectos, productos, equipos o contextos son adecuados para su adopción [7].

En la actualidad, la agilidad se ha convertido en un concepto central para describir los procesos modernos de desarrollo de software. Un equipo ágil es eficiente, adaptable y capaz de responder oportunamente al cambio. El cambio es inherente al software: puede originarse en los requerimientos del sistema, en el equipo de trabajo, en la adopción de nuevas tecnologías o en factores externos que afectan al proyecto. Por ello, los procesos deben apoyar el cambio de manera continua. Un equipo ágil reconoce que el software es construido por personas que colaboran entre sí, y que su capacidad para comunicarse y trabajar en conjunto es esencial para el éxito [8].

Desde la perspectiva de Jacobson, la agilidad encuentra su principal justificación en la constante evolución de los requerimientos del software.

Si bien las diversas metodologías ágiles se fundamentan en el desarrollo iterativo e incremental de funcionalidades, cada una de ellas establece mecanismos, roles, artefactos y prácticas específicas para llevar a cabo dicho proceso, en función del contexto y las necesidades del proyecto. Estas diferencias se reflejan, por ejemplo, en la forma de planificar el trabajo, gestionar los requisitos, coordinar a los equipos o evaluar los avances obtenidos en cada iteración. Sin embargo, a pesar de estas particularidades operativas, todas las metodologías ágiles convergen en un marco conceptual común.

Las características comunes derivadas de estos principios constituyen la base para analizar y comparar las distintas metodologías ágiles, así como para comprender su impacto en los

procesos de comunicación, coordinación y toma de decisiones dentro de los equipos de trabajo. En la Tabla 2.1 se presentan dichos principios fundamentales, los cuales han sido tomados de [9] y sirven como referencia para el análisis desarrollado en este capítulo.

Tabla 2.1 Los principios de los métodos ágiles

Principio	Descripción
Participación del cliente	Los clientes deben intervenir estrechamente durante el proceso de desarrollo. Su función consiste en ofrecer y priorizar nuevos requerimientos del sistema y evaluar las iteraciones de este.
Entrega incremental	El software se desarrolla en incrementos y el cliente especifica los requerimientos que se van a incluir en cada incremento.
Personas, no procesos	Tienen que reconocerse y aprovecharse las habilidades del equipo de desarrollo. Debe permitirse a los miembros del equipo desarrollar sus propias formas de trabajar sin procesos establecidos.
Adoptar el cambio	Esperar a que cambien los requerimientos del sistema y, de este modo, diseñar el sistema para adaptar dichos cambios.
Mantener simplicidad	Enfocarse en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Siempre que sea posible, trabajar de manera activa para eliminar la complejidad del sistema.

2.2 Marco de trabajo Scrum

Scrum es un marco ágil diseñado para apoyar la solución de problemas complejos en la construcción de productos, ofreciendo gran flexibilidad sin sacrificar la productividad del proceso [10].

Scrum se basa en la idea de que el saber se crea a partir de la experiencia anterior adquirida durante el desarrollo de proyectos pasados y del progreso efectivo de las tareas planeadas en cada Sprint. Esto significa que se toman decisiones basándose en datos y resultados comprobables. Por lo tanto, Scrum utiliza una metodología incremental e iterativa que tiene como objetivo optimizar la previsibilidad y sostener un control apropiado sobre los riesgos del proyecto. Para hacer precisamente eso, este marco se basa en tres pilares:

1. Transparencia

Todos los elementos significativos del proceso deben ser transparentes para quienes tienen la responsabilidad sobre los resultados. Esto permite que cada miembro del equipo

identifique el estado de avance de las actividades y comprenda los pasos necesarios para alcanzar los objetivos establecidos.

2. Inspección

Los equipos que utilizan Scrum deben revisar de manera periódica los artefactos y el progreso hacia los Objetivos del Sprint, con el fin de detectar desviaciones no deseadas. Estas inspecciones no deben interferir con el flujo de trabajo.

3. Adaptación

Si durante la inspección se identifican variaciones que puedan comprometer el producto final, se deben realizar los ajustes necesarios de manera inmediata para corregir el rumbo del proyecto. Estos cambios deben implementarse tan pronto como sea posible para evitar desviaciones adicionales.

Debido a estos pilares, Scrum permite que las etapas del proyecto avancen sin depender estrictamente de la finalización completa de fases anteriores. La visibilidad del progreso en tiempo real facilita identificar necesidades de corrección y reduce el tiempo y los recursos necesarios para completar el proyecto.

Scrum define un conjunto de roles, prácticas y eventos, que sirven como base para estructurar el proceso de trabajo dentro de un equipo o proyecto [11].

Roles principales en Scrum

- **Scrum Master:** facilita la adopción del marco, elimina impedimentos y promueve la mejora continua.
- **Product Owner:** representa a los *Stakeholders* y es responsable de gestionar el *Product Backlog*.
- **Equipo de desarrollo:** grupo autoorganizado encargado de construir el incremento del producto.

Estructura del Sprint

El Sprint puede durar entre una y cuatro semanas, según lo defina el equipo, y debe mantenerse lo más corto posible para favorecer la retroalimentación y la adaptación. Durante este periodo se produce un incremento del sistema que puede ser utilizado de manera potencial.

Las funcionalidades que se desarrollan en cada *Sprint* provienen del *Product Backlog*, una

lista priorizada de requerimientos de alto nivel (*Product Backlog Items*, PBI). Los elementos que serán incluidos en el *Sprint* se seleccionan durante la Reunión de Planificación del *Sprint*. En esta sesión:

1. El *Product Owner* presenta los PBIs que desea ver implementados.
2. El equipo discute con el *Product Owner* para determinar qué cantidad de trabajo puede comprometerse a realizar en el *Sprint* siguiente.

Una vez iniciado el *Sprint*, el *Sprint Backlog* (lista de tareas comprometidas) no puede modificarse, lo que implica que los requerimientos permanecen estables durante ese ciclo de desarrollo.

Scrum promueve la formación de equipos autoorganizados y fomenta la comunicación verbal entre todos los integrantes del equipo y los involucrados en el proyecto. Idealmente, todos los miembros del equipo deberían compartir un mismo espacio físico, tal como se muestra en la Figura 2.1.

2.1 Desarrollo de Software Distribuido o Global

El desarrollo de software global (DGS) (*Global Software Development*, GSD) se refiere a la creación de software por parte de equipos ubicados en diferentes regiones geográficas. Estos equipos pueden pertenecer a la misma organización o a distintas organizaciones. La distribución geográfica puede variar desde edificios separados dentro de la misma ciudad hasta ubicaciones situadas en diferentes continentes [12].

La motivación inicial del DGS fue la reducción de los costos laborales, siendo India uno de los primeros países en beneficiarse de este enfoque; no obstante, con el tiempo esta práctica se ha extendido ampliamente debido a otras motivaciones relevantes, como la necesidad de desarrollar sistemas complejos que requieren personal con conocimientos especializados y diversas habilidades, la posibilidad de realizar desarrollo continuo durante las 24 horas del día aprovechando las diferencias horarias, y la conveniencia de mantener una mayor cercanía con los clientes o usuarios finales en mercados específicos.

Retos del Desarrollo de Software Global

El DGS representa una fuente importante de desafíos para la ingeniería de software. Entre los más relevantes se encuentran los siguientes [12]:

Procesos. Los modelos de proceso tradicionales están diseñados generalmente para una sola organización e incluyen esquemas de subcontratación. Sin embargo, no contemplan mecanismos suficientemente robustos para organizar la colaboración y la coordinación entre múltiples organizaciones que trabajan en un mismo proyecto.

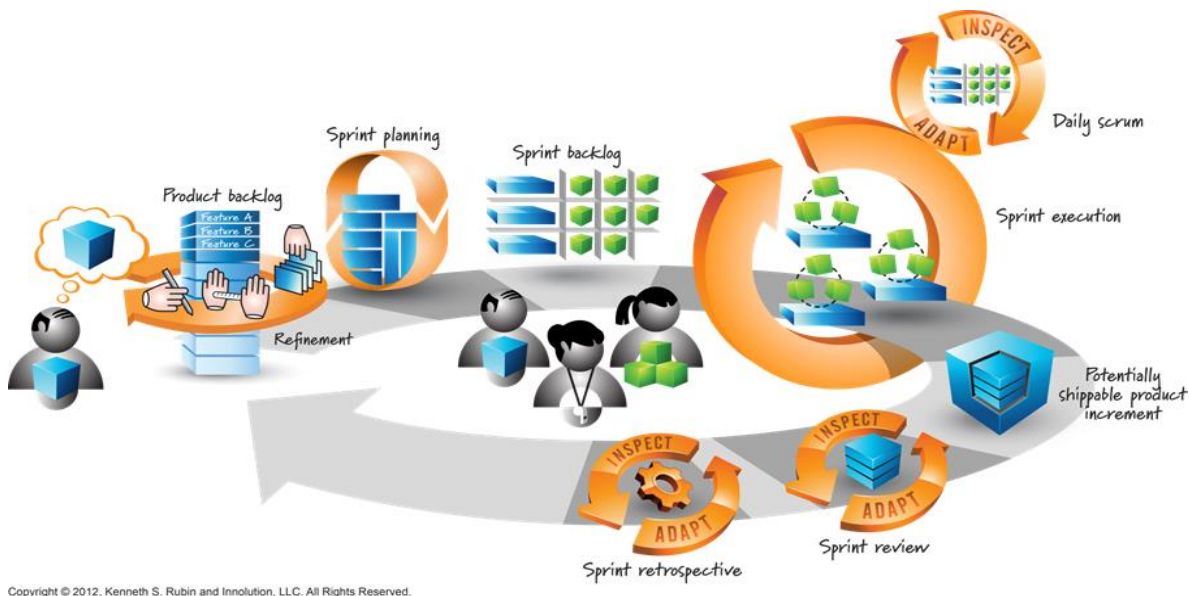


Figura 2.1. Marco de trabajo Scrum

Comunicación. La comunicación puede ser compleja incluso dentro de un solo equipo o entre el equipo y el cliente. En un entorno distribuido, esta complejidad se incrementa significativamente. La comunicación inadecuada puede provocar malentendidos, omisiones, errores y, en consecuencia, retrabajo y sobrecostos.

Cuestiones culturales. Las diferencias lingüísticas, las distintas formas de pensamiento y comunicación, así como los diversos hábitos de trabajo, pueden provocar tensiones entre los equipos, disminuir la confianza y generar retrasos en el proyecto.

Coordinación del trabajo. Asignar tareas entre múltiples ubicaciones y diferentes zonas horarias resulta más complejo que en un proyecto ejecutado en un solo sitio. Esto incrementa los costos de coordinación y puede provocar que los resultados se obtengan con retraso.

Visibilidad y control. En proyectos centralizados, el conocimiento sobre la estructura del proyecto, roles, responsabilidades, actividades y estado de avance se transfiere con relativa

facilidad a todos los participantes. No obstante, en proyectos con equipos dispersos geográficamente, garantizar la entrega oportuna de esta información representa un desafío, especialmente cuando intervienen múltiples empresas o zonas horarias.

Métricas del proyecto. La recopilación de métricas en proyectos globales puede verse limitada por diferencias en las infraestructuras tecnológicas o por procesos heterogéneos, lo que conduce a datos inconsistentes y dificulta la evaluación cuantitativa del desempeño del proyecto.

Cuestiones políticas. Pueden surgir tensiones internas cuando los empleados perciben riesgos de pérdida de empleo o competencia con equipos remotos. Asimismo, pueden existir diferencias regulatorias o intereses divergentes entre países o regiones que generen objetivos contradictorios o agendas ocultas.

Protección de la propiedad intelectual. La protección del conocimiento y los activos tecnológicos puede representar una preocupación significativa, especialmente en regiones donde las leyes de propiedad intelectual son débiles o inconsistentes.

Infraestructura y herramientas de desarrollo. En equipos distribuidos pertenecientes a diversas organizaciones, es común que existan diferencias en herramientas e infraestructuras. Organizaciones más maduras pueden contar con plataformas propietarias avanzadas, mientras que equipos más pequeños suelen utilizar herramientas de código abierto. Estas disparidades pueden generar problemas de compatibilidad en los entornos de desarrollo e integración.

2.2 Administración de Proyectos

La gestión de proyectos implica aplicar habilidades, conocimientos, técnicas y herramientas a las actividades del proyecto con el propósito de satisfacer sus requerimientos. La administración de proyectos implica dirigir y coordinar dichas actividades para alcanzar los resultados esperados. Los equipos pueden lograr estos resultados mediante diversos enfoques de gestión, tales como métodos híbridos, adaptativos o predictivos [13].

Un proyecto se concibe como un esfuerzo temporal destinado a producir un resultado, producto o servicio único. Su naturaleza temporal implica que posee un inicio y un final claramente establecidos, ya sea para el proyecto completo o para alguna de sus fases. Los proyectos tienen la posibilidad de ejecutarse de forma individual o integrarse en un sistema

más amplio o un portafolio.

La persona a la que la organización ejecutora ha asignado la responsabilidad de dirigir al equipo del proyecto y asegurar que se alcancen las metas establecidas es el director del proyecto. Entre sus obligaciones están facilitar el trabajo del equipo, organizar actividades y manejar los procedimientos necesarios para lograr los resultados esperados. El equipo del proyecto está integrado por los individuos encargados de realizar las actividades establecidas en el plan de trabajo, con el fin de lograr los objetivos del proyecto.

2.3 Comunicación efectiva

La comunicación efectiva es un proceso que implica compartir mensajes de forma clara y comprensible, tanto al transmitir una idea como al verificar que ésta haya sido entendida correctamente. Para lograrlo, es indispensable contar con un alto nivel de empatía y habilidades de escucha [14].

De acuerdo con [15], la comunicación efectiva se define como un proceso en el que los mensajes son enviados y comprendidos de forma precisa y clara, lo que genera un efecto positivo en las relaciones interpersonales. Se centra en la relevancia de edificar confianza, escuchar con atención, comunicar ideas apropiadamente y administrar conflictos de manera constructiva.

Comunicación efectiva en el desarrollo de proyectos SCRUM en un ambiente distribuido.

A fin de asegurar el éxito del proyecto, resulta esencial comprobar que la comunicación entre los miembros del equipo de desarrollo sea efectiva dentro de un entorno Scrum distribuido. Para verificar que se ha realizado una comunicación efectiva, es imprescindible examinar cada uno de los eventos en Scrum y garantizar que ciertos indicadores se estén cumpliendo. Algunos aspectos esenciales para cada evento en un contexto distribuido se exponen a continuación:

Reunión de Planificación del Sprint (*Sprint Planning*). En esta reunión, los objetivos del Sprint deben estar claramente definidos y ser entendidos por todos los integrantes del equipo; las historias de usuario se dividen en tareas concretas y se asignan a los responsables, y se acuerda de manera explícita el alcance y los entregables del Sprint, especificando con exactitud qué historias serán atendidas durante la fase.

Reuniones diarias (*Daily Scrum*). En las juntas diarias, todos los integrantes del equipo se involucran activamente, intercambian novedades acerca de su progreso personal, de los retos afrontados y de las tareas que tienen por delante. Además, se reconocen y abordan los inconvenientes o estancamientos que podrían obstaculizar el avance del *Sprint*.

Revisión del *Sprint* (*Sprint Review*). En esta reunión se presentan los resultados obtenidos y las funcionalidades completadas, se recoge la retroalimentación de los Stakeholders y se analizan las acciones futuras y las modificaciones requeridas con base en los comentarios recibidos.

Retrospectiva del *Sprint* (*Sprint Retrospective*). En la Retrospectiva del Sprint se analizan los desafíos y aspectos positivos del ciclo, junto con las oportunidades para optimizar la colaboración, la comunicación y los procesos del equipo. Además, se determinan acciones concretas para atender las áreas de mejora identificadas.

Retroalimentación del cliente. El nivel de satisfacción del cliente con el producto entregado constituye un indicador fundamental del desempeño del equipo Scrum, ya que evidencia que sus expectativas y necesidades fueron entendidas y comunicadas adecuadamente. En cambio, si existen dificultades con el producto, puede ser señal de errores comunicacionales en alguna fase del proceso.

Además de estos aspectos particulares para cada evento de Scrum, existen otras señales generales que indican una comunicación efectiva en un contexto distribuido:

- Empleo efectivo de herramientas en línea para la colaboración, como Slack [16] y Microsoft Teams [17], así como Jira [18] y Trello [19] para la administración y seguimiento del trabajo del proyecto. También se incluyen correo electrónico y la aplicación recomendada Comunica-Scrum.
- Todos los miembros del equipo deben involucrarse de manera activa en las discusiones y reuniones.
- En todas las interacciones, una comunicación precisa y clara, sin malentendidos ni ambigüedades.
- Respuestas a tiempo a las preguntas y consultas de los integrantes del equipo.
- La posibilidad de que los miembros del equipo estén disponibles y sean accesibles para colaborar y comunicarse.

Para asegurar una comunicación efectiva a lo largo de todo el desarrollo del proyecto Scrum

en un entorno distribuido, es fundamental supervisar y evaluar de manera frecuente estos indicadores en cada evento y adaptar las prácticas comunicativas según se requiera.

2.4 Modelos de comunicación

La comunicación es un proceso intencional mediante el cual dos o más personas intercambian información con el propósito de transmitir o recibir datos u opiniones. Este proceso abarca varias etapas fundamentales: la creación de la intención de comunicar, la elaboración del mensaje, su codificación (cifrar), la transmisión de la señal, la recepción por parte del destinatario, la decodificación (descifrar) y, finalmente, la interpretación del mensaje.

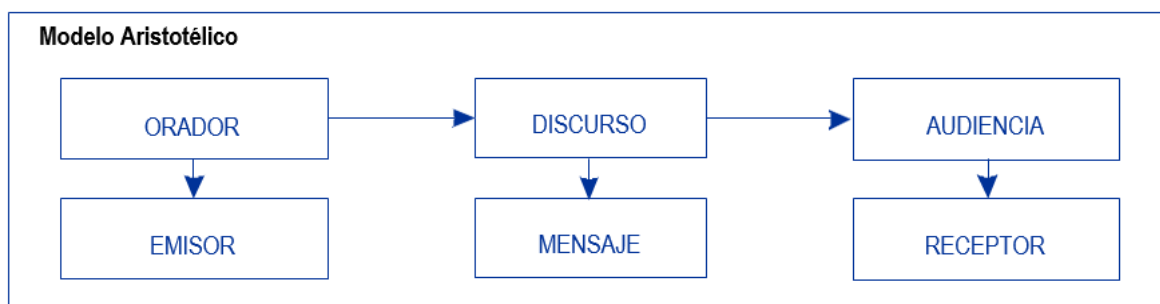
Un modelo básico de comunicación humana contempla a un emisor que envía un mensaje y a un receptor que lo recibe e interpreta.

A continuación, se presentan algunos modelos de comunicación:

Modelo Aristotélico

Fue el primero en representar el proceso comunicativo dentro de su retórica, señalando que la comunicación consiste en emplear todos los recursos de persuasión disponibles para convencer a la audiencia [20].

Componentes del proceso comunicativo propuestos por Aristóteles: Orador (¿quién?) → Discurso (¿qué?) → Oyente (¿a quién?) (ver Figura 2.2).



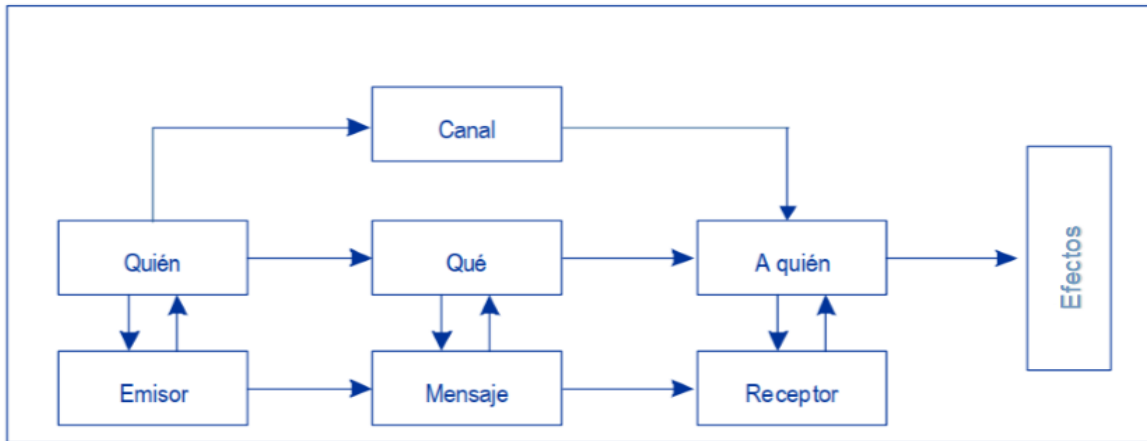
*Figura 2.2. Modelo Aristotélico
Fuente: tomada de [20]*

Modelo de Lasswell

El modelo propuesto por Harold Lasswell (véase Figura 2.3) es esencialmente descriptivo y tiene como propósito definir los ámbitos de análisis de los actos comunicativos, los cuales

pueden caracterizarse respondiendo cinco preguntas fundamentales [20].

Una manera adecuada de describir un acto de comunicación consiste en plantear las siguientes interrogantes: ¿Quién dice?, ¿Qué dice?, ¿Por qué canal?, ¿Para Quién?, ¿Con qué efecto?



*Figura 2.3. Modelo de Lasswell
Fuente: tomada de [20]*

Modelo de Shannon y Weaver

Este es un modelo de comunicación, o más precisamente, una teoría de la información, concebido desde la perspectiva de la cibernética, disciplina enfocada en el estudio del funcionamiento de las máquinas, especialmente las electrónicas. Para Claude Elwood Shannon, el concepto de información posee un significado estrictamente vinculado a la electrónica digital, distinto del asociado a las noticias difundidas por la prensa, la radio o la televisión [21].

El modelo de Shannon puede ser usado en cualquier mensaje, sin importar lo que signifique. Su teoría posibilita el análisis del volumen de información que un mensaje contiene, dependiendo de la capacidad del medio por el cual es transportado. Esta habilidad se manifiesta en términos binarios (0 y 1) a través de los bits, que están vinculados con la velocidad de transmisión. Esta última puede verse disminuida debido a la existencia de ruido.

De acuerdo con Shannon, la duración de la transmisión de un mensaje es proporcional al volumen de información que incluye; en otras palabras, si la información transmitida es más extensa, el tiempo requerido será mayor.

Su objetivo fue crear una ecuación matemática para evaluar el valor informativo de los mensajes, es decir, la precisión con que el mensaje del emisor llega al receptor.

El modelo de Shannon considera la información como un valor que se puede medir en los procesos de comunicación.

Los objetivos fundamentales de este modelo se enfocan en incrementar la eficiencia en la transmisión de información mediante la mejora de la velocidad de creación y envío de mensajes, el fortalecimiento de la capacidad de los canales utilizados y la implementación de codificaciones claras y precisas que eliminen la ambigüedad y reduzcan al mínimo los ruidos que puedan interferir entre el emisor y el receptor.

La comunicación es entendida por el modelo de Shannon, que se ilustra en la Figura 2.4, como un proceso que consta de cinco componentes fundamentales: el emisor, el canal, el receptor, la fuente y el destinatario. Asimismo, incluye la noción de ruido, que se comprende como toda alteración que impacta la señal durante su transmisión [21].

Los componentes del modelo se explican de la siguiente forma:

- a) **Origen:** Es la fuente del proceso de comunicación y crea las palabras o signos que constituyen el mensaje a enviar.
- b) **Transmisor:** Es responsabilidad del emisor técnico transformar el mensaje en un conjunto de códigos o señales adecuados para ser transmitidos por medio del canal.
- c) **Canal:** Es el canal técnico que lleva las señales codificadas por el emisor. Con respecto al teléfono, pueden ser utilizados cables o redes de microondas para las comunicaciones internacionales.
- d) **Receptor:** Es el receptor técnico, cuya tarea es contraria a la del transmisor; convierte las señales que recibe en un mensaje que el destinatario final puede entender.
- e) **Destinatario:** El receptor verdadero es la persona o la entidad a quien se destina el mensaje.
- f) **Ruido:** Se refiere a cualquier interrupción que modifique la señal durante su transmisión, como interferencias en televisión o radio, ruidos no deseados, distorsiones visuales o cualquier componente que obstaculice la recepción del mensaje.

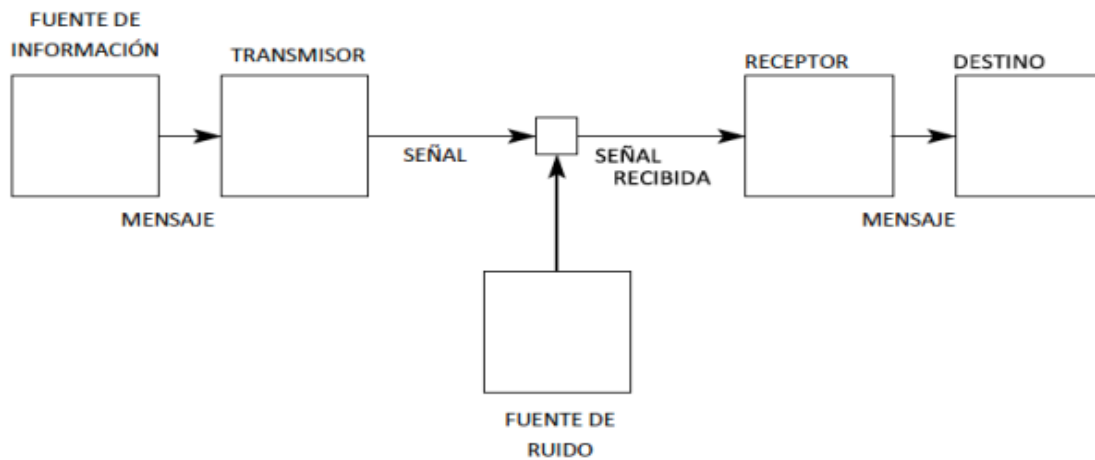


Figura 2.4. Modelo de Shannon y Weaver
Fuente: tomada de [21]

Modelo de Berlo

De acuerdo con David K. Berlo, la comunicación es un proceso dinámico en el que alguien (o un grupo) actúa como fuente: genera una idea o intención, la convierte en un mensaje (la codifica) y la envía por algún medio o canal para que otra persona o grupo, el receptor, la interprete (la decodifique) y la entienda [22].

Berlo tomó el modelo original de Shannon y Weaver, que era básicamente técnico y matemático (pensado para la transmisión de señales), y lo transformó para que sirviera a la comunicación entre seres humanos, incorporando factores más personales y sociales. Los componentes principales del modelo de Berlo (ver Figura 2.5) son los siguientes:

- **Fuente:** Tiene un propósito claro al querer transmitir algo. Es quien inicia la comunicación, ya sea una persona o grupo de personas. La manera en que lo realiza se basa en sus habilidades de comunicación (saber hablar, escuchar, escribir, leer o pensar con claridad), su actitud (lo que siente sobre el tema, sobre sí mismo o sobre la persona a la que le está transmitiendo el mensaje), su grado de conocimiento acerca de lo que dice y el contexto sociocultural donde ha sido educado.
- **Mensaje:** Es el contenido en sí mismo, es decir, las ideas que la fuente traduce a un código (palabras, imágenes, gestos, etc.) que tanto emisor como receptor conocen y comparten.

- **Canal:** El canal físico por el que se transmite el mensaje. Berlo lo vincula directamente con los cinco sentidos: vista (ver imágenes y leer), oído (escuchar música o la voz), tacto (contacto y texturas), olfato y gusto (aunque se emplean con menor frecuencia en la comunicación diaria).
- **Receptor:** La persona o personas que reciben el mensaje y lo descifran. Tienen exactamente los mismos factores que la fuente (habilidades, actitudes, conocimientos, cultura) y esos elementos influyen en cómo interpretan lo que se les dice.

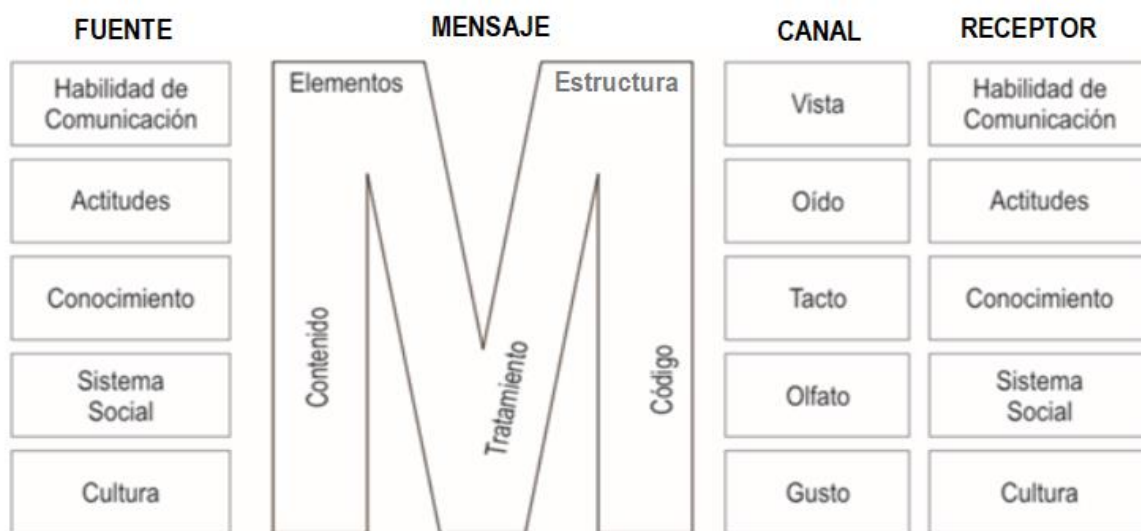


Figura 2.5. Modelo de Berlo
Fuente: tomada de [22]

Modelo de Jakobson

Desde la perspectiva de la Lingüística Funcional, la lengua no es un conjunto abstracto de reglas, sino un sistema que surge directamente de la actividad humana cuyo propósito esencial es permitir que una persona exprese sus intenciones y las haga llegar a los demás.

Uno de los aportes más valiosos en este sentido proviene de Roman Jakobson, quien, al analizar las funciones del lenguaje, mostró hasta qué punto éste es mucho más que un simple vehículo de datos: sirve para convencer, crear vínculos afectivos y sociales, construir versiones compartidas de la realidad, entre muchas otras cosas. Su trabajo destaca las sorprendentes coincidencias entre los enfoques más modernos del análisis lingüístico y los

conceptos centrales de la Teoría Matemática de la Comunicación, especialmente en torno a ideas como redundancia, cantidad de información, código y ruido semántico [23].

Aún con esas conexiones evidentes, la lingüística no se limita a estudiar la comunicación como un proceso técnico; también se entiende como la base misma de la cultura y como la principal herramienta que tenemos para relacionarnos con los otros.

Jakobson, precisamente, parte de la necesidad de contar con un esquema claro de la comunicación antes de definir las funciones del lenguaje. Por eso presenta su conocido modelo bajo el título “Sobre los Factores de Comunicación y las Funciones del Lenguaje” (véase Figura 2.6): un emisor envía un mensaje a un destinatario. Ese mensaje siempre remite a un contexto o referente que el receptor debe ser capaz de reconocer y comprender.

Para que la comunicación funcione, ambos participantes han de compartir un código común que permite al emisor codificar sus ideas y al receptor descifrarlas correctamente. Finalmente, está el contacto, entendido por Jakobson no solo como el canal físico (voz, papel, pantalla, etc.), sino también como el vínculo psicológico que mantiene abierta la interacción entre las dos partes.

Lo que hace especialmente relevante el modelo de Jakobson, pese a su evidente influencia de la teoría matemática de la comunicación, es que deja de reducir el mensaje a una simple transferencia de información y lo coloca en el centro de una experiencia humana mucho más compleja y rica [23].

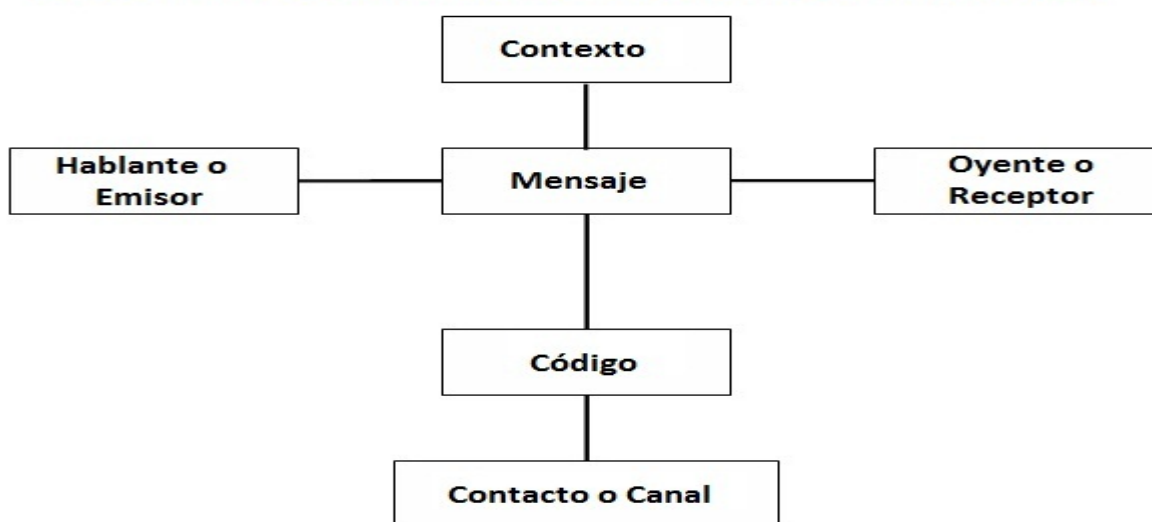


Figura 2.6. Modelo de Jakobson
Fuente: tomada de [23]

Modelo de Schramm

Wilbur Schramm, en su trabajo de 1954, propone varios modelos que buscan explicar cómo influyen los medios de comunicación en las audiencias, pero a diferencia del enfoque de Lasswell, que veía el flujo básicamente unidireccional, Schramm reconoce que la sociedad misma puede actuar como emisor, incidiendo directamente en quienes producen y gestionan la información.

En uno de sus modelos más representativos, centrado en la comunicación interpersonal humana, distingue claramente el rol de cada componente: la fuente actúa como comunicador que codifica (cifra) el mensaje; el mensaje propiamente dicho se convierte en la señal que viaja, y el destino es el receptor que percibe y decodifica (descifra) esa señal.

Un aspecto innovador de Schramm es la incorporación de canales múltiples, porque los seres humanos no nos comunicamos por una sola vía. Incluye, por ejemplo, el canal auditivo (la voz, el volumen, la entonación), pero también los gestos de las manos, las expresiones faciales y todo el lenguaje corporal en general [24].

La idea central que desprende de estos modelos es que la comunicación humana es un fenómeno profundamente complejo. Cada persona arrastra un bagaje de vivencias ligado a su trayectoria personal, su nivel cultural y su entorno social; por eso, ningún mensaje es neutro: siempre lleva impregnada esa carga experiencial. Para que la comunicación realmente funcione, el mensaje tiene que caer dentro del terreno de experiencia compartida entre emisor y receptor. Schramm habla de “sintonización”: cuanto mayor sea el solapamiento de experiencias entre ambos, más efectiva será la comprensión.

Además, le da al mensaje una doble dirección: no solo se envía, sino que genera una respuesta. A ese flujo de vuelta lo llama retorno o comunicación de retorno (*feedback*), un concepto que queda claramente representado en la Figura 2.7.

Modelo de Braddock

Richard Braddock presentó en 1958 una ampliación del clásico esquema de Lasswell que aparece recogida en la Figura 2.8. Su propuesta, conocida como modelo de Braddock, pone el acento en la interacción real entre los participantes del proceso comunicativo y agrega dos variables que los enfoques anteriores solían pasar por alto: por un lado, la finalidad concreta que persigue el emisor al emitir el mensaje y, por otro, el conjunto de circunstancias específicas (lugar, momento, condiciones sociales, etc.) en las que tiene

lugar el acto de comunicación. Con estas incorporaciones, el modelo obliga a plantearse preguntas más precisas y completas sobre cualquier situación comunicativa: ¿Quién?, ¿Dice qué?, ¿Por qué canal?, ¿A quién?, ¿En qué circunstancias?, ¿Con qué intención? y ¿Qué efectos produce? [25].

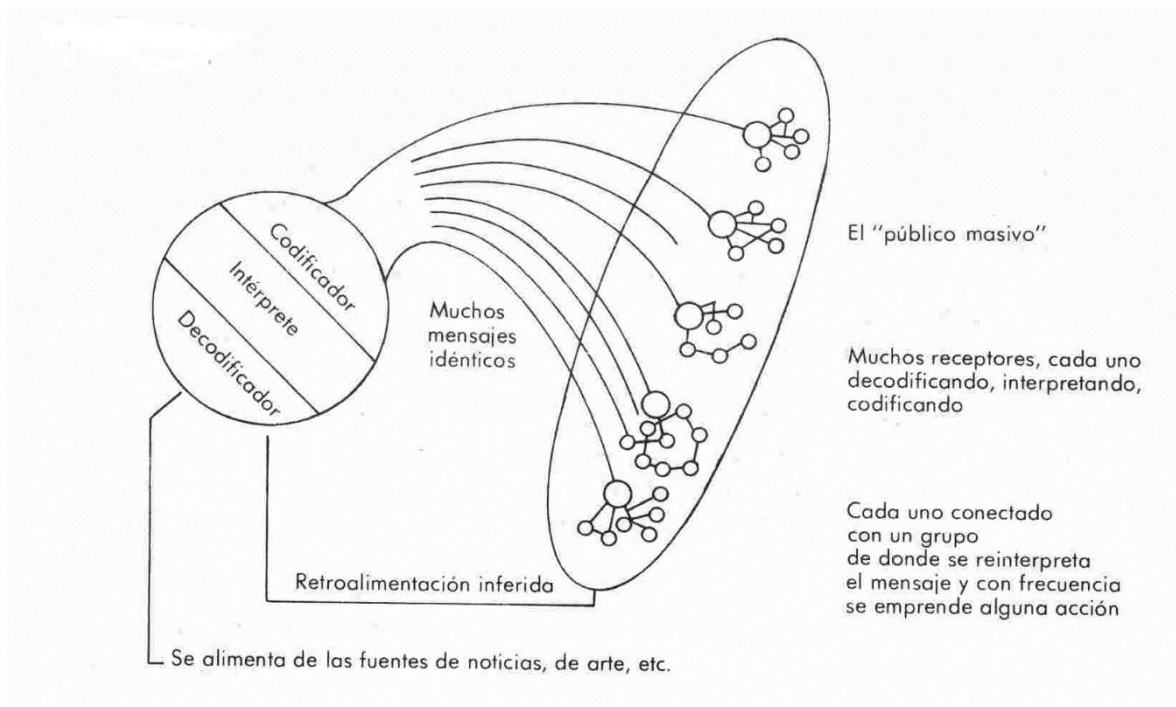


Figura 2.7. Modelo de Schramm
Fuente: tomada de [24]

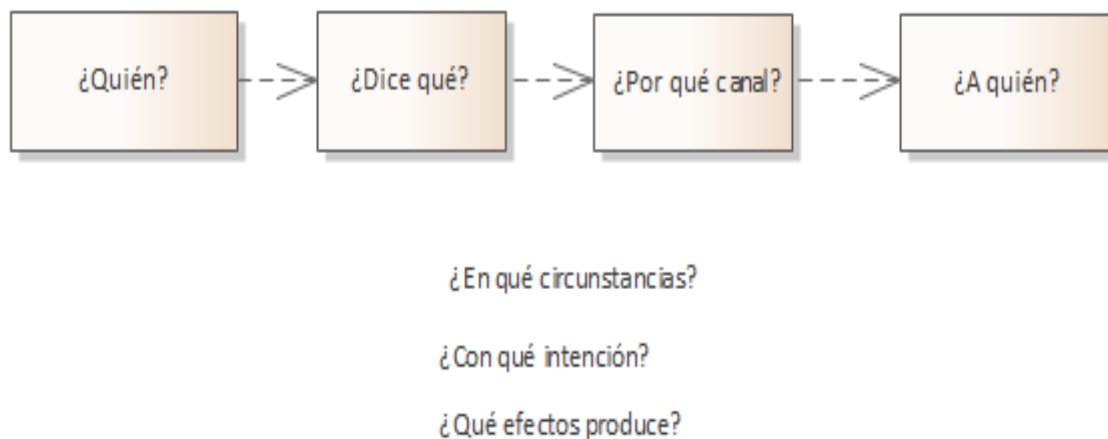


Figura 2.8. Modelo de Braddock
Fuente: tomada de [25]

Modelo de Maletzke

En 1963, Gerhard Maletzke publicó un trabajo pionero sobre la psicología de la comunicación masiva en el que analiza los procesos de la comunicación colectiva y construye un modelo que parte de los componentes clásicos de la comunicación, pero los enriquece con una visión mucho más profunda de los factores psicológicos y sociales que intervienen [26].

Para Maletzke, la comunicación de masas es un fenómeno extraordinariamente complejo que solo se completa gracias a la interacción constante y a las influencias mutuas entre todos sus elementos. Muchos de estos vínculos no son materiales: se trata de procesos y relaciones mentales que se producen dentro de las personas y entre ellas, aunque estén mediadas por objetos físicos.

El autor identifica seis relaciones fundamentales que ayudan a entender su esquema (véase la Figura 2.9):

1. Comunicador-mensaje (intención): Aquí entran en juego dos cuestiones clave: cómo el comunicador elabora el contenido y, al mismo tiempo, cómo el propio mensaje termina ejerciendo cierta presión o condicionamiento sobre quien lo produce.
2. Comunicador-medio: Se refiere a las restricciones y posibilidades que impone el medio elegido a la hora de hacer llegar el mensaje al público.
3. Comunicador-receptor: En la comunicación masiva los roles están fijados de antemano (no hay intercambio inmediato como en la conversación cara a cara). El comunicador suele ser percibido como una figura de autoridad o líder de opinión, y el receptor construye una imagen determinada de él.
4. Mensaje-medio: El contenido adquiere una forma concreta según el medio que lo transmita (prensa, radio, televisión, etc., aunque ni el mensaje ni el medio, por sí solos, generan efectos psicológicos directos.
5. Receptor-mensaje: La manera en que el público recibe e interpreta el contenido está marcada de manera significativa por lo que espera o necesita.
6. Receptor-medio: En gran medida, la disposición del receptor para aceptarlo está determinada por el prestigio, la credibilidad o la imagen que tiene del medio.

En este modelo el mensaje siempre es transmitido al receptor por medio de un canal de

comunicación de masas; cada uno de esos canales tiene características técnicas propias que influyen en cómo se percibe el contenido, la experiencia emotiva que genera y los efectos que provoca en el público. El receptor no es un simple elemento pasivo: decide de manera activa qué medios informativos utilizar y qué mensajes recibir.

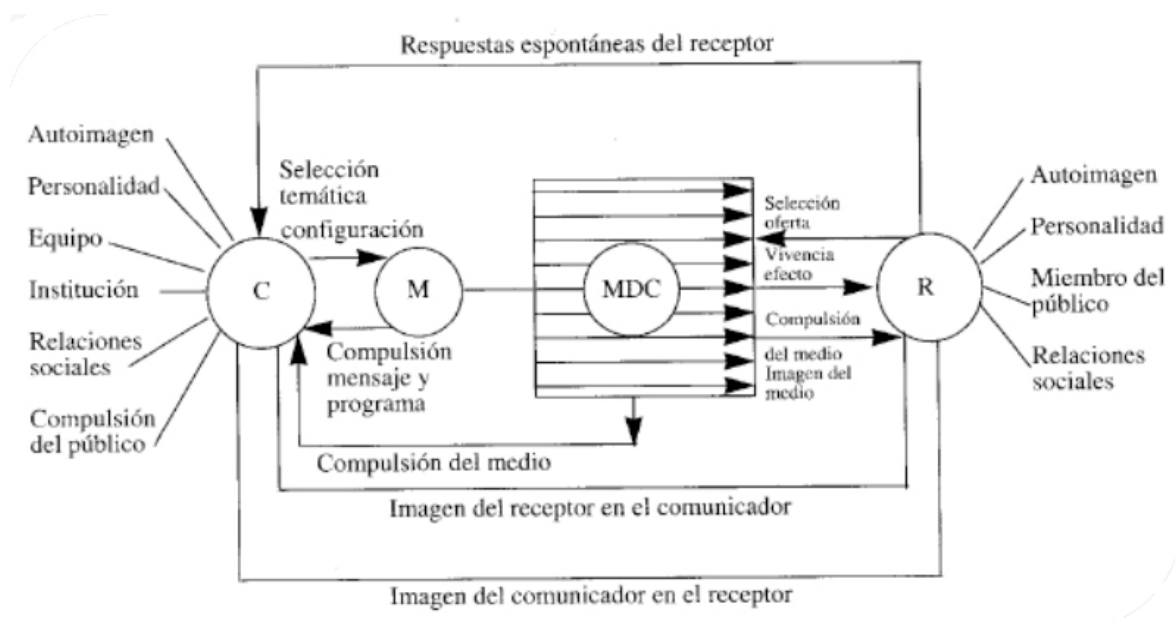


Figura 2.9. Modelo de Maletzke
Fuente: tomada de [26]

Además, está sometido a lo que Maletzke llama “compulsión del medio” y tiene una imagen preconcebida de cada canal de comunicación. (La compulsión hace referencia a las normas, valores y opiniones dominantes en la sociedad que empujan al público a comportarse de cierta manera, incluso cuando va en contra de sus preferencias personales) [26].

Por último, en la Tabla 2.2 se presenta una comparación sistemática de los distintos modelos de comunicación considerados en esta investigación, con el propósito de identificar sus principales coincidencias y diferencias desde una perspectiva conceptual. El análisis comparativo permite observar que, a pesar de haber sido propuestos en contextos históricos y disciplinarios diversos, todos los modelos convergen en la inclusión de tres elementos esenciales del proceso comunicativo: el emisor, el mensaje y el receptor, los cuales constituyen la estructura básica sobre la que se desarrolla cualquier acto de comunicación.

De igual manera, la tabla evidencia que algunos modelos amplían esta estructura elemental al incorporar componentes adicionales, tales como la codificación y decodificación del mensaje, la retroalimentación, el efecto producido en el receptor, la intencionalidad comunicativa y la consideración de receptores técnicos. Estas variaciones reflejan la evolución teórica de los estudios en comunicación, al pasar de enfoques predominantemente lineales hacia modelos más integrales que reconocen la naturaleza dinámica e interactiva del proceso comunicativo. En este sentido, la comparación presentada no solo facilita la comprensión de las aportaciones particulares de cada modelo, sino que también aporta sustento teórico para la selección de los elementos más pertinentes que serán retomados y adaptados en la propuesta desarrollada en el marco de esta investigación.

Tabla 2.2 Características comparativas de los modelos de comunicación

Modelo	Aristotélico	Lasswell	Shannon y Weaver	Berlo	Jakobson	Schramm	Braddock	Maletzke
Emisor	✓	✓	✓	✓	✓	✓	✓	✓
Mensaje	✓	✓	✓	✓	✓	✓	✓	✓
Receptor	✓	✓	✓	✓	✓	✓	✓	✓
Canal	✗	✓	✓	✓	✓	✓	✓	✓
Efecto	✗	✓	✗	✗	✗	✓	✓	✓
Transmisor Técnico	✗	✗	✓	✗	✗	✗	✗	✗
Receptor Técnico	✗	✗	✓	✗	✗	✗	✗	✗
Ruido	✗	✗	✓	✗	✗	✗	✗	✗
Codificación Decodificación	✗	✗	✓	✓	✓	✓	✗	✗
Factores Personales	✗	✗	✗	✓	✗	✗	✗	✗
Mensaje Organizado	✗	✗	✗	✓	✗	✗	✗	✗
Contexto	✗	✗	✗	✗	✓	✗	✗	✗
Retroalimentación	✗	✗	✗	✗	✗	✓	✗	✓
Circunstancia	✗	✗	✗	✗	✗	✗	✓	✗
Intención	✗	✗	✗	✗	✗	✗	✓	✗
Influencias Psicosociales	✗	✗	✗	✗	✗	✗	✗	✓

Justificación del Modelo de seleccionado

En el ámbito de la comunicación, existen diversos modelos teóricos que explican el proceso de transmisión y comprensión de los mensajes. No obstante, para el desarrollo de esta investigación se seleccionó el modelo SMCR propuesto por David Berlo, debido a que ofrece un enfoque analítico detallado de los elementos que intervienen en el proceso

comunicativo. Dicho modelo no se limita únicamente a la transmisión del mensaje, sino que profundiza en las características del emisor y del receptor, así como en la estructura, el contenido y el código del mensaje, aspectos fundamentales para el análisis de la comunicación que se genera durante las ceremonias del marco de trabajo Scrum.

Asimismo, el modelo de Berlo permite examinar la comunicación desde una perspectiva adaptable, lo que resulta pertinente en entornos ágiles y colaborativos donde los mensajes se producen de manera continua y requieren ser comprendidos de forma clara y oportuna por todos los integrantes del equipo. Esta flexibilidad posibilita analizar y mitigar diversos retos identificados en la literatura, tales como las diferencias culturales y horarias que afectan la decodificación de los mensajes, la falta de retroalimentación oportuna y los malentendidos en las ceremonias ágiles. En este sentido, la estructura simple pero conceptualmente robusta del modelo SMCR facilita su adaptación a los roles y eventos de Scrum, integrando el uso de canales digitales y mecanismos de retroalimentación bidireccional, lo que justifica su selección como base teórica principal para el modelo de comunicación propuesto.

Desde esta perspectiva, el modelo de Berlo sustenta la conveniencia de estructurar los mensajes emitidos en las ceremonias Scrum mediante plantillas que permitan organizar el contenido, el código y el tratamiento del mensaje, facilitando así su recepción, decodificación y retroalimentación, especialmente en entornos de desarrollo de software colocalizados y distribuidos.

2.5 Método Delphi

El Método Delphi es una técnica de encuesta que se emplea para alcanzar un consenso entre un grupo de expertos mediante varias rondas de preguntas. Este método fue creado por la corporación RAND durante las décadas de 1950 y 1960 para prepararse frente a las amenazas de seguridad nacional durante la Guerra Fría. Desde entonces, se ha aplicado como método de investigación en diversos campos, como marketing, publicidad, educación y ciencias médicas. Un estudio típico de Delphi incluye múltiples rondas de cuestionarios escritos enviados a expertos seleccionados. Tras cada ronda, el facilitador recopila y revisa los datos, y luego distribuye un informe de resumen a cada experto, quien revisa el informe y puede estar de acuerdo o en desacuerdo con las respuestas de los demás expertos. Este proceso se repite hasta alcanzar un consenso [27].

La razón para realizar un estudio Delphi se basa en la teoría de que las respuestas colectivas del grupo son superiores a las individuales. La teoría sugiere que incluso el experto más calificado y bien informado puede tener ideas que no son "las mejores", pero cuando se consideran colectivamente, las mejores ideas tienden a destacarse. El artículo de Niederberger y Spranger [28] de 2020 señala que, en algunos campos, como las ciencias técnicas y naturales, el método Delphi se utiliza para analizar desarrollos futuros, mientras que, en otros campos, como las ciencias de la salud, se emplea para alcanzar consensos.

En biblioteconomía y ciencia de la información, los estudios Delphi se han convertido en un método de investigación popular; los investigadores han utilizado estudios Delphi para responder preguntas como cuáles son las mejores prácticas para el desarrollo de colecciones o la instrucción en alfabetización informacional. Aunque los estudios Delphi son técnicas válidas, consideran que mantienen un nivel de evidencia más bajo que los metaanálisis o estudios de correlación. Por esta razón, los estudios Delphi son menos adecuados para preguntas sobre la efectividad de intervenciones y más apropiados para cuestiones sobre prácticas o tendencias. Frecuentemente, los resultados de un estudio Delphi se usan para redactar documentos de posición, técnicas o directrices [27].

Los estudios ideales para aplicar este método presentan principalmente las siguientes características [29]:

- La información disponible es escasa o inexistente, permitiendo que este método extraiga el conocimiento de cada participante.
- El problema no puede ser abordado con técnicas analíticas precisas, pero se beneficia de juicios subjetivos colectivos.
- Se requiere la participación de más expertos de los que podrían interactuar eficientemente en un encuentro cara a cara.
- Los costos y el tiempo de los participantes impiden realizar reuniones presenciales periódicas.
- Se opta por preservar la diversidad entre los participantes con el fin de reforzar la solidez y la credibilidad de los hallazgos, priorizando esta modalidad por encima de las sesiones presenciales, ya que así se reduce el riesgo de que voces más dominantes o personalidades intensas terminen imponiéndose sobre las demás.
- En general, esta técnica es adecuada cuando los expertos están dispersos geográficamente y necesitan mantener el anonimato.

2.6 V de Aiken

Para determinar la validez de contenido de un cuestionario o test, se utiliza el índice V de Aiken. Se utiliza sobre todo para establecer si los elementos de un instrumento son pertinentes y apropiados para medir el constructo que se desea evaluar. Esta acción se aplica en la elaboración de tests psicológicos, educativos y otras herramientas de medición [30].

El índice conocido como V de Aiken posibilita medir hasta qué punto un conjunto de expertos concuerda con respecto a la importancia que tienen los ítems de un instrumento. El objetivo es garantizar que el contenido del instrumento sea adecuado y válido para la finalidad de la evaluación.

Cálculo de la V de Aiken:

- Elección de expertos: Se elige un panel de especialistas que examinarán cada elemento del instrumento.
- Evaluación de Ítems: Cada experto evalúa la importancia de cada ítem en una escala ordinal (por ejemplo, del 1 al 4, donde 1 significa "nada relevante" y 4 "muy relevante").
- Fórmula de la V de Aiken:
$$V = \frac{\sum(X-L)}{n(k-L)}$$

Donde:

- ✓ X es la calificación que cada experto otorga.
- ✓ L es la calificación más baja posible en la escala, por ejemplo, 1.
- ✓ k es la máxima calificación que se puede obtener en la escala (por ejemplo, 5).
- ✓ n representa la cantidad de expertos.

Interpretación

- Valores altos (aproximadamente 1): Señalan que existe un alto nivel de consenso entre los especialistas, lo cual sugiere que los ítems son pertinentes y apropiados para evaluar el constructo.

-
- Valores bajos (aproximándose a 0): Señalan que existe poco acuerdo, lo que sugiere que los ítems podrían no ser apropiados y requerirían ser revisados o eliminados.

2.7 Alfa de Cronbach

El coeficiente Alpha de Cronbach es un instrumento estadístico que se usa frecuentemente para establecer el nivel de fiabilidad interna de una serie de ítems o preguntas en un cuestionario [31].

Esta medida permite valorar hasta qué punto los reactivos se comportan de manera coherente entre sí; cuando todos ellos apuntan a un mismo constructo, el resultado suele reflejarse en un valor alfa elevado, lo cual sugiere una alta fiabilidad interna.

Es importante señalar que el Alfa de Cronbach no se ocupa de verificar si el instrumento realmente mide el concepto para el cual fue diseñado. Un conjunto de preguntas puede mostrar una consistencia interna muy alta y, aún así, no corresponder al constructo teórico que se pretende evaluar. En este sentido, la validez, y no la confiabilidad interna, es la que determina la relación entre el instrumento y el concepto que se busca medir [31].

El cálculo del Alfa se basa en analizar la correlación entre todos los ítems del cuestionario. El coeficiente tiende a crecer cuando estos reaccionan de manera parecida frente al mismo constructo. Por otro lado, la aparición de preguntas que no son coherentes con las demás tiende a resultar en un valor alfa bajo.

El coeficiente Alfa de Cronbach se calcula mediante la siguiente fórmula:

$$\alpha = \frac{N}{N-1} \left(1 - \frac{\sum_{i=1}^N \sigma_i^2}{\sigma_T^2} \right)$$

Donde:

- α es el alfa de Cronbach.
- N es el número de ítems.
- σ_i^2 es la varianza del i-ésimo ítem.
- σ_T^2 es la varianza total de la suma de todos los ítems.

La fórmula coteja la varianza total de las puntuaciones registradas con la que se esperaría

si todos los ítems midieran lo mismo. Un α elevado indica que las preguntas son más consistentes entre sí, lo cual señala un mayor grado de confiabilidad del cuestionario.

Interpretación

- **Valores próximos a 1 (entre 0.8 y 1.0):** Señalan una gran confiabilidad interna. El instrumento es consistente y los ítems están muy correlacionados entre sí.
- **Valores entre 0.7 y 0.8:** Señalan una confiabilidad que se puede aceptar. El instrumento tiene una consistencia razonable.
- **Valores comprendidos entre 0.6 y 0.7:** Señalan un grado de fiabilidad dudoso. Puede que sea preciso revisar algunos elementos del instrumento.
- **Valores menores a 0.6:** Señalan baja confiabilidad. El instrumento puede no ser coherente y podría requerir una revisión o alteración importante.

Capítulo 3 Estado del arte

El presente capítulo tiene como objetivo analizar de manera sistemática la literatura científica más relevante relacionada con la comunicación en proyectos de desarrollo de software, en entornos ágiles, distribuidos y colocados. A partir de estudios empíricos, revisiones sistemáticas y propuestas conceptuales, se examinan los principales problemas de comunicación que afectan tanto a equipos colocados como a equipos distribuidos, evidenciando que la proximidad física no elimina, por sí sola, las barreras comunicativas, y que la dispersión geográfica introduce retos adicionales de naturaleza cultural, temporal y organizacional. Asimismo, se revisan las implicaciones de estos problemas en el marco de trabajo Scrum, donde la comunicación constituye el eje central para la coordinación de roles, la ejecución de ceremonias y la gestión de artefactos.

3.1 Introducción

La comunicación efectiva constituye el pilar fundamental del éxito en proyectos de desarrollo de software, independientemente de si los equipos están colocados o distribuidos geográficamente [32], [33]. Este papel central ha sido reconocido desde los primeros estudios de ingeniería de software, donde se señala que la calidad de la comunicación entre los miembros del equipo determina la comprensión de los requisitos, la coordinación de actividades, la resolución de problemas y, en última instancia, la calidad del producto final [7], [9]. En proyectos ágiles, y especialmente en Scrum— la comunicación se convierte en el mecanismo que sostiene la entrega continua, la colaboración multidisciplinaria y la adaptación al cambio.

En entornos de Desarrollo Global de Software (DGS), estos desafíos se intensifican debido a diferencias culturales, zonas horarias, barreras lingüísticas y limitaciones tecnológicas [34], [35], [36], [37]. De igual forma, estudios muestran que la distancia física reduce la comunicación informal, dificulta la transferencia de conocimiento tácito y afecta la construcción de confianza [1], [38], [39], [40], [41].

Sin embargo, investigaciones recientes también evidencian que los equipos colocados (*on-site*) enfrentan problemáticas comunicativas importantes. Entre las más frecuentes se encuentran la sobrecarga de reuniones, la falta de estandarización en los mensajes, la

informalidad excesiva, las interrupciones constantes y la ausencia de mecanismos de retroalimentación estructurada [42], [43], [44], [45]. Asimismo, los equipos presenciales suelen presentar tensiones interpersonales, estilos comunicativos dispares y malentendidos durante la interacción diaria, factores que también afectan la coordinación y el rendimiento del equipo.

En consecuencia, tanto en equipos distribuidos como on-site, la literatura coincide en que la comunicación continúa siendo uno de los factores más problemáticos y determinantes en el éxito de los proyectos de software. Esta condición evidencia la necesidad de modelos estructurados que orienten el proceso comunicativo y reduzcan la ambigüedad, especialmente en marcos ágiles como Scrum.

3.2 Problemas de comunicación en equipos colocados (on-site)

Conforme a las pruebas empíricas, los problemas de comunicación persisten, incluso cuando los equipos están situados en el mismo espacio físico, a pesar de que la colocación se considera normalmente una condición óptima para el trabajo ágil. Las investigaciones revisadas indican que la colocación no elimina las distancias cognitivas, organizacionales o psicológicas, y que éstas influyen significativamente en la coordinación y el flujo de información.

En el caso del estudio [46] los equipos de una organización colocados, pero distribuidos en distintos edificios y pisos experimentaron demoras en la comunicación e interrupciones en la coordinación, ya que era necesario verificar por medios digitales (correo o Skype) si la otra persona estaba disponible antes de intentar interacción presencial, lo cual generaba retrasos y pérdida de inmediatez en el intercambio de información. Los propios participantes señalaban que “no puedes simplemente ir con esa persona” debido a la dispersión física interna.

Otro hallazgo relevante es la escasa conciencia de las dependencias entre equipos colocados. Aun en proximidad física, los equipos carecían de visibilidad sobre lo que otros estaban desarrollando, lo que provocaba errores, bloqueos y falta de alineación en decisiones técnicas y de requisitos. En [46] identifican que esta falta de visibilidad deriva tanto de distancias cognitivas como organizacionales y suele causar retrasos en la resolución de problemas.

Las distancias cognitivas y psicológicas también son determinantes: equipos que se perciben poco accesibles o que poseen dominios de conocimiento muy especializados tienden a comunicarse menos, generando malentendidos y suposiciones no validadas. En los grupos focales del estudio de [46], algunos equipos expresaron percepciones negativas hacia otros, lo cual incrementaba la resistencia a colaborar y aumentaba las brechas de comunicación.

Desde la perspectiva de la Teoría de Distancias [47], incluso en ambientes colocalizados existen distancias: geográficas (entre pisos/edificios), organizacionales (diferencias en objetivos y jerarquías), cognitivas (diferentes niveles de conocimiento) y psicológicas (esfuerzo percibido para contactar a otra persona). Todas estas requieren esfuerzo adicional para ser superadas y aumentan el riesgo de fallas comunicativas.

Los autores en [44] muestran que, aun con equipos trabajando presencialmente, la falta de estructuras claras en reuniones - como *Scrum of Scrums* o planificación - genera sobrecarga de reuniones, pérdida de foco y repetición de información, afectando la coordinación general del proyecto. Esto evidencia que la proximidad física no sustituye la necesidad de prácticas formales de alineación.

Los trabajos en [42], [43] señalan que en equipos colocalizados, la comunicación informal tiende a dominar, pero esta informalidad también genera ambigüedad, pérdida de trazabilidad y mensajes inconclusos, lo que deriva en decisiones no documentadas o mala interpretación de prioridades del equipo. Incluso cuando herramientas digitales no son obligatorias (por estar físicamente juntos), su ausencia provoca falta de registro y deterioro en la coordinación.

Más recientemente, los propios firmantes originales del Manifiesto Ágil, entrevistados en [48], reconocen que el principio de “la forma más eficiente y efectiva de transmitir información es cara a cara” solo resulta plenamente funcional en equipos muy reducidos (menores a 10–12 personas) y con una cultura altamente colaborativa. En este contexto, la denominada **comunicación osmótica**, entendida como la adquisición informal y no intencional de información derivada de la convivencia en un mismo espacio físico, al escuchar conversaciones, observar interacciones o percibir el trabajo de otros, presenta limitaciones claras para escalar en organizaciones grandes y jerárquicas, donde la simple colocalización no garantiza por sí misma una comunicación efectiva [49].

Finalmente, en [45] muestran que, al comparar equipos presenciales y remotos, incluso quienes defienden la presencialidad reconocen que los entornos colocalizados sufren interrupciones frecuentes, exceso de comunicación no estructurada, ruido organizacional y pérdida de tiempo en reuniones improductivas. Además, señalan que los equipos colocalizados suelen depender demasiado de la comunicación espontánea, lo cual provoca inconsistencia en la difusión de información y desigual acceso al contexto del proyecto.

3.3 Comunicación en el Desarrollo de Software, en Equipos Scrum Distribuidos

La comunicación en el desarrollo de software ha sido identificada como uno de los elementos más críticos para el éxito de los proyectos, ya que influye directamente en la **comprensión de requisitos**, la coordinación técnica y la cohesión social de los equipos. Pressman y Maxim destacan que la comunicación interpersonal y el intercambio informal de información constituyen mecanismos esenciales para compartir avances, tomar decisiones de diseño, resolver impedimentos y transmitir conocimiento tácito [7] [9]. Diversos estudios confirman que las fallas comunicativas continúan siendo un factor determinante de retrasos, conflictos, errores y fallas en proyectos de software. En una revisión exhaustiva de la literatura [32], concluyen que la comunicación es una variable transversal a la efectividad del proyecto, el trabajo en equipo y el desarrollo global, sin importar la metodología utilizada.

En el contexto del Desarrollo de Software Global (DGS), estos retos se ven intensificados por factores característicos de la dispersión geográfica. La literatura reporta que la distancia introduce barreras adicionales como diferencias culturales y lingüísticas [36], [50], [51], zonas horarias con solapamiento reducido [34], [35], falta de confianza y cohesión [52], [53] y herramientas inadecuadas para sostener el trabajo colaborativo [41], [54]. También se ha señalado que la insuficiente documentación y la disponibilidad desigual de información constituyen barreras recurrentes que dificultan la transmisión clara de conocimiento [50]. Revisiones amplias confirman que la comunicación es el principal factor de riesgo en DGS [32], [55]. En este sentido, la distancia afecta la comunicación entre diseñadores, clientes, desarrolladores y administradores remotos, complicando la coordinación y aumentando la posibilidad de discrepancias [56].

Los retos en DGS se vuelven todavía más complejos cuando se combinan con prácticas ágiles, donde la comunicación es el eje de la transferencia de conocimiento, la negociación de requisitos y la sincronización entre roles. En [50] se destacan que la comunicación es esencial para ejecutar procesos ágiles de manera eficiente. Un modelo conceptual ampliamente citado identifica trece cuestiones de comunicación en el AGSD, entre ellas distancia geográfica, diferencias horarias, y lenguaje diferente (ver Figura 3.1) [1]. La combinación de prácticas ágiles con equipos distribuidos puede generar beneficios como entrega rápida, reducción de costos y disponibilidad de talento global [57], [58], [59], pero también demanda una gestión comunicativa altamente disciplinada debido a sus ciclos iterativos, retroalimentación continua y dependencia de interacciones frecuentes [60].

Scrum, como marco ágil, fue originalmente diseñado para equipos pequeños y colocados [61]. Sin embargo, su adopción en entornos distribuidos ha revelado limitaciones importantes: la falta de comunicación clara reduce la calidad y la eficiencia [62], [55], mientras que las ceremonias del marco requieren gestión estricta para evitar malentendidos, especialmente en contextos de distancia cultural y temporal [33], [63], [64]. Las reuniones diarias de 15 minutos juegan un papel clave para difundir avances y bloqueos, pero deben gestionarse adecuadamente para evitar confusión o pérdida de información [65]. Artefactos como el Product Backlog, el Sprint Backlog y los tableros de tareas proporcionan visibilidad al progreso del equipo [39]. La planificación del sprint requiere la participación activa del Product Owner para definir características, mientras que la revisión y la retrospectiva promueven la aceptación del incremento y la mejora continua [61], [66].

En proyectos distribuidos, estas ceremonias enfrentan desafíos adicionales. La falta de solapamiento horario puede limitar la participación, la diversidad cultural puede afectar el estilo y la claridad del mensaje, y las herramientas pueden introducir fricciones si no están estandarizadas. En [67] señalan que la comunicación efectiva en Scrum distribuido requiere herramientas apropiadas, confianza interpersonal, reuniones diarias organizadas y una diferencia horaria mínima para facilitar la colaboración. En [68] identificaron 19 beneficios, 14 desafíos y 16 prácticas de apoyo para la adopción de métodos ágiles en DGS, resaltando nuevamente que la comunicación es el eje central de todos ellos. Además, proyectos AGOSD (*Agile Global Outsourced Software Development*) enfrentan dificultades particulares relacionadas con la comunicación cliente-proveedor, en las que la claridad, la disponibilidad y la negociación son esenciales para el éxito [69].

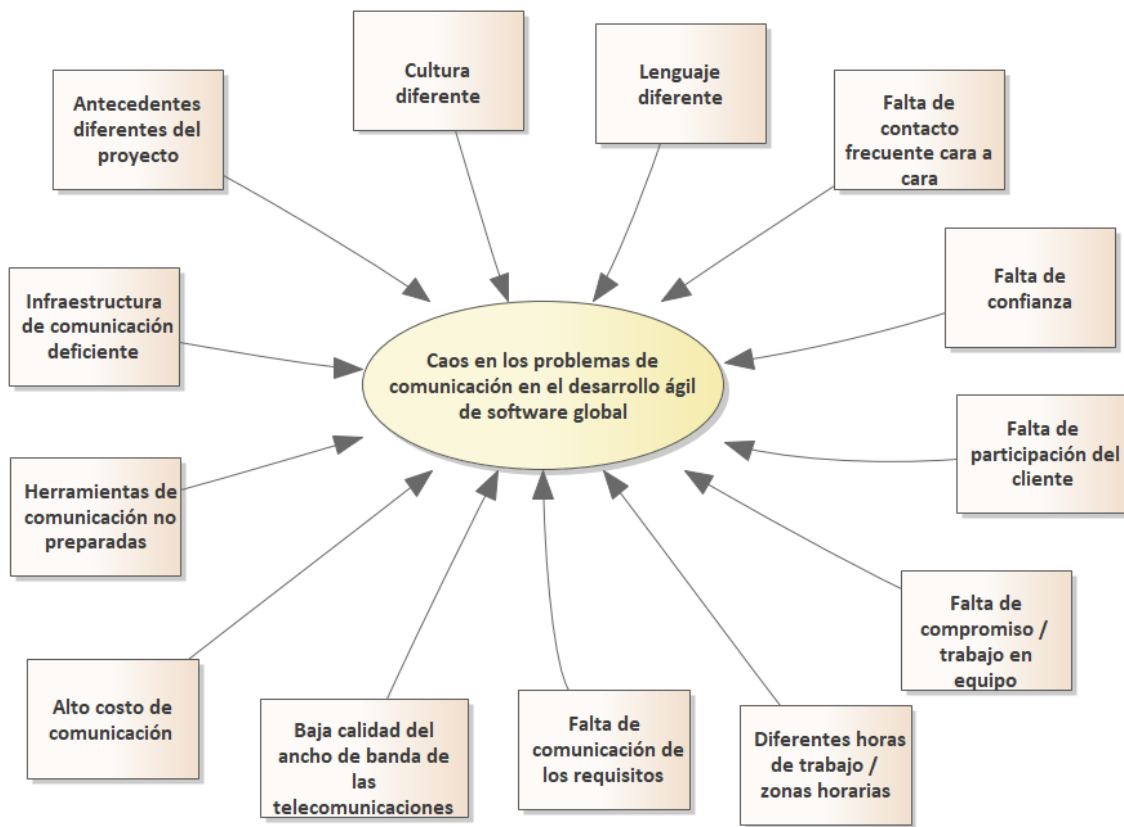


Figura 3.1. Esquema conceptual de las fallas en la comunicación

La gestión de proyectos en DGS también enfrenta carencias en estándares y guías formales. Los autores en [70] proponen la integración del PMBOK con aspectos específicos del DGS para mejorar la coordinación, la visibilidad y el flujo de comunicación entre partes interesadas. Las prácticas ágiles, con sus ciclos iterativos y mecanismos de retroalimentación, pueden compensar algunas de estas deficiencias, pero requiere disciplina comunicativa, plantillas estandarizadas y canales eficientes para ser efectivas a distancia. Finalmente, la evidencia académica ha argumentado reiteradamente qué errores aparentemente pequeños en la comunicación pueden causar pérdidas considerables en proyectos de TI, y que la comunicación constante, más que la documentación excesiva, es esencial en entornos ágiles [39]. Asimismo, la comunicación informal interpersonal es útil para compartir ideas y generar cohesión, aunque debe equilibrarse con la claridad y la formalidad necesarias para evitar ambigüedades [71].

3.4 Desafíos en la Comunicación en el Desarrollo Global de Software y en Entornos Ágiles Distribuidos

La comunicación se ha consolidado como el principal desafío en el Desarrollo de Software Global (DGS), un modelo caracterizado por la distribución geográfica de equipos y actividades, motivado por beneficios como reducción de costos, acceso a talento especializado y continuidad del desarrollo 24/7 [40], [51], [59], [57], [58]. A pesar de estas ventajas, la literatura es consistente en afirmar que la comunicación es el factor de riesgo más crítico en proyectos distribuidos [1], [38], [41], [62]. La dispersión geográfica introduce dificultades como diferencias horarias, barreras culturales e idiomáticas, infraestructura tecnológica insuficiente, falta de comunicación informal, pérdida de conocimiento tácito, retrasos en retroalimentación y bajos niveles de confianza entre equipos remotos. En [72] y [37] demostraron empíricamente que la comunicación es un factor decisivo para el éxito de proyectos DGS, incluso por encima de elementos técnicos o metodológicos.

En este contexto, cuando el DGS se combina con metodologías ágiles, surgen retos aún mayores. En los entornos de Desarrollo Ágil Distribuido (AGSD), los autores en [34] identificaron que los principales desafíos comunicativos incluyen diferencias horarias, barreras idiomáticas, escasa documentación y dificultades para mantener una visión compartida del producto. La falta de comunicación clara afecta directamente la calidad del software, como lo demostraron en [62], mientras que en [35] clasificaron 22 barreras críticas para escalar métodos ágiles, destacando la comunicación como el eje central que condiciona la coordinación y efectividad del equipo. Estos hallazgos confirman que la agilidad, basada en ciclos cortos de retroalimentación y colaboración continua, depende profundamente de la estructura, calidad y ritmo de la comunicación, lo cual se vuelve más complejo a medida que aumenta la distancia geográfica y cultural entre los involucrados.

Scrum, uno de los marcos ágiles más adoptados, requiere comunicación constante y eficiente para sostener sus ceremonias y artefactos. Sin embargo, en equipos distribuidos su implementación enfrenta desafíos significativos. La reunión diaria (Daily Scrum), por ejemplo, puede verse afectada por fallas técnicas, dificultades de participación, diferencias culturales, problemas de claridad idiomática y retrasos en la recepción de información clave [38], [39], [73]. En [55], mediante un metaanálisis, concluyeron que Scrum por sí solo no es suficiente para garantizar una comunicación efectiva en contextos distribuidos: sin

mecanismos explícitos de alineación y canales estandarizados, las ceremonias pueden volverse ineficientes o perder su propósito.

Debido a estos desafíos, el uso de herramientas de comunicación toma un rol fundamental. Estudios como en [41] muestran que el empleo de pizarras inteligentes, repositorios centralizados y sistemas de colaboración visual mejoran la percepción de co-ubicación y facilitan la cohesión del equipo al proporcionar mayor visibilidad del trabajo. Asimismo, el marco EasyComm [54] demostró que un centro de comunicación unificado puede optimizar la captura de requisitos y reducir inconsistencias. No obstante, la literatura aclara que las herramientas, por sí mismas, no son suficientes para resolver los problemas comunicativos: la efectividad depende de cómo se integran en el flujo de trabajo, de la claridad de los mensajes y de la disciplina comunicativa de los equipos [69], [66].

3.5 Modelos y marcos existentes

A lo largo de la última década se han propuesto algunos modelos, marcos y herramientas con el propósito de mejorar la comunicación en entornos ágiles distribuidos. La Tabla 3.1 sintetiza los principales aportes publicados entre 2016 y 2023, comparándolos con la contribución de la presente tesis.

Tabla 3.1 Comparativa de modelos y marcos de comunicación en equipos Scrum distribuidos

Autores	Año	Enfoque principal	Fortalezas	Vacíos detectados
[74]	2013	Modelo preliminar de madurez en comunicación para DSD	Introduce niveles de madurez y diagnóstico	Modelo preliminar; no adaptado a Scrum
[34]	2016	Revisión sistemática (21 estudios)	Sintetiza desafíos y técnicas	No modelo estructurado para Scrum
[41]	2016	Percepciones sobre herramientas e infraestructura de comunicación	Mejora sensación de co-ubicación	Enfoque solo en herramientas
[75]	2016	Extensión del modelo de madurez en comunicación	Métricas e indicadores más sólidos; enfoque estructurado	No considera procesos ágiles ni ceremonias Scrum
[76]	2017	Escala de Comunicación Efectiva	Participación, herramientas cultura,	No alineado con ceremonias Scrum

[32]	2017	Revisión de 184 artículos	Áreas clave de investigación	Enfoque amplio, sin modelo específico
[77]	2017	Evaluación del método de madurez en comunicación	Validación experimental parcial; operacionaliza el diagnóstico	No integra guías prácticas; sin adaptación a Scrum
[54]	2019	EasyComm (herramienta centralizada)	Validado con expertos	Solo cliente-equipo
[62]	2019	Impacto en eficiencia y calidad	Demuestra efectos negativos	No propone modelo
[52]	2019	Scrum en proyectos científicos	Mejora coordinación	No aborda comunicación distribuida
[78]	2019	Revisión sistemática sobre desafíos de comunicación entre <i>Stakeholders</i> distribuidos	Clasificación clara de barreras clave; útil para identificar riesgos	Sin modelo estructurado; enfoque solo en <i>Stakeholders</i>
[35]	2020	22 barreras + fuzzy-AHP	Taxonomía útil	Enfoque general
[79]	2020	CFDSD: marco de comunicación para DSD	Framework completo con roles, artefactos y procesos	No diseñado específicamente para Scrum; requiere adaptación
[80]	2021	Modelo RCM (87 % efectividad)	Ajustado a ágiles	Solo gestión de cambios
[81]	2021	Entrevista a profesionales de Australia, China, India	Rol de documentación, desafíos de comunicación	Sin solución estructurada
[53]	2021	Errores de comunicación → fracaso	Identifica causa clave	Sin modelo específico
[36]	2021	Barreras culturales India-Suecia	Necesidad de formación cultural	Enfoque sociocultural
[82]	2021	Marco para superar desafíos de comunicación en equipos ágiles distribuidos	Identifica barreras reales; propone un marco práctico para mejorar la comunicación remota	No orientado a Scrum; carece de plantillas y lineamientos operativos

[33]	2022	Modelo teórico (transparencia, calidad, disciplina) + 11 acciones	Principios y prácticas claras	Sin plantillas ni integración tecnológica
[55]	2023	Metaanálisis casos Scrum distribuidos	Identifica barreras persistentes	Necesidad de modelos adaptados
Modelo propuesto (esta tesis)	2025	Adaptación SMCR de Berlo + Comunica-Scrum	Plantillas por ceremonia, retroalimentación continua, personalización lingüística/cultural, aplicable onsite y distribuido, un estudio de caso	Requiere más validación empírica

3.6 Revisión de herramientas comerciales y gratuitas empleadas en la gestión de proyectos de software

Los instrumentos evaluados para la administración de proyectos de software presentan diferencias en cuanto a su nivel de apoyo a Scrum. Kunagi [83], Jira [84], IceScrum [85], SprintoMeter [86] y ScrumDo [87] están enfocadas directamente en las metodologías ágiles y posibilitan el manejo de historias de usuarios, estimaciones, métricas de progreso y *Sprints*. La fortaleza de Jira para planificar, supervisar y crear informes del proyecto lo hace sobresalir.

Aunque no están específicamente diseñadas para Scrum, otras herramientas como Trello [88] y Asana [89] facilitan la organización de tareas a través de listas y tableros. Por último, Slack [90] y Discord [91] son plataformas de comunicación y cooperación grupal que son útiles para organizar actividades, aunque no tienen características específicas de gestión de proyectos.

A continuación, se presenta un cuadro comparativo de las herramientas analizadas; consúltese la Tabla 3.2.

El análisis comparativo de las herramientas comerciales y gratuitas empleadas en la gestión de proyectos de software permite concluir que, si bien existen soluciones ampliamente adoptadas como Kunagi, Jira, IceScrum, SprintoMeter y ScrumDo, estas se encuentran principalmente orientadas a la gestión operativa del proyecto, es decir, al seguimiento de tareas, historias de usuario, estimaciones, métricas y planificación de Sprints. En particular,

Jira destaca por su robustez para la planificación, el control del avance y la generación de reportes; no obstante, su enfoque sigue siendo predominantemente administrativo.

Tabla 3.2 Cuadro comparativo de herramientas similares

	Gestión de Proyectos Ágil	Gestión de Proyectos Kanban	Cuadro de Mando	Seguimiento de proyecto	Historias de usuario (priorizar)	Product Backlog	Panel interactivo del sprint	Planning poker	Distribución de tareas al equipo de desarrollo	Gráfica burn down	Datos visuales en tiempo real	Adjuntar archivos	Comunicación Colaborativa	Salas de chat	Manejo de canales
Kunagi	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
ScrumDo	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
SprintoMeter	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
IceScrum	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
Jira	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Trello	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
Asana	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗
Slack	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓
Discord	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓

Por otra parte, herramientas como Trello y Asana ofrecen mecanismos flexibles para la organización visual del trabajo mediante tableros y listas, pero carecen de una alineación explícita con las ceremonias, roles y artefactos definidos por Scrum. Asimismo, plataformas de comunicación como Slack y Discord facilitan la interacción y el intercambio informal de mensajes; sin embargo, no proporcionan estructuras formales que aseguren la trazabilidad, la estandarización del contenido ni la vinculación directa de los mensajes con los eventos de Scrum.

La necesidad de una herramienta adicional para soportar el modelo de comunicación propuesto radica en que ninguna de las herramientas analizadas fue diseñada específicamente para estructurar los mensajes emitidos en cada ceremonia de Scrum conforme a componentes comunicativos bien definidos (emisor, receptor, mensaje, código, canal y retroalimentación).

Capítulo 4 Metodología de la solución

En esta metodología se definen las etapas, acciones y procesos requeridos para ajustar el modelo de comunicación al marco de trabajo Scrum, para su posterior puesta en marcha y valoración. La metodología se divide en tres etapas: Marco Analítico, Desarrollo del Modelo y Evaluación del Modelo, las cuales se describen a continuación y se muestran en la Figura 4.1.

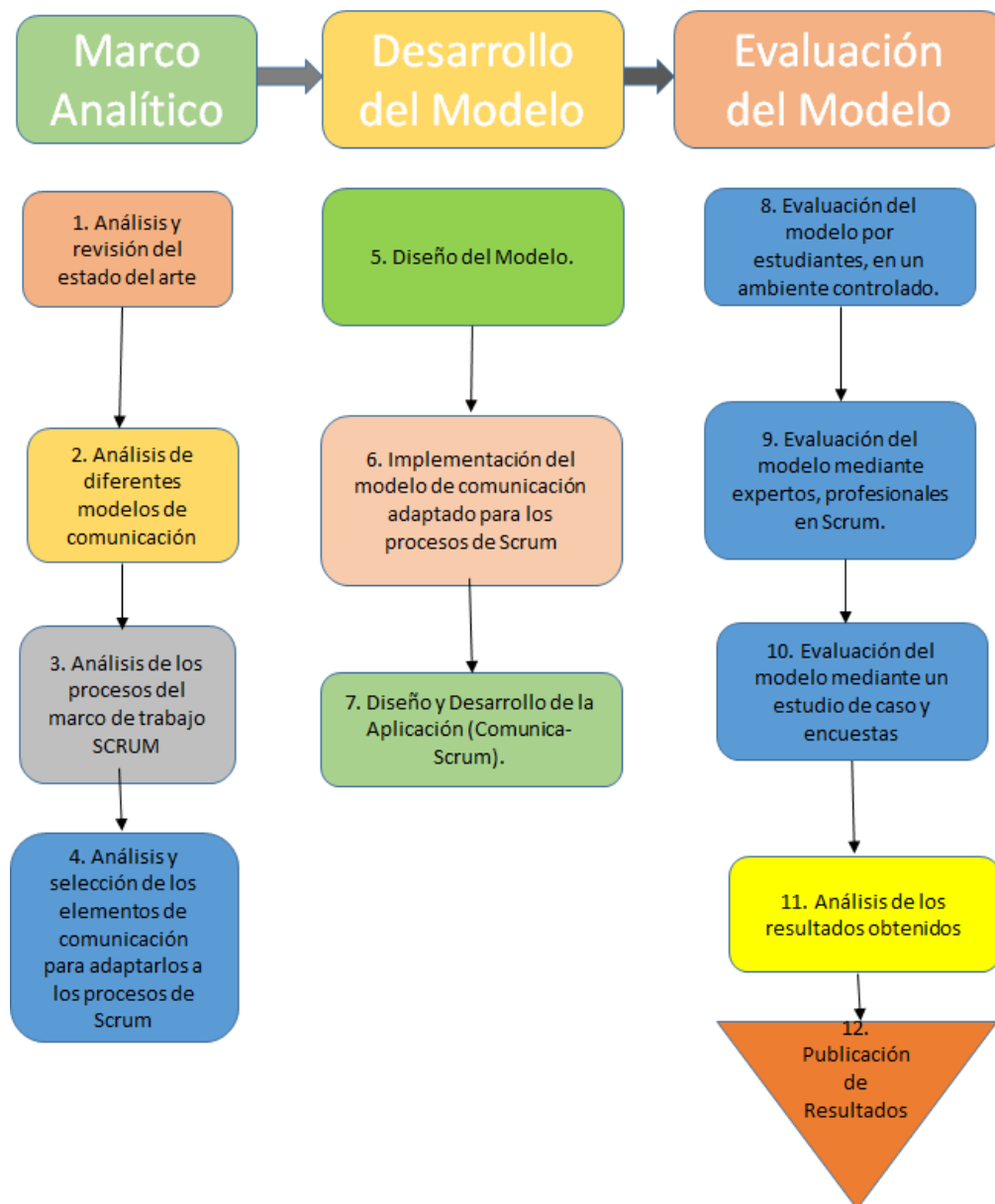


Figura 4.1. Metodología de Desarrollo
Fuente: Elaboración propia

Fase 1: Marco Analítico

Esta fase es el fundamento metodológico y conceptual del proyecto. Su propósito es comprender el contexto, analizar referentes teóricos importantes e identificar los elementos que integrarán el modelo de comunicación adaptado.

1. Análisis y revisión del estado del arte

Se examinan investigaciones recientes, modelos teóricos y estudios vinculados a la comunicación en grupos de desarrollo de software, ya sea en contextos distribuidos o colocalizados. Este análisis posibilita la identificación de oportunidades de mejoramiento, limitaciones y tendencias que guían la propuesta del modelo de comunicación adaptado.

2. Estudio de diversos modelos de comunicación

Se analizan diferentes modelos de comunicación y sus elementos para determinar cuáles son viables en el contexto del desarrollo de software con Scrum. Esto posibilita elegir los atributos que tienen la capacidad de ser incluidos en el modelo sugerido

3. Análisis de los procesos del marco de trabajo Scrum

Se realiza un análisis de los roles, ceremonias, artefactos y dinámicas de comunicación que ha establecido Scrum. Esta evaluación permite detectar las interacciones más importantes, los flujos de información y los ámbitos en los que es posible mejorar o normalizar la comunicación.

4. Análisis y selección de los elementos de comunicación para adaptarlos a Scrum

Se escogen los componentes de comunicación más adecuados para ser incorporados funcionalmente en el proceso Scrum, apoyándose en los análisis realizados con anterioridad. Este paso garantiza que los mensajes intercambiados durante los *Sprints* sean eficaces, pertinentes y claros.

Fase 2. Desarrollo del Modelo

Esta etapa se centra en la construcción conceptual y operativa del modelo de comunicación. Aquí se formaliza el modelo de comunicación adaptado a Scrum y en el desarrollo de la aplicación Comunica-Scrum.

5. Diseño del Modelo

El diseño del modelo implicó definir su estructura, sus componentes y la manera en que

estos se integran a los procesos propios de Scrum. Asimismo, se establecen las relaciones entre los actores del proceso comunicativo y las condiciones de uso del modelo.

6. Implementación del modelo adaptado a los procesos de Scrum

Una vez establecido el diseño conceptual, se procedió a documentar su funcionamiento e integrar sus elementos a los flujos del marco ágil. Esto incluyó la definición de responsabilidades, el uso de canales específicos y la descripción de las tareas comunicativas que cada rol debe ejecutar durante el desarrollo del proyecto.

7. Diseño y Desarrollo de la aplicación Comunica-Scrum

Como parte del proceso, se diseñó y se creó una aplicación Web, esta herramienta posibilita la implementación del modelo y simplifica el manejo de la comunicación en el equipo. La aplicación se desarrolló mediante ciclos iterativos que abarcaron la definición de requerimientos, el diseño de interfaces, la programación, las pruebas y los ajustes necesarios para garantizar su funcionamiento.

Fase 3: Evaluación del modelo

La finalidad de esta etapa es verificar el modelo de comunicación adaptado y la herramienta tecnológica vinculada, analizando su relevancia, transparencia, operatividad y utilidad en situaciones reales de desarrollo de software.

8. Evaluación del modelo por estudiantes en un ambiente controlado

El modelo se implementó primero en un ambiente académico para evaluar su comprensión, aplicabilidad y funcionamiento operativo antes de probarlo en un contexto empresarial.

9. Evaluación del modelo mediante expertos, profesionales en Scrum

La evaluación del modelo de comunicación adaptado a Scrum, tanto en entornos de Desarrollo Global de Software (DGS) como en equipos que trabajan en la misma ubicación, se llevó a cabo en tres etapas sucesivas y complementarias: (a) validación del contenido del instrumento a través del método Delphi con la participación de expertos del área, (b) aplicación del instrumento ya validado a un grupo de profesionales del sector del desarrollo de software, (c) validación empírica mediante su implementación en un caso real dentro de la empresa. Posteriormente, se realizó una encuesta dirigida específicamente al equipo Scrum que participó en dicha validación empírica.

10. Evaluación mediante estudio de caso y encuestas

Se lleva a cabo el modelo en un proyecto real de una compañía (estudio de caso). El equipo Scrum en cuestión emplea el modelo y la aplicación Comunica-Scrum; después, se realizan encuestas para analizar su rendimiento, eficacia y utilidad en la práctica.

11. Análisis de los resultados obtenidos

Se analizan los datos producidos en las tres fases de evaluación tanto cualitativa como cuantitativamente, comprobando los indicadores de contribución, aceptabilidad y validez del modelo.

12. Publicación de resultados

Los resultados alcanzados que muestran las contribuciones del proyecto se documentan y se divulgan mediante artículos científicos, informes académicos y presentaciones técnicas, a la comunidad profesional y académica.

4.1 Análisis del modelo

Es conveniente destacar que el primer paso de la metodología de la solución propuesta, consistente en el análisis y revisión del estado del arte, se presentó en el Capítulo 3. Los pasos subsiguientes, específicamente el paso 2 estudio de diversos modelos de comunicación y el paso 3 análisis detallado de los procesos del marco de trabajo Scrum, fueron tratados en el Capítulo 2.

A partir de estos fundamentos previos, el presente apartado aborda el paso 4 de la metodología: análisis y selección de los elementos de comunicación que se integrarán en el proceso Scrum.

Elegir los elementos de comunicación que se incluirán en el proceso Scrum es un paso esencial para asegurar que el modelo sea efectivo. En esta fase, se analizan los elementos que benefician la interacción entre los miembros del equipo, tomando como referencia las etapas anteriores: el estado del arte, los modelos teóricos y los procesos propios de Scrum.

El propósito es integrar únicamente los elementos que favorezcan la claridad, la trazabilidad y la coherencia de los mensajes durante los *Sprints*. Esta revisión permite identificar factores que suelen obstaculizar la comunicación y que, como señala [71], pueden derivar

en desviaciones relevantes respecto a los requerimientos originales del proyecto. Entre los principales problemas detectados se encuentran los siguientes:

Alcance mal definido o cambiante. Si los requisitos iniciales del proyecto no se definen clara y completamente desde el principio, o si sufren cambios significativos durante el desarrollo, es probable que los objetivos originales no se cumplan. Los cambios en el alcance pueden resultar en la omisión de características clave o en la adición de nuevas, lo que puede afectar el cumplimiento de los requisitos iniciales.

Falta de comprensión de las necesidades del cliente. Si el equipo de desarrollo no comprende adecuadamente las necesidades y expectativas del cliente, es probable que los requisitos no reflejen con precisión lo que el cliente realmente quiere o necesita.

Comunicación deficiente con el cliente. La falta de una comunicación efectiva con el cliente durante todo el proyecto puede llevar a malentendidos y a la interpretación errónea de los requisitos. Esto puede resultar en la entrega de un producto que no satisface las expectativas del cliente.

Cambios frecuentes en los requisitos. Los cambios continuos en los requisitos durante el desarrollo pueden dificultar que el equipo cumpla con los requisitos iniciales. Cada cambio requiere tiempo y recursos adicionales.

Estimaciones inexactas de tiempo y recursos. Si las estimaciones iniciales de tiempo y recursos son incorrectas, es posible que el equipo no tenga los recursos necesarios para cumplir con los requisitos tal como se plantean inicialmente.

Falta de documentación adecuada de requisitos. Si los requisitos no se documentan de manera clara y completa, es probable que haya malentendidos y ambigüedades que dificultan el cumplimiento de estos.

Problemas de calidad del código. Los errores y problemas de calidad del código pueden requerir tiempo adicional para corregir, lo que puede afectar el cumplimiento de los requisitos en términos de tiempo y presupuesto.

Gestión de cambios ineficaz. La falta de un proceso sólido de gestión de cambios puede hacer que los cambios en los requisitos no se evalúen adecuadamente en términos de su impacto en el tiempo y el presupuesto.

Falta de pruebas y validación adecuadas. Si no se realizan pruebas exhaustivas y validaciones con el cliente, es posible que los problemas con el cumplimiento de los

requisitos no se detecten a tiempo.

Recursos insuficientes o mal asignados. La falta de personal apropiado o la distribución ineficaz de los recursos puede dificultar que el equipo se ajuste a los plazos y presupuestos establecidos.

Cambios en la dirección o liderazgo del proyecto. La rotación de líderes o la falta de liderazgo sólido puede llevar a cambios en la dirección del proyecto, lo que puede afectar la continuidad en la ejecución de los requisitos.

Estos problemas se muestran de manera gráfica en la Figura 4.2, mediante un diagrama de Ishikawa.

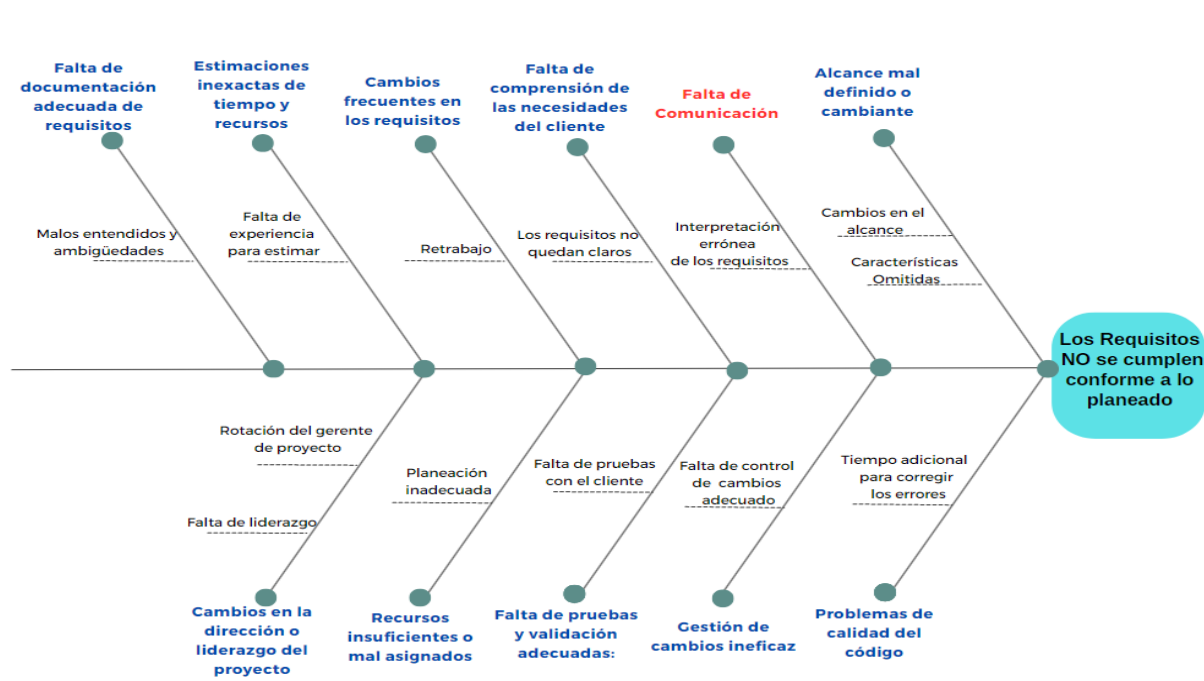


Figura 4.2. Problemas de un proyecto de desarrollo de software que afectan el cumplimiento de los requerimientos iniciales
Fuente: Elaboración propia

Este trabajo se centra en el análisis del problema de la comunicación deficiente entre los actores involucrados en proyectos de desarrollo de software gestionados bajo el marco de trabajo Scrum en entornos distribuidos. En particular, como se ilustra en la Figura 4.3, adaptada de [1], la investigación se enfoca en abordar causas específicas que inciden directamente en dicha problemática, tales como las diferencias de huso horario entre los integrantes del equipo Scrum y las partes interesadas, así como la coexistencia de distintos idiomas.

En el ámbito del desarrollo y la gestión de proyectos de software, uno de los obstáculos más frecuentes surge precisamente de una comunicación insuficiente, poco clara o inadecuada entre las personas que integran el equipo [65].

La Figura 4.4, (figura adaptada de Kennet S. Rubin and Innolution, LLC), muestra los momentos clave del marco Scrum en los que se produce interacción entre los participantes y, por tanto, donde suelen aparecer la mayoría de los fallos comunicativos. El primer punto de contacto relevante ocurre entre el cliente y el *Product Owner*. Este último tiene la responsabilidad de establecer un lenguaje compartido y de mantener la coherencia y claridad de los requisitos que forman el *Product Backlog*.

El siguiente evento es la reunión de *Sprint Planning*, en la que el *Product Owner* explica con detalle cuáles son las funcionalidades prioritarias y las historias de usuario que se van a abordar durante el *Sprint*, con el objetivo de que todo el equipo comprenda perfectamente el propósito del ciclo.

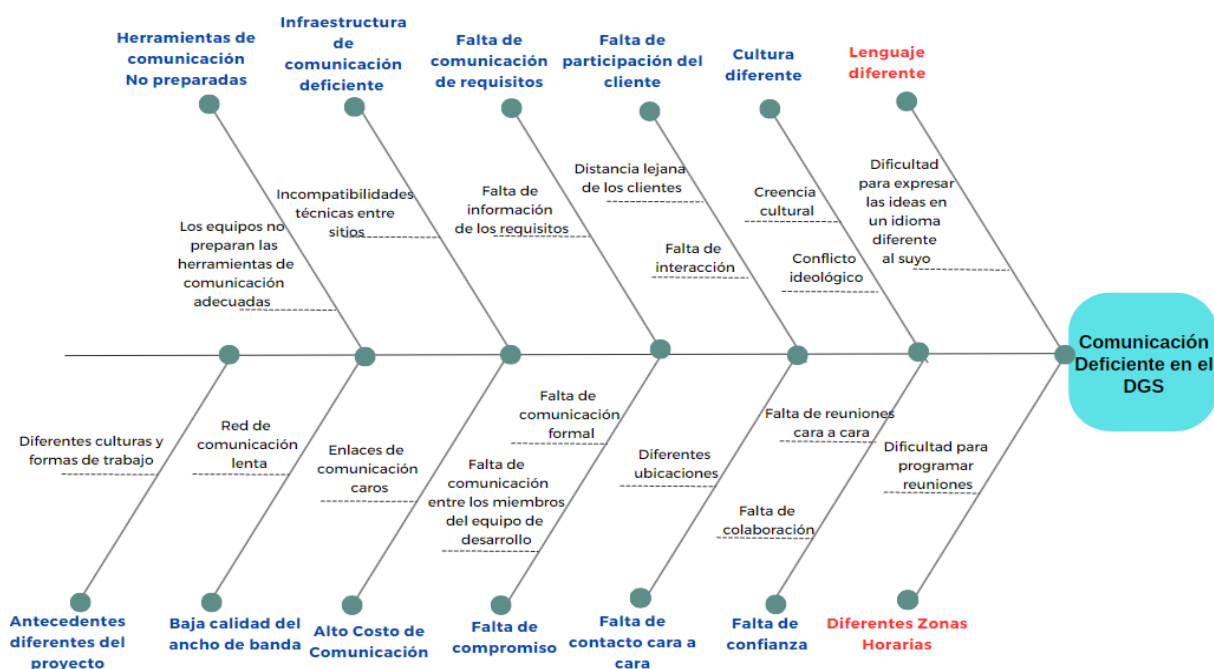


Figura 4.3. Problemas en la comunicación en el desarrollo de Proyectos Scrum en un entorno distribuido, DGS

Otro momento crítico es el *Daily Scrum* o reunión diaria de sincronización, donde la comunicación fluida y la colaboración resultan imprescindibles para mantener la productividad; aquí es fundamental que cada miembro del equipo comprenda lo que necesitan los demás y que la información circule de forma precisa y oportuna

La *Sprint Review* es un momento en el que se establece un amplio diálogo entre el equipo de desarrollo y todas las partes interesadas (*Stakeholders*). Es fundamental contar con estas últimas para identificar a tiempo cualquier problema que pueda presentarse y para obtener sus impresiones acerca del aumento proporcionado.

Por último, la *Sprint Retrospective* concluye el ciclo al permitir identificar oportunidades para mejorar. En esta ceremonia se promueve una comunicación abierta y honesta entre el *Product Owner*, el *Scrum Master* y el resto del equipo de desarrollo, con el fin de garantizar que los mensajes se transmitan correctamente y que las lecciones aprendidas se traduzcan en acciones concretas para los siguientes *Sprints*.

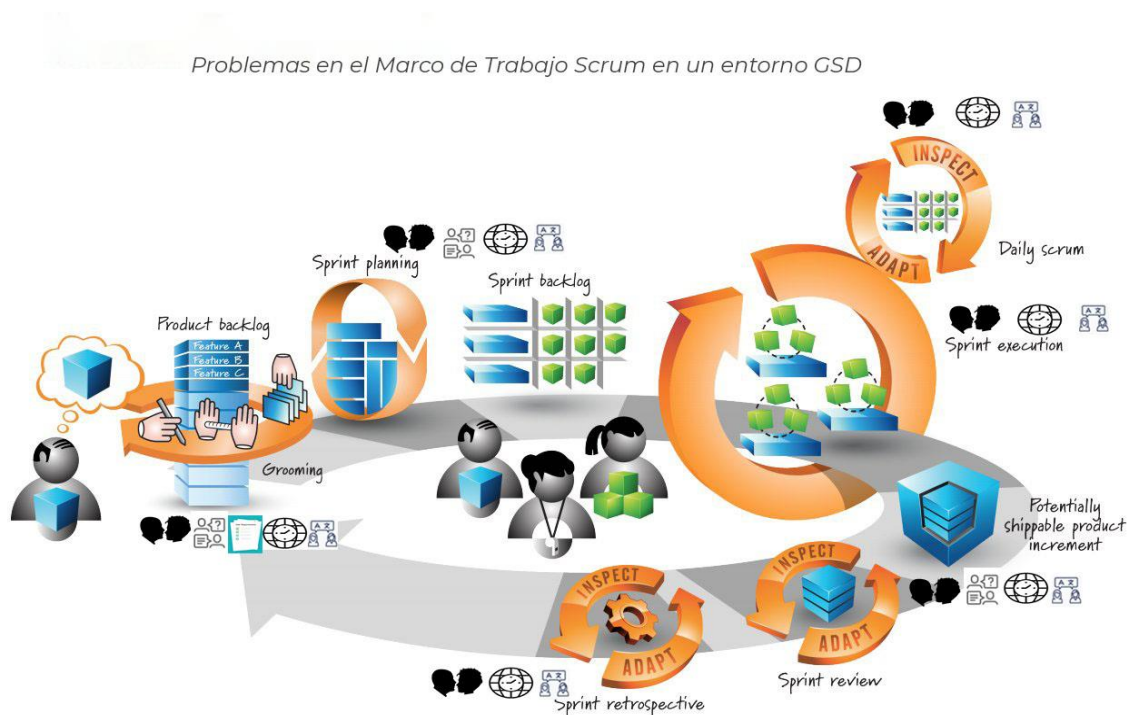


Figura 4.4. Reconocimiento de las ceremonias de Scrum donde interviene la comunicación

En todos los eventos de Scrum, señalados anteriormente surgen dificultades comunicativas recurrentes, principalmente cuando los equipos están distribuidos geográficamente en distintas zonas horarias, con barreras de idioma y diferencias culturales. Estos obstáculos generan ambigüedad en los requisitos, escasa implicación del cliente, dificultades para manejar la incertidumbre y detectar riesgos a tiempo, lo que termina impactando directamente en el alcance, el costo y los plazos del proyecto [1] .

Por esa razón, se plantea la adaptación de un modelo de comunicación efectiva que incida de forma directa en cada ceremonia de Scrum, con el objetivo de aumentar las

probabilidades de cumplir los requisitos definidos al comienzo del proyecto. El modelo busca fomentar una comunicación clara y oportuna entre personas y procesos en proyectos de software desarrollados bajo Scrum, tanto en entornos de desarrollo global de software (DGS) como en equipos que trabajan en la misma ubicación.

Dependiendo del evento Scrum y del rol que ocupe cada participante (emisor o receptor), se determinará el flujo exacto de la información.

La Figura 4.5 presenta la versión adaptada del modelo de comunicación efectiva de Berlo, diseñada específicamente para su aplicación en proyectos gestionados con el marco Scrum en entornos de desarrollo global de software (DGS) y en equipos colocalizados.

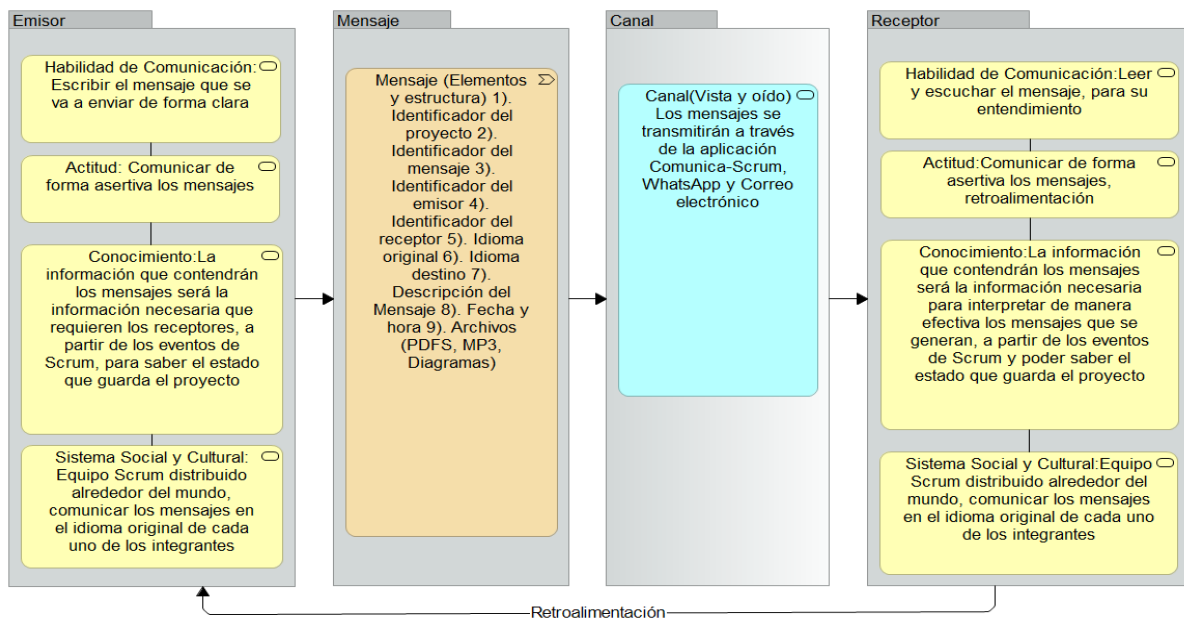


Figura 4.5. Adaptación del modelo comunicativo propuesto por Berlo
Fuente: Adaptada de [22]

Cuando se trabaja en proyectos de software bajo metodología Scrum, ya sea en entornos de Desarrollo Global de Software (DGS) o en equipos que comparten ubicación física, resulta necesario contar con un vocabulario compartido y consistente a lo largo de todo el ciclo del proyecto; esto facilita una comunicación sin ambigüedades y una colaboración más fluida entre miembros distribuidos geográficamente, tal como se recoge con detalle en el Anexo C.

4.2 Diseño del Modelo

En esta fase se establece la estructura básica de la propuesta, sus elementos principales y

cómo éstos interactúan con los procedimientos determinados por Scrum. Este diseño permite especificar cómo se integran los elementos comunicativos dentro del marco de trabajo Scrum, cómo interactúan los distintos roles y en qué condiciones opera el modelo durante el desarrollo del proyecto. A partir de estas decisiones, se construye un marco conceptual que orienta tanto la implementación técnica como la aplicación práctica del modelo en contextos reales.

La Figura 4.6 detalla el esquema conceptual de la metodología desarrollada para mejorar la comunicación en equipos Scrum distribuidos y colocalizados. Esta metodología define un canal de comunicación principal, a través del cual se transmitirán los mensajes que generados por los emisores y dirigidos a los receptores, en el contexto de los eventos que ocurren durante el desarrollo de software, tanto en entornos DGS como en equipos colocalizados. Algunos de estos mensajes se generan de forma automática, incorporando la información estructurada del emisor, el contenido del mensaje y el receptor correspondiente, garantizando así una transferencia eficiente y oportuna de la información crítica del proyecto.

En la parte izquierda de la Figura 4.6 aparece el emisor. Todo comienza con el *Product Backlog*, que actúa como fuente inicial de requisitos; desde allí, una flecha roja señala que los ítems seleccionados disparan la necesidad de comunicación. Dentro del bloque del emisor se muestran los cinco eventos clásicos de Scrum desde la perspectiva de quien genera la información: el refinamiento del *Product Backlog*, la planeación del *Sprint*, la reunión diaria, la revisión del *Sprint* y la retrospectiva. Cada evento produce un mensaje estructurado, representado por un documento con candado, muchos de ellos generados automáticamente con los datos del emisor, el contenido exacto y el receptor correspondiente.

Ese mensaje viaja por la flecha superior, etiquetada como “Envío del mensaje mediante la aplicación Comunica-Scrum”, directamente hacia el receptor, situado a la derecha. El bloque del receptor es prácticamente un espejo del emisor: recibe la misma clase de información para cada uno de los cinco eventos, pero ahora representada con sobres cerrados y candado abierto, resaltando que la entrega es segura y ordenada. En el centro de la figura está el equipo Scrum propiamente dicho, con el *Product Owner* en el núcleo, rodeado por el *Scrum Master* y los desarrolladores, recordándonos que los roles pueden alternar entre emisor y receptor según el momento.

Existe una segunda flecha, en la parte inferior, que regresa del receptor al emisor bajo la

etiqueta “Retroalimentación mediante Comunica-Scrum”. De esta forma se cierra un ciclo completo de comunicación bidireccional que permite confirmar compromisos, aclarar dudas, aprobar cambios o coordinar ayuda de manera rápida y sin perder el rastro.

Cuando el equipo refina el *Backlog*, las propuestas y prioridades viajan automáticamente por el canal hacia los miembros distribuidos; en la *Daily*, cada persona envía su actualización estructurada y el resto del equipo la recibe de inmediato; al terminar el *Sprint Review*, el acta con las decisiones tomadas llega a todos los interesados sin depender de que alguien recuerde enviarla manualmente. Lo mismo ocurre con las acciones de mejora de la retrospectiva.

Entre las ventajas más evidentes de este enfoque están la reducción del ruido comunicacional, la garantía de que la información crítica llega siempre a la persona correcta y en el momento adecuado, la trazabilidad total de lo que se dijo y quién lo recibió, y la posibilidad de automatizar gran parte del flujo. Todo ello sin perder la bidireccionalidad esencial que necesita cualquier equipo ágil para funcionar bien, ya sea que sus integrantes estén en la misma sala o repartidos por diferentes continentes.

El modelo de Berlo es considerado en la solución debido a que otorga un peso central al código, al contenido y al tratamiento que se le da al mensaje, aspectos que están estrechamente ligados a las competencias comunicativas, las actitudes, el conocimiento previo y el entorno sociocultural tanto del emisor como del receptor, justo aquellos elementos que, como se ha visto, generan la mayor parte de los conflictos comunicativos en equipos Scrum. Así, al asegurarse de que el mensaje se codifique en un lenguaje claro y se envíe por la vía más apropiada, se aumenta de manera significativa la probabilidad de una interpretación correcta y, con ello, las posibilidades de éxito del proyecto.

A continuación, se explica cómo se han adaptado los componentes del modelo de Berlo a la realidad de los eventos Scrum en equipos distribuidos globalmente (DGS):

Fuente y Receptor

- **Habilidades comunicativas:** Los mensajes los elaborarán los propios participantes, las partes interesadas o el sistema Comunica-Scrum de forma automática, pero siempre apoyándose en plantillas y estándares previamente definidos que aseguren precisión y claridad.
- **Actitudes:** Cada vez que un mensaje sea recibido y comprendido, el emisor obtendrá una notificación de confirmación, lo que genera mayor confianza y una

actitud más colaborativa entre las partes.

- **Conocimiento:** El flujo continuo y organizado de mensajes mantiene a todos los implicados permanentemente informados tanto del proceso de comunicación como del avance real del proyecto, lo que reduce notablemente los malentendidos.
- **Sistema social y cultural:** Se definirán estándares compartidos de redacción, formatos y convenciones que ayuden a salvar las diferencias culturales; a mayor coincidencia de contexto entre emisor y receptor, más fluida y eficaz será la interacción.

Mensaje

Todo mensaje tendrá un código, un contenido y un tratamiento totalmente estandarizado a través de plantillas predefinidas, pieza clave del modelo planteado, y que permitirá asegurar la claridad, coherencia y trazabilidad de la información que se intercambie, más aún en donde las diferencias culturales, lingüísticas y de zona horaria pueden multiplicar los riesgos de errores. Entre los códigos comunes se pueden encontrar historias de usuario, criterios de aceptación, diagramas UML, tableros de tareas o cualquier otro artefacto cuyo significado y representación hayan sido definidos colectivamente en las ceremonias pertinentes. Al usar plantillas con un propósito específico, estructura y estilo consistentes, se reducen ambigüedades en la comunicación entre el emisor y el receptor, permitiendo una comprensión inmediata del mensaje y fortaleciendo la colaboración a lo largo del proyecto.

Canal

El canal principal será la aplicación web Comunica-Scrum, que se encargará de codificar, enviar, recibir, decodificar y confirmar la llegada de los mensajes, así como de reenviarlos por correo electrónico únicamente a los interesados relevantes. Una de sus características más útiles es la generación de reportes, de modo que toda la información quede disponible de forma instantánea y permanentemente actualizada para quien la necesite.

Como ejemplo el caso particular del evento de la reunión de la revisión del *Sprint* se muestra en la Figura 4.7.

La Tabla 4.1 ofrece una comparación entre el modelo de comunicación original de Berlo y la versión adaptada específicamente al evento Scrum conocido como *Sprint Review* o Reunión de Demostración del *Sprint*.

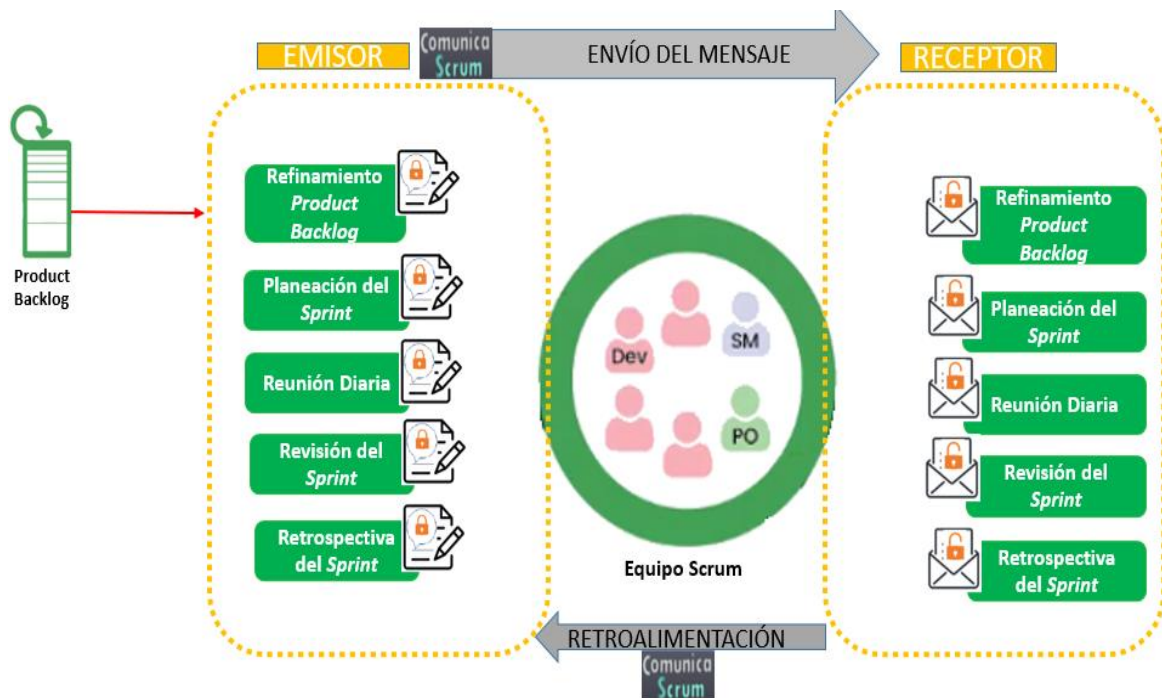


Figura 4.6. Propuesta de solución
Fuente: Elaboración propia

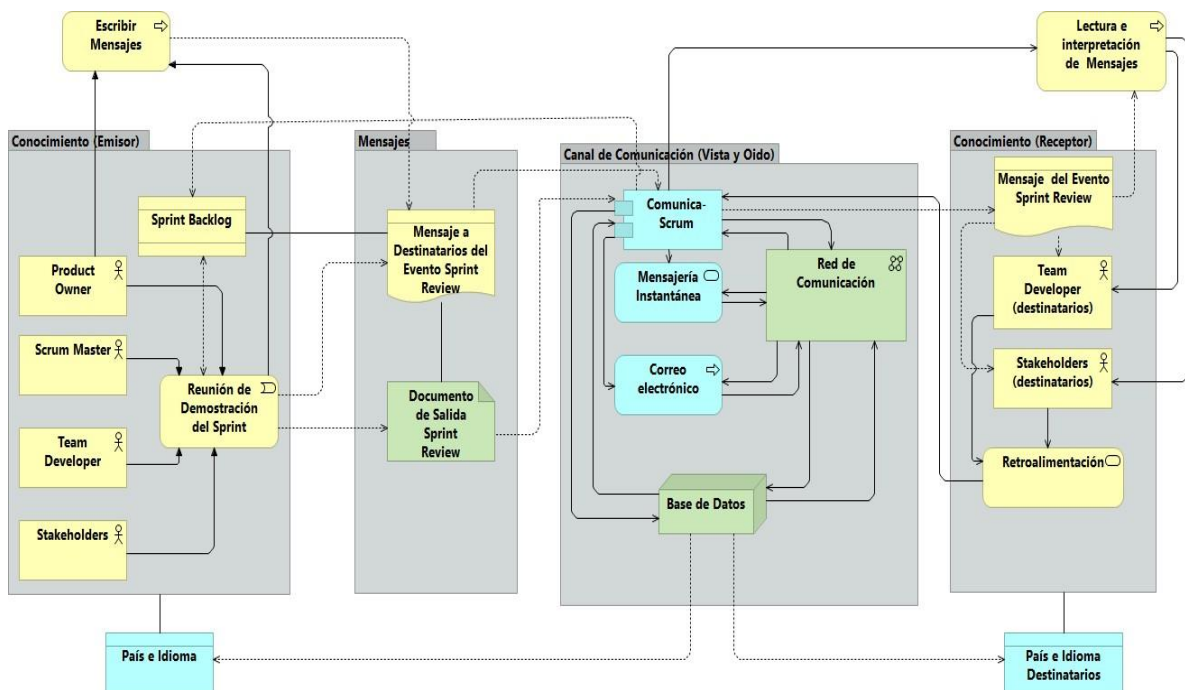


Figura 4.7. Adaptación del modelo de comunicación (Reunión de Revisión del Sprint)

Tabla 4.1 Comparativo Modelo de Berlo y Modelo de Comunicación Adaptado (Reunión de Revisión del Sprint)

Modelo de Comunicación de Berlo	Adaptación del modelo, Evento Reunión de Demostración del Sprint
Fuente o Emisor	
Habilidad de Comunicación	Escribir los acuerdos y acciones a seguir por parte del <i>Product Owner</i> y enviar a los involucrados.
Actitud	Comunicar con claridad y de forma asertiva los acuerdos de la reunión, abordar las áreas de mejora identificadas, además de discutir los aspectos positivos y los desafíos experimentados durante el <i>Sprint</i> .
Conocimiento	El elemento principal es el <i>Sprint Backlog</i> que se utilizará para verificar si las historias de usuario que contiene el <i>Sprint</i> fueron atendidas por el equipo de desarrollo y aceptadas por el <i>Product Owner</i> . El equipo de desarrollo conoce el contenido y estatus del <i>Sprint Backlog</i> , al finalizar la reunión todos los involucrados también deberán conocerlo, además de conocer el avance del proyecto.
Sistema Social y Cultura	Los mensajes deberán mostrarse en el idioma del receptor.
Mensaje	
Elementos	Ver ejemplo de un mensaje en la Figura 4.8, en la Figura 4.9, se muestra un ejemplo del mensaje de retroalimentación.
Contenido	
Estructura	En la Figura 4.10 generada por la aplicación Comunica-Scrum se muestra un ejemplo de una gráfica del avance del proyecto, al finalizar el <i>Sprint</i> .
Código	
Tratamiento	Estatus de las historias de usuario del <i>Sprint Backlog</i>
Canal	
Vista	Aplicación Comunica-Scrum y Correo Electrónico
Oído	
Tacto	N/A
Olfato	N/A
Gusto	N/A

Receptor	
Habilidad de Comunicación	Lectura de los acuerdos y acciones a seguir por parte del <i>Product Owner</i> por parte de todos los involucrados en esta reunión.
Actitud positiva	Comunicar con claridad y de forma asertiva, como parte de la retroalimentación del receptor hacía los involucrados en esta reunión.
Conocimiento	Poder consultar el estatus de cada una de las historias de usuario, del <i>Sprint Backlog</i> y del proyecto en general, además de interpretar de manera efectiva los mensajes que se generan, desde el emisor.
Sistema Social y Cultural	Los mensajes deberán mostrarse en el idioma del receptor.

4.3 Implementación del Modelo

Los mensajes que se producen en cada una de las ceremonias de Scrum, junto con la identificación del emisor, el canal utilizado y los receptores involucrados, fueron examinados considerando los cinco componentes esenciales del mensaje establecidos por el modelo de comunicación de Berlo: elementos, código, estructura, contenido y tratamiento. La Tabla 4.2 expone estos aspectos de manera organizada, ofreciendo una representación clara y sistemática de cómo se desarrolla la interacción comunicativa dentro del marco de trabajo Scrum.

Tabla 4.2 Correlación entre los Componentes de Berlo y las Ceremonias Scrum

Ceremonias Refinamiento		Planeación	Diaria	Revisión	Retrospectiva
Emisor	Product Owner	Product Owner	Team Developer	Product Owner	Scrum Master
Mensajes	1. Descripción de nuevas funcionalidades 2. Preguntas sobre historias, posibles obstáculos 3. Tamaño y complejidad de las historias 4. Prioridad de las HU	1. Historias de Usuario Priorizadas 2. Análisis de viabilidad y dudas 3. Tareas y Metas del Sprint	1. Qué hice ayer y qué haré hoy 2. Impedimentos detectados 3. Recordatorio de objetivos, desbloqueo de problemas	1. Demostración del trabajo completado 2. Opinión sobre el Incremento 3. Ajustes en el Backlog según el feedback	1. Preguntas para analizar el Sprint 2. Qué salió bien, qué mejorar 3. Acciones concretas para el próximo Sprint

Elementos	Historias de usuario refinadas y acuerdos (HU)	Planificación del <i>Sprint</i> y desglose de <i>backlog</i> .	Estado del progreso del <i>Sprint</i> , planificación y obstáculos	Estado del Sprint (HU) completadas, avances, estimaciones, Porcentaje de aceptación de historias) y resultados de la reunión	Reflexión sobre el desempeño del Sprint y propuestas de mejora
Código	Formato estructurado del backlog, lenguaje técnico de Scrum, idioma	Terminología del marco de trabajo Scrum, estructura tabular, idioma	Terminología del marco de trabajo Scrum, estructura estándar de preguntas, uso de terminología técnica, idioma	Terminología del marco de trabajo Scrum, estructura de la plantilla y tablas, idioma	Terminología del marco de trabajo Scrum, estructura estándar de retrospectivas, idioma
Estructura	Organización clara de asistentes, backlog, prioridades, comentarios y Detalles generales (proyecto, fecha, país, idioma).	Organización en secciones: Información general (proyecto, fecha, país, idioma), asistentes, backlog, tareas, objetivos y comentarios	Organización en secciones: Datos generales (proyecto, fecha, país, idioma), asistentes, reportes individuales, obstáculos, planificación	Organización en secciones: Información general (Proyecto, fecha, país, idioma), asistentes, backlog, objetivo, comentarios	Organización en secciones: Datos generales (proyecto, fecha, país, idioma), asistentes, reflexiones, oportunidades de mejora
Contenido	Historias de usuario detalladas con criterios de aceptación, estimaciones y prioridades	Sprint Backlog con historias y tareas, asignación de desarrolladores, estimaciones y objetivos	Actividades completadas, obstáculos técnicos y planificación para el siguiente día	Datos específicos Del Sprint (historias completadas, avances, estimaciones, comentarios).	Evaluación del Sprint, identificación de problemas y definición de acciones correctivas
Tratamiento	Presentación clara mediante tablas, priorización y comentarios explicativos	Presentación clara mediante tablas, uso de prioridades y comentarios acerca de la reunión	Comunicación breve y efectiva, con un formato homogéneo para agilizar la reunión	Presentación optimizada con tablas, comentarios estructurados e indicadores	Comunicación clara y concisa con enfoque en la mejora continua
Canal	Comunica-Scrum, Correo electrónico, Mens. Instantánea	Comunica-Scrum, Correo electrónico, Mens. Instantánea	Comunica-Scrum, Correo electrónico, Mens. Instantánea	Comunica-Scrum, Correo electrónico, Mens. Instantánea	Comunica-Scrum, Correo electrónico, Mens. Instantánea
Receptor	Team Developer, Stakeholders	Team Developer, Stakeholders	Team Developer, Scrum Master, Product Owner	Team Developer, Stakeholders	Team Developer

La Figura 4.8 ilustra la plantilla generada durante la ceremonia de Revisión del Sprint, en la cual se registran los mensajes emitidos por el emisor y aquellos dirigidos al receptor. De manera complementaria, la Figura 4.9 muestra un ejemplo de mensaje de retroalimentación, donde el receptor establece comunicación con el emisor original con el fin de aclarar dudas derivadas del mensaje presentado en la Revisión del *Sprint*. La plantilla correspondiente a esta ceremonia concentra información como el estado de las historias de usuario, los roles involucrados y el objetivo del *Sprint*; es decir, reúne tanto el contenido del mensaje, según los componentes planteados por Berlo, como la estructura con la cual se presenta durante el desarrollo de un proyecto. Los datos incluidos en dicha plantilla

proviene de un proyecto real orientado a la creación de una “plataforma funeraria XYZ”.

Reunión de Revisión del Sprint

Proyecto	Desarrollo de la plataforma Funeraria XYZ									
Fecha y hora	17 de Octubre de 2025 a las 16:00									
Revisión del Sprint 3										
Idioma	Español									
País	México									
Asistentes:	Nombre	Rol	Status	Tipo de Asistencia						
	Jmiquel	Product Owner	Obligatorio	Sincrónica						
	Lorenzo	Developers	Obligatorio	Sincrónica						
	Brayan	Developers	Obligatorio	Sincrónica						
	Samuel	Developers	Obligatorio	Sincrónica						
	Jovany	Scrum Master	Obligatorio	Sincrónica						
	Yesenia	Product Owner	Obligatorio	Sincrónica						
Sprint Backlog	HU	Nombre	Status	Aceptado por PO	%Avance	%Av Real	Estimado	Real	Rest Est.	Rest Real
	HU012	Cálculo de tiempos estimados de servicio	Aceptada	SI	100.00%	100.00%	75	76	0	0
	HU013	Sistema de alertas internas e incidencias	Aceptada	SI	98.67%	100.00%	75	76	1	0
	HU014	Generación de reportes de desempeño	Aceptada	SI	98.33%	100.00%	60	60	1	0
	HU015	Notificaciones automáticas de compra y seguimiento	Incompleta	NO	60.00%	70.67%	75	46	30	22
	HU016	Libro de visitas con mensajes y recuerdos	Aceptada	SI	98.67%	100.00%	75	74	1	0
					90.83%	93.89%				
						Horas:	360	332	33	22
						Días:	45.00	41.50	4.13	2.75
Objetivo de la Reunión	17/10/2025 16:00. Revisión Sprint. Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.									
Comentario de Lorenzo:	Logramos completar el módulo de cálculo de tiempos de servicio y la interfaz de alertas internas. Las validaciones de cronogramas funcionan en tiempo real y se reflejan correctamente en el tablero de incidencias. Tuvimos un pequeño reto con la sincronización de los eventos de alerta, que resolvimos ajustando el componente de listeners en React.									
Comentario de Brayan:	Completé el backend del sistema de incidencias y el módulo de reportes de desempeño. Los reportes ahora pueden exportarse en formato PDF y CSV desde el módulo administrativo. En la HU015 (notificaciones automáticas), se desarrolló el flujo básico de envío, pero quedó pendiente la conexión con el servicio SMTP de producción. Estuve enfermo 3 días durante el Sprint 3									
Comentario de Samuel:	Implementé la lógica de negocio para las notificaciones automáticas y completé el libro de visitas con publicación y moderación de mensajes. La interfaz permite al usuario dejar condolencias y compartir recuerdos, con filtros por fecha y relación con el difunto.									
Comentario de Jovany:	El equipo mantuvo buena coordinación y compromiso. Logramos cumplir con cuatro historias completas y una parcialmente.									

Figura 4.8. Ejemplo práctico de una plantilla generada para la Ceremonia de Revisión del Sprint

La Tabla 4.3 describe la estructura del mensaje para la Revisión del *Sprint*, que compila la información clave que se debe comunicar al final de cada iteración. Para garantizar que el mensaje sea correctamente identificado, la plantilla comienza con información general del proyecto: nombre, país, idioma, número de *Sprint* y fecha de la reunión. Asimismo, incluye el registro de los asistentes, especificando su nombre, rol, condición de participación y modalidad de asistencia; esto posibilita la documentación exacta de quién participó en la revisión.

En la parte central del mensaje se encuentran las historias de usuario que han sido evaluadas, las cuales incluyen su clave, nombre, estado final, aceptación por parte del

Product Owner, porcentajes de progreso y horas estimadas, efectivas y restantes. Estos rasgos posibilitan que el progreso del *Sprint* y el esfuerzo realizado por el equipo se comuniquen con claridad. Asimismo, se indica el porcentaje total de progreso del *Sprint* e incluye la meta definida al principio, lo cual hace más fácil comparar los resultados logrados con el objetivo propuesto. Por último, la plantilla incluye un espacio para las observaciones de los participantes y la incorporación de una gráfica de progreso y capturas de pantalla, que son elementos visuales que respaldan la evidencia del trabajo ejecutado.

Tabla 4.3 Descripción de la plantilla de la Ceremonia de Revisión del Sprint

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Programación de la reunión de demostración del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm
Número del Sprint	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Asistentes	Listado de los asistentes a la reunión:
	Nombre: Nombre de la persona, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
	Status: Indica si su asistencia es obligatoria u opcional, valor numérico.
	Tipo de asistencia: Valores permitidos Síncrona o Asíncrona.
Historias de usuario	Listado de las historias de usuario del <i>Sprint Backlog</i> .
	Clave: Clave de la historia de usuario, formato HUddd, ddd significa que son tres dígitos.
	Nombre: Nombre de la historia de usuario, longitud máxima 80
	Status: Estado de la historia de usuario al finalizar el Sprint, valores permitidos: En <i>Sprint</i> , Sin atender, En progreso, Completada, Aceptada.
	Status del Criterio de Aceptación: Se utiliza para que el Product Owner indique si la historia de usuario es Aceptada o NO, valores permitidos: SI y NO.
	%Avance: Porcentaje de avance planeado de la historia de usuario con el que finalice al terminar el <i>Sprint</i> .
	%Avance Real: Porcentaje de avance real de la historia de usuario.
	Hrs. Estimadas: Horas estimadas, valor numérico entero.

	Hrs. Reales: Cantidad de horas reales dedicadas a la historia de usuario durante el <i>Sprint</i> .
	Hrs. Restantes Estimadas: Horas restantes estimadas, según lo planeado
	Hrs. Restantes Real: Horas restantes reales, de acuerdo a los desarrolladores
	Real: Esfuerzo real aplicado a la historia de usuario durante el <i>Sprint</i> .
Porcentaje de avance del Sprint, estimado y real	Valor numérico en formato de porcentaje que indica el porcentaje de avance real y estimado al finalizar el <i>Sprint</i> en revisión.
Objetivo del Sprint	Es la descripción del objetivo del <i>Sprint</i> , este se establece al iniciar el <i>Sprint</i> , longitud máxima de 500 caracteres.
Comentarios	Comentarios surgidos durante la reunión por parte de los asistentes, longitud máxima de 2000 caracteres.
Gráfica de avance	Imagen en formato PNG o JPG que indica el avance del <i>Sprint</i>
Captura de pantallas	Imágenes de las historias usuario que se atendieron durante el <i>Sprint</i>

La plantilla del mensaje de retroalimentación que se crea por parte de algunos de los involucrados después de la revisión del *Sprint* se especifica en la Tabla 4.4. Este mensaje mantiene la misma filosofía ordenada y confiable del canal Comunica-Scrum, pero ahora en dirección contraria. Registra los datos básicos del proyecto (nombre, *Sprint*, país, idioma y fecha/hora de la respuesta), determina con claridad quién es el emisor de la retroalimentación (su nombre y su rol) y enumera a quienes va dirigida específicamente. Los dos campos fundamentales son el contenido principal:

Proyecto	Creación de una aplicación WEB					
Fecha y hora	04/03/2023 10:00					
Retroalimentación de la Revisión del <i>Sprint</i> n						
País	Estados Unidos					
Idioma	Inglés					
Emisor	Pedro Páramo	Stakeholder				
Destinatarios	Nombre	Rol				
	Jorge Hernández	Product Owner				
Retroalimentación	Porque no se concluyó la Historia de usuario HU005					
Respuesta	Una de las tareas de la Historia de usuario HU005, estuvo mal estimada, por lo que no se logró terminar en este <i>Sprint</i>					

Figura 4.9. Ejemplo de una plantilla de Retroalimentación de la Ceremonia de Revisión del *Sprint*

- **Retroalimentación:** Aprobaciones, comentarios o dudas del receptor.
- **Respuesta:** Espacio para la respuesta oficial de la persona destinataria.

Así, cualquier observación que se haga después de la revisión quedará documentada, registrada y enviada a través del mismo canal seguro, manteniendo el flujo Comunica-Scrum. El resultado es un ciclo cerrado de comunicación (revisión → retroalimentación → respuesta) que asegura que todos estén incluidos en la conversación, sin importar su ubicación o zona horaria, y elimina las conversaciones dispersas.

Tabla 4.4 Descripción de la plantilla de la Retroalimentación de la Ceremonia de Revisión del Sprint

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Fecha y hora de la retroalimentación por parte de algunos de los destinatarios del mensaje de la reunión de demostración del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm
Número del Sprint	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Emisor	Nombre: Nombre del emisor del mensaje, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del emisor, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Destinatarios	Listado de los destinatarios a los que se les dirige el mensaje:
	Nombre: Nombre del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Retroalimentación	Comentarios y/o dudas del receptor del mensaje de la reunión de demostración del <i>Sprint</i> , longitud máxima de 2000 caracteres.
Respuesta	Respuesta normalmente por parte del <i>Product Owner</i> , longitud máxima de 2000 caracteres.

La gráfica Burndown (véase la Figura 4.10) de horas reestimadas correspondiente al Sprint 3 (29-09-2025 al 17-10-2025) muestra el comportamiento del trabajo pendiente a lo largo del Sprint, comparando la línea ideal con el avance real del equipo.

En términos generales, se observa que durante los primeros días del Sprint la línea real se mantiene por encima de la línea ideal, lo que indica que el equipo avanzó a un ritmo ligeramente menor al planificado inicialmente. Esto sugiere posibles ajustes tempranos,

como una subestimación inicial de tareas, actividades no previstas o una curva de adaptación al inicio del Sprint. Sin embargo, a partir de la mitad del periodo se aprecia una convergencia progresiva entre ambas líneas, reflejando una mejora en la velocidad de ejecución y una mayor alineación con el plan establecido.

Hacia la fase final del Sprint, la pendiente de la línea real se vuelve más pronunciada, lo que evidencia un incremento en la productividad y un cierre efectivo de las tareas pendientes. Finalmente, el Sprint concluye con un nivel de horas restantes muy cercano a cero, lo cual indica que el equipo logró cumplir prácticamente con el trabajo comprometido. En conjunto, la gráfica refleja un Sprint con ajustes y reestimaciones durante su desarrollo, pero con una gestión adecuada del esfuerzo, permitiendo alcanzar los objetivos planteados dentro del periodo establecido.

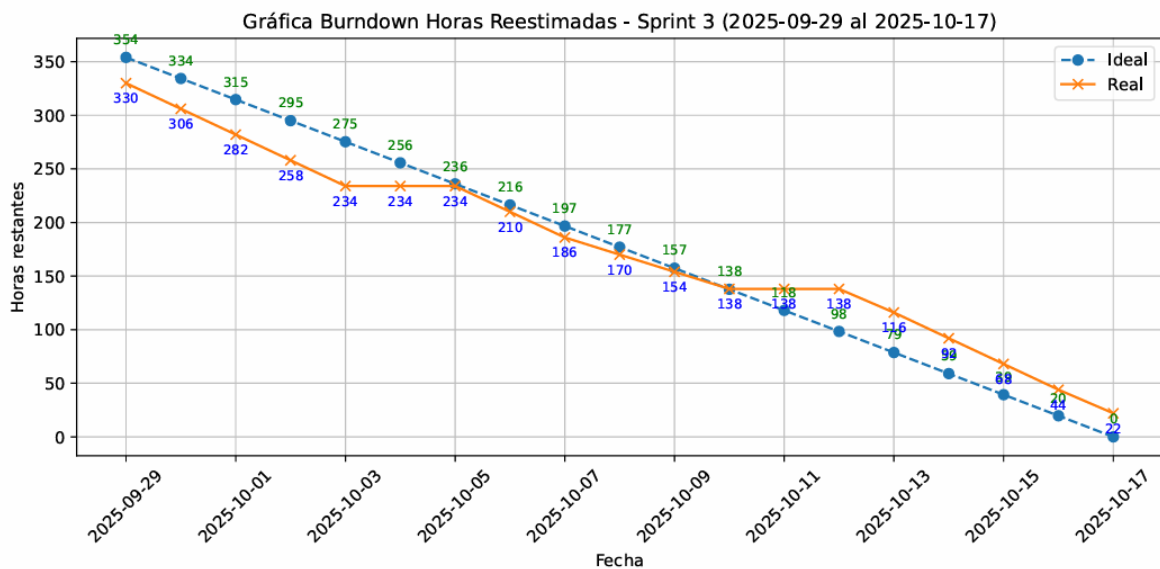


Figura 4.10. Gráfica Burndown, esfuerzo real VS planeado en el Sprint

El Anexo B incluye los diagramas de cada una de las ceremonias Scrum junto con ejemplos reales de los mensajes que se generan durante su desarrollo.

4.4 Diseño y desarrollo de Comunica-Scrum

Como parte de la adaptación del modelo de comunicación efectiva de Berlo al marco Scrum, se han diseñado dos diagramas de clases principales que cubren la gestión de proyectos y

la gestión de mensajes.

En concreto, la Figura 4.11 presenta el diagrama de clases encargado de controlar la administración de los proyectos Scrum, con especial énfasis en garantizar una comunicación efectiva tanto en equipos distribuidos como en los colocados, en el anexo F se muestran las clases para la gestión de proyectos Scrum con sus atributos y métodos.

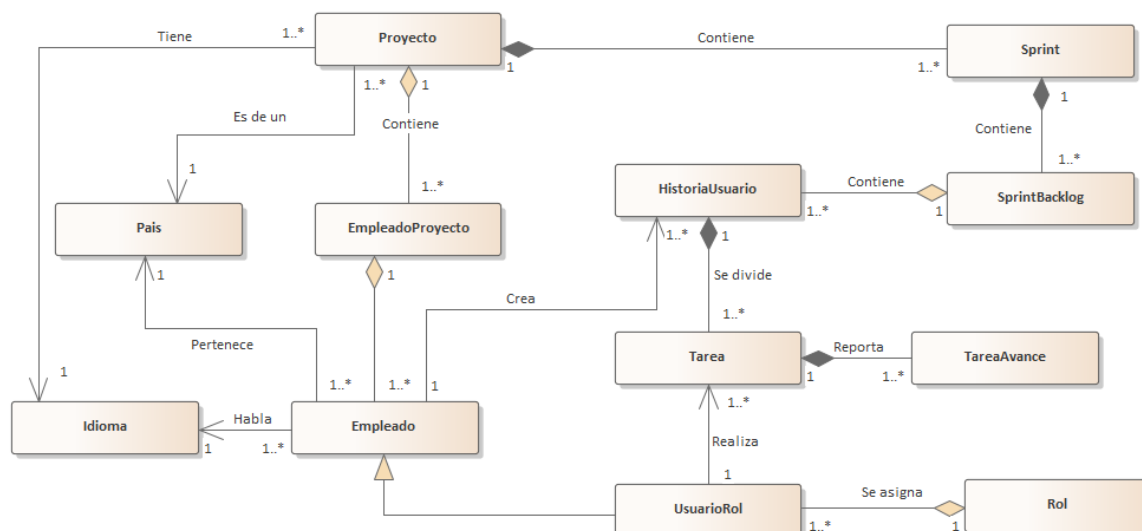


Figura 4.11. Modelo de clases aplicado a la gestión de proyectos con Scrum
Fuente: Elaboración propia

La Tabla 4.5 recoge las clases que se han definido para gestionar los proyectos Scrum aplicando el modelo de comunicación de Berlo.

Tabla 4.5 Clases para la administración de proyectos Scrum

Clase	Descripción
Proyecto	Administra los proyectos Scrum: registra fechas de inicio y fin, y mantiene alineada la visión compartida.
País	Maneja el catálogo de países.
Idioma	Administra el catálogo de idiomas disponibles.
ProyectoUsuarios	Relaciona los usuarios que forman parte de cada proyecto.
Usuario	Representa al personal de la organización que actúa como parte interesada en el desarrollo del software.
HistoriaUsuario	Controla las historias de usuario de un proyecto, incluyendo prioridad y estimación de horas.
ProductBacklog	Agrupar y ordena todas las historias de usuario del proyecto.
Tarea	Permite desglosar una historia de usuario, asignarla a un responsable y a un rol, y comparar fechas planeadas vs. Reales.

Rol	Gestiona los roles que participarán en un Proyecto Scrum.
UsuarioRol	Administra los roles que pueden existir dentro de un proyecto Scrum.
TareaAvance	Registra el avance de tareas, historias de usuario y sprints mediante las horas invertidas.
Sprint	Gestiona cada sprint: objetivo, fechas de inicio y fin.
SprintBacklog	Contiene las historias de usuario seleccionadas para un sprint concreto.

Para hacer posible la interacción comunicativa entre emisores y receptores se emplea el diagrama de clases que aparece en la Figura 4.12, en el anexo F se muestran las clases que representa la comunicación entre emisor y receptor con sus atributos y métodos.

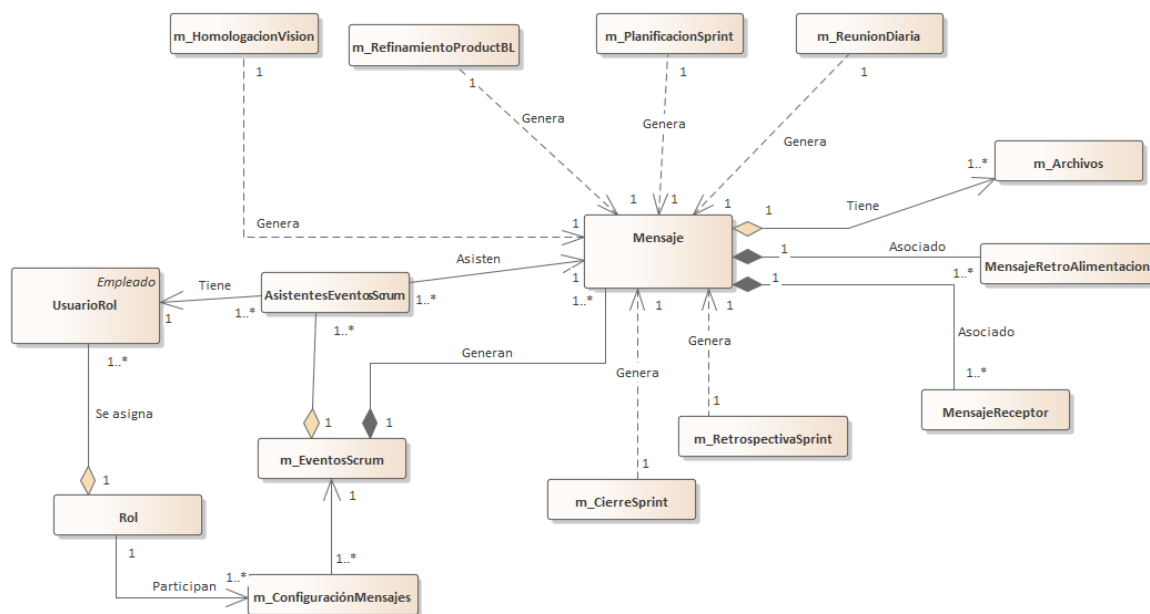


Figura 4.12. Diagrama de clases que representa la comunicación entre emisor y receptor
Fuente: Elaboración propia

En la Tabla 4.6 se describen las clases involucradas en el diagrama de la Figura 4.12.

Tabla 4.6 Clases encargadas de gestionar la comunicación entre emisores y receptores

Clase	Descripción
m_HomologacionVision	Administra los mensajes relacionados con la alineación y validación de la visión del proyecto entre emisor y receptores.
m_RefinamientoProductBL	Controla los mensajes generados durante las sesiones de refinamiento del Product Backlog.
m_PlanificacionSprint	Maneja los mensajes intercambiados en la reunión de planificación del Sprint entre emisor y receptores.

m_ReunionDiaria	Registra y gestiona los mensajes producidos en las reuniones diarias a lo largo del Sprint.
m_CierreSprint	Administra los mensajes correspondientes al cierre del Sprint (incluida la revisión o demo)
m_RetrospectivaSprint	Controla los mensajes generados durante la retrospectiva del Sprint entre emisor y receptores.
m_ConfiguracionMensajes	Sirve para configurar los mensajes de cada uno de los eventos de Scrum, indicando quien es el tipo de emisor y quienes son los tipos de destinatarios.
Mensaje	Gestiona todos los mensajes generados por el emisor de los eventos Scrum.
MensajeReceptor	Administra los mensajes que son enviados a los receptores de los eventos Scrum.
m_Archivos	Administra los archivos que tiene asociado el mensaje, de todos los eventos Scrum.
m_MensajeRetroalimentacion	Gestiona los mensajes de retroalimentación que los receptores harán a los mensajes que le envió el emisor.
m_EventosScrum	Gestiona el catálogo de los eventos de Scrum.
AsistenteEventoScrum	Indica los usuarios que deberán participar en cada uno de los eventos Scrum

En el Anexo D se presenta la descripción completa de la aplicación web Comunica-Scrum, incluyendo sus principales funcionalidades y características.

4.5 Evaluación del Modelo

La evaluación del modelo de comunicación adaptado al marco Scrum en entornos de DGS y en equipos colocalizados se realizó en tres fases complementarias: (a) validación del contenido del instrumento mediante la técnica Delphi con la participación de expertos del ámbito, (b) aplicación del instrumento validado a profesionales del sector, y (c) validación empírica, primeramente del modelo en entornos académicos y posteriormente mediante un estudio de caso real en la empresa Funeraria XYZ utilizando la aplicación Comunica-Scrum.

Esta estrategia de validación mixta permitió asegurar tanto la calidad metodológica del instrumento, así como la validez teórica, técnica y operativa del modelo desarrollado.

4.5.1 Validación del instrumento mediante el método Delphi (por expertos)

El método Delphi garantiza la validez de contenido del instrumento diseñado. Esta técnica

estructurada recopila y consolida opiniones de expertos mediante rondas iterativas y anónimas hasta alcanzar consenso [29].

El panel estuvo conformado por diez expertos: ocho especializados en desarrollo de software ágil y dos en estadística. Con el propósito de facilitar la comprensión del modelo, se proporcionó a los expertos una presentación interactiva explicativa del modelo adaptado de Berlo al marco Scrum que detallaba la estrategia de adaptación, permitiéndoles familiarizarse con sus elementos clave antes de proceder a responder el cuestionario <https://view.genially.com/6633da800bbd36001428862f/presentation-modelo-de-comunicacion>.

El cuestionario consta de 26 ítems (13 sociodemográficos y 13 evaluativos) que valoran pertinencia y claridad de redacción.

Se realizaron dos rondas. En la primera, la mayoría de los *ítems* en relación con la **redacción** de las preguntas obtuvo *coeficientes V de Aiken* superiores a 0.85, destacando P7, P13 y P16 con valores cercanos a 0.972. Los ítems P3, P12, P17 y P19 registraron valores < 0.8, como se muestra en la Figura 4.13, por lo que fueron reformulados.

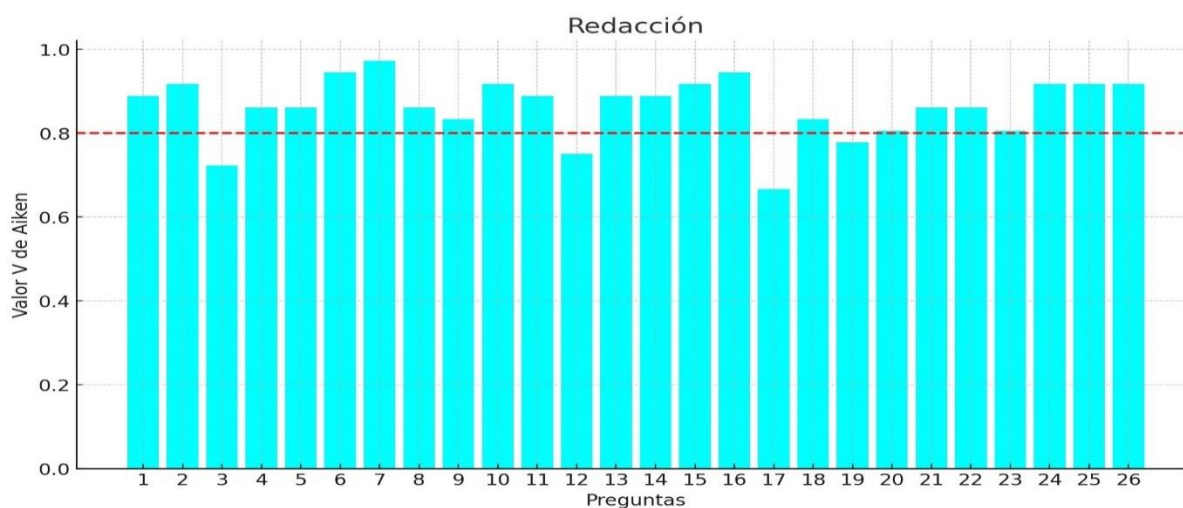


Figura 4.13. Valor V de Aiken de la redacción de las preguntas, 1ª vuelta

En la segunda ronda, tras los ajustes, todos los ítems rebasaron $V \text{ de Aiken} > 0.8$, confirmando alta concordancia en **pertinencia**, ver Figura 4.14 y **redacción**, ver Figura 4.15. Este proceso iterativo aseguró que el instrumento fuera comprensible, pertinente y representativo del constructo teórico.

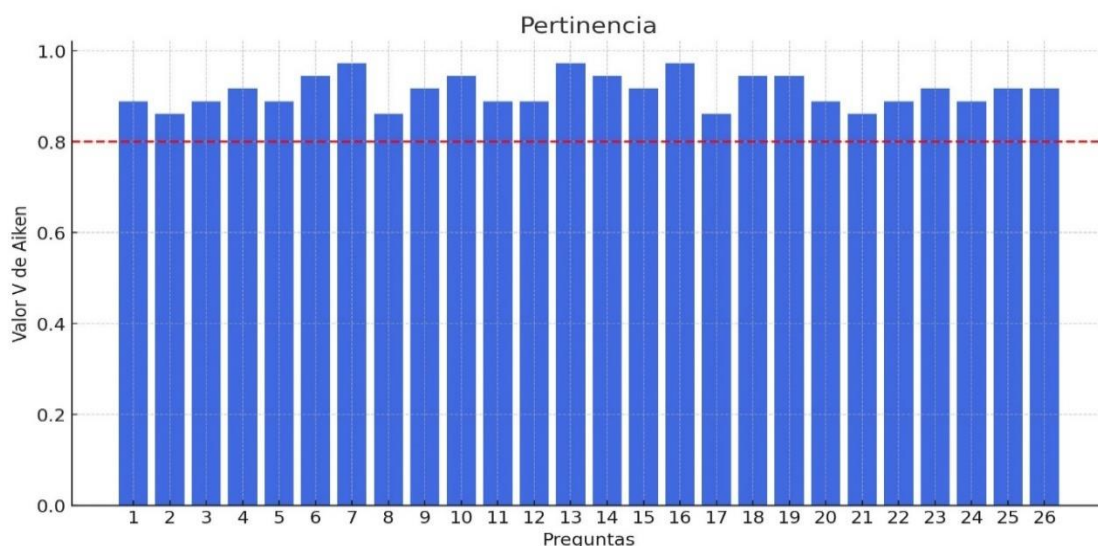


Figura 4.14. Valor V de Aiken de la pertinencia de las preguntas

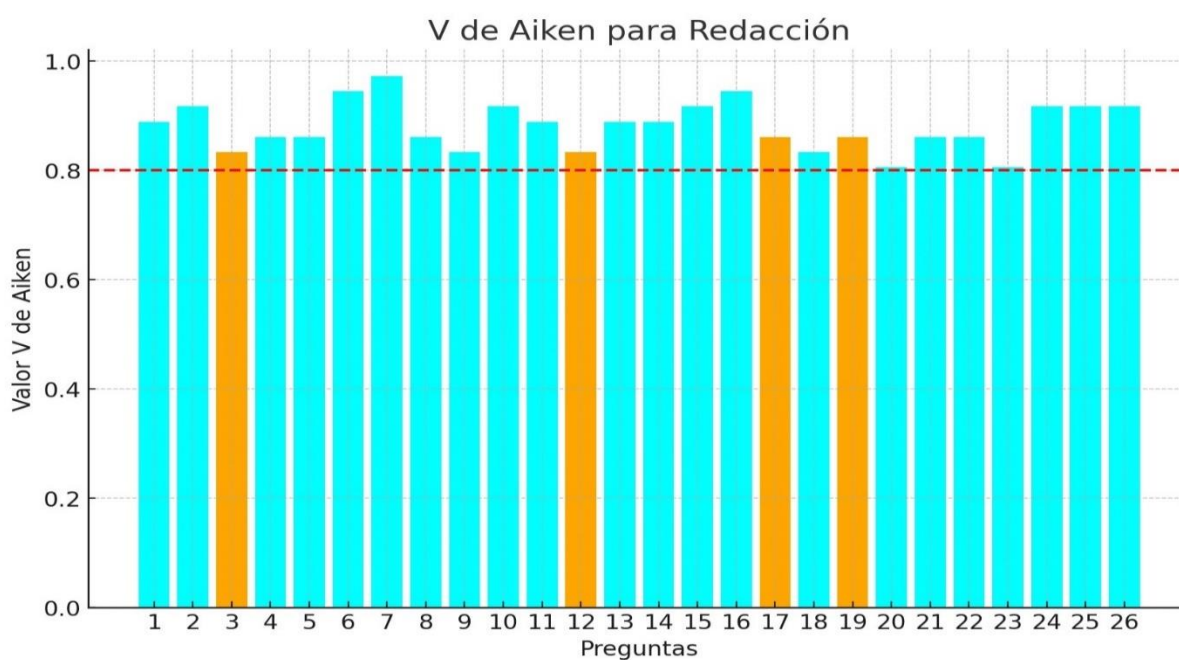


Figura 4.15 Valor V de Aiken de la redacción de las preguntas, 2a vuelta

4.5.2 Aplicación del instrumento a profesionales en desarrollo de software bajo Scrum

Una vez completado el proceso de validación del instrumento en términos de claridad y pertinencia, se procedió a su aplicación en una segunda fase, orientada a obtener una evaluación teórica del modelo. El cuestionario fue distribuido por correo electrónico a un grupo de profesionales con experiencia en desarrollo de software bajo el marco Scrum, con

la finalidad de obtener información sobre su percepción respecto al modelo de comunicación adaptado.

Selección de participantes: El estudio se dirigió a organizaciones dedicadas al desarrollo de software que emplean metodologías ágiles, especialmente Scrum, en contextos con equipos distribuidos. La población incluyó empresas en las que la comunicación constituye un elemento crítico para garantizar la coordinación y el desempeño del proyecto. La muestra estuvo integrada por **44 profesionales provenientes de siete países**, representando un entorno laboral global y diverso, tanto en ubicación geográfica como en idioma. La unidad de análisis correspondió a equipos Scrum distribuidos, evaluando la efectividad de sus procesos comunicativos.

Aplicación de la encuesta: El instrumento se envió de manera digital vía correo electrónico, solicitando a los participantes responder con base en su experiencia profesional y su interacción con equipos Scrum en entornos de Desarrollo Global de Software (DGS). Previamente se les informó que debían conocer el modelo de comunicación propuesto, el cual está descrito en una presentación disponible en la herramienta Genially <https://view.genially.com/66597384363a7d00132887fe/presentation-evaluacion-del-modelo-de-comunicacion-propuesto>.

Muestreo: Se utilizó un muestreo no probabilístico de tipo intencional, considerando únicamente a profesionales con experiencia comprobada en Scrum y desarrollo distribuido, en virtud de que dicho perfil resulta pertinente para evaluar el modelo propuesto.

Diseño del estudio: La investigación adoptó un diseño no experimental, al no manipular variables, enfocándose en la observación y medición de la comunicación en equipos distribuidos. La técnica empleada fue la encuesta, que permitió recolectar datos cuantitativos para describir el fenómeno y detectar fortalezas y áreas de oportunidad en la interacción del equipo.

Instrumento: El cuestionario utilizado incluyó ítems estructurados en escala tipo Likert, diseñados para medir la percepción de los participantes respecto al modelo de comunicación adaptado al contexto ágil, las preguntas asociadas a esta medición se muestran en la Tabla 4.7.

Tabla 4.7 Preguntas para Valorar la Implementación del Modelo de Comunicación en Entornos Scrum

Número	Pregunta
P1	¿En qué medida cree que el modelo de comunicación propuesto pueda mejorar la comunicación en su equipo?
P2	La información generada en cada una de las ceremonias se enviará a cada uno de los involucrados, a pesar de que no asistan a las reuniones por las diferencias de horario o alguna otra situación, ¿en qué medida cree que esto mejore la comunicación entre ellos?
P3	La información generada en cada una de las ceremonias se enviará a cada uno de los involucrados en el idioma de cada uno de los destinatarios, ¿cree que esto mejore la comunicación entre ellos?
P4	¿Cree que haya algún beneficio de llevar un control de los mensajes enviados, indicando si ya fue comprendido o no por parte de los destinatarios?
P5	La implementación del modelo de comunicación propuesto, a través de una aplicación, ¿podría contribuir a que los mensajes transmitidos en las ceremonias Scrum lleguen de manera oportuna y completa a todos los participantes, optimizando la comunicación en su equipo?
P6	¿La frecuencia de los mensajes enviados y recibidos puede promover una retroalimentación para mejorar la comunicación entre el personal de los equipos distribuidos?
P7	¿Considera que la comprensión de los mensajes puede mejorar, una vez que se utilice el modelo de comunicación propuesto?
P8	¿Considera que la colaboración entre el equipo Scrum se puede mejorar, una vez que se utilice el modelo de comunicación propuesto?
P9	¿Existirá una mejora en el acceso a la información, una vez que se utilice el modelo de comunicación propuesto?
P10	¿Considera que la retroalimentación se puede mejorar, una vez que se utilice el modelo de comunicación propuesto?
P11	¿Considera que el uso de plantillas para el envío de mensajes y la retroalimentación puede mejorar la comunicación una vez que se utilice el modelo propuesto?
P12	Los canales de comunicación utilizados, Comunica-Scrum, Correo Electrónico y una Aplicación de Mensajería Instantánea, ¿son adecuados para el tipo de información que se comparte?
P13	¿Recomendarías el modelo de comunicación propuesto a otros equipos Scrum?

Fiabilidad del instrumento: La fiabilidad del instrumento se calculó a partir del coeficiente Alfa de Cronbach, indicador estándar para medir la consistencia interna [31]. El cuestionario utilizado, titulado “Evaluación del Modelo de Comunicación de Berlo Aplicable al Marco Scrum en un Entorno de Desarrollo Global de Software”, obtuvo así una verificación rigurosa de su confiabilidad.

Para el análisis se consideraron únicamente los ítems orientados a evaluar el modelo, excluyendo aquellos de carácter sociodemográfico. El valor obtenido fue **0.862**, lo que

evidencia una consistencia interna alta y adecuada (intervalo óptimo: 0.8–0.9). Este resultado confirma que los ítems son coherentes entre sí y miden de manera fiable el constructo evaluado. El cálculo se realizó mediante software estadístico, utilizando las respuestas de 44 profesionales de siete países, lo que refuerza la estabilidad del instrumento para evaluar el modelo en equipos Scrum distribuidos.

En la Tabla 4.8 se presentan los principales estadísticos descriptivos obtenidos de la encuesta aplicada: media, mediana, moda y desviación estándar de cada ítem.

Tabla 4.8 Estadísticos descriptivos de la encuesta a profesionales en desarrollo de software con Scrum

Pregunta	Encuestados	Media	Desviación estándar	Varianza	Mediana	Moda	Mínimo	Máximo
P1	44	4.136	0.668	0.446	4	4	3	5
P2	44	4.205	0.594	0.353	4	4	3	5
P3	44	4.432	0.728	0.530	5	5	3	5
P4	44	4.364	0.780	0.609	5	5	2	5
P5	44	4.591	0.542	0.294	5	5	3	5
P6	44	4.409	0.542	0.294	4	4	3	5
P7	44	4.409	0.542	0.294	4	4	3	5
P8	44	4.500	0.550	0.302	5	5	3	5
P9	44	4.432	0.587	0.344	4	4	3	5
P10	44	4.523	0.549	0.302	5	5	3	5
P11	44	4.455	0.663	0.440	5	5	3	5
P12	44	4.205	0.765	0.585	4	4	2	5
P13	44	4.386	0.538	0.289	4	4	3	5

En general, las respuestas a la encuesta indican percepciones positivas, con la mayoría de las calificaciones agrupadas en torno a 4 y 5. La media de las respuestas varía entre 4.136 y 4.591, indicando que los encuestados tienden a otorgar calificaciones altas. La media de las respuestas es alta para todas las preguntas, con P5 obteniendo la media más alta (4.591), lo que sugiere una percepción alta para esta pregunta.

La mediana y la moda también refuerzan la tendencia de respuestas altas, siendo 4 o 5 en la mayoría de los casos. La desviación estándar oscila entre 0.542 y 0.780, lo que indica cierta variabilidad en las respuestas, aunque no es excesiva, sigue siendo relativamente baja, indicando un consenso general entre los encuestados.

Los resultados de la encuesta a profesionales del desarrollo de software indican que, en su mayoría, se tiene una opinión positiva sobre el uso del modelo de comunicación de Berlo en el marco Scrum para el Desarrollo Global de Software y equipos colocalizados.

4.5.3 Validación Empírica del Modelo

4.5.3.1 Validación del Modelo de Comunicación en Entornos Académicos

La evaluación del modelo de comunicación se llevó a cabo mediante dos estudios de caso implementados en entornos académicos con estudiantes del CENIDET (México) y de la Universidad Nacional de Colombia (UNAL), Sede Medellín. En el CENIDET, el equipo trabajó durante tres *Sprints* en el desarrollo de una aplicación para la “Evaluación Docente”, utilizando Comunica-Scrum como canal principal de comunicación. Los resultados mostraron una mejora notable entre el Sprint 1 (75 % de avance) y el Sprint 2 (100 %), así como una alta comprensión de los mensajes generados durante las ceremonias Scrum. Aunque se registraron algunos casos que requirieron retroalimentación adicional, estos fueron resueltos oportunamente, evidenciando la utilidad del modelo para promover claridad, seguimiento y colaboración dentro del equipo. Además, la curva de aprendizaje inicial en programación fue superada progresivamente, lo que permitió a los estudiantes adaptarse al flujo de trabajo y aprovechar de manera más efectiva la herramienta.

En el caso de la Universidad Nacional de Colombia (UNAL), el proyecto desarrollado correspondió a la implementación de un “Repositorio de Juegos”. A diferencia del caso de estudio previo, el uso de la herramienta Comunica-Scrum fue limitado, principalmente debido a dificultades percibidas en la experiencia de usuario, confirmándose problemas en el registro de horas y en el acceso a determinadas funcionalidades. Asimismo, se identificó la necesidad de simplificar la interfaz y automatizar ciertos procesos con el fin de facilitar su adopción. Estas observaciones fueron atendidas y constituyeron oportunidades de mejora relevantes para la herramienta, orientadas a incrementar su usabilidad tanto en entornos reales como educativos.

Derivado de lo anterior, se realizaron ajustes a la aplicación Comunica-Scrum orientados a mejorar su usabilidad y funcionalidad. Entre las principales modificaciones, se optimizó el acceso a la captura del esfuerzo dedicado a cada tarea, se actualizaron las imágenes de algunos íconos, se incorporó la gráfica de avance del Sprint (Burndown) y se automatizó la inclusión de archivos asociados, tales como mensajes, plantillas y la gráfica de avance del Sprint, dentro de la opción de envío de mensajes correspondiente a cada ceremonia. Adicionalmente, durante la creación de las reuniones de las ceremonias Scrum, se integró de manera automática el listado de asistentes asignados al proyecto.

En conjunto, ambos estudios permitieron validar el modelo de comunicación en contextos

académicos, destacando su potencial para mejorar la coordinación y claridad informativa en Scrum, así como áreas de mejora relacionadas con la interfaz y facilidad de uso de la plataforma Comunica-Scrum.

4.5.3.2 Estudio de Caso de la Empresa Grupo Funerario

Con el fin de validar el modelo en un entorno real de aplicación, se llevó a cabo un caso de estudio dentro de una organización, cuyo nombre se mantiene en reserva por razones de confidencialidad. Dicho caso de estudio se desarrolló siguiendo las recomendaciones metodológicas establecidas en [92] y tuvo como objetivo evaluar de manera empírica el modelo de comunicación propuesto, mediante el análisis de su impacto en la claridad de los mensajes, la efectividad de los canales de comunicación utilizados, el nivel de comprensión de los receptores y la calidad de la retroalimentación generada durante las distintas ceremonias del marco de trabajo Scrum.

El estudio de caso se desarrolló en torno a la construcción de la plataforma “*Funeraria XYZ*” durante tres *Sprints*, empleando de manera exclusiva la aplicación Comunica-Scrum como canal de comunicación estructurada conforme al modelo SMCR adaptado, así como herramienta de apoyo para el seguimiento y ejecución de las diferentes ceremonias Scrum. El proyecto fue realizado por un equipo distribuido, integrado por un *Product Owner*, un *Scrum Master* y tres desarrolladores, del 18 de agosto al 17 de octubre de 2025.

Descripción del proyecto y resultados operativos.

Se planificaron 16 historias de usuario (HU) distribuidas en tres *Sprints*, con una duración de tres semanas cada uno:

- Sprint 1 (18/08–05/09/2025): 5 HU → 100 % completadas y aceptadas.
- Sprint 2 (08/09–26/09/2025): 6 HU → 100 % completadas y aceptadas.
- Sprint 3 (29/09–17/10/2025): 5 HU → 4 completadas y aceptadas; 1 avanzada al 70.7 % (HU016) y no aceptada por enfermedad de un desarrollador.

Cumplimiento global: 15/16 HU (93.8 %), superando el indicador propuesto de ≥ 90 % de historias de usuario cubiertas por sprint.

Actividad comunicativa registrada

- Mensajes enviados: 57

- Mensajes recibidos: 285
- Retroalimentaciones generadas: 32 → todas atendidas y resueltas por *Product Owner/Scrum Master*.
- Comprensión final de mensajes: 100 %, inicialmente el 11.22% de los mensajes no fueron comprendidos, por lo que fueron generados los mensajes de retroalimentación para que fueran comprendidos en su totalidad.
- Comentarios totales en reuniones: 165 (135 en *Daily Scrum*)

Instrumento y aplicación

Al finalizar la prueba en el tercer sprint se aplicó el cuestionario de 29 ítems (escala Likert 1-5) más una pregunta abierta cualitativa, diseñado y dirigido para los cinco integrantes del equipo Scrum mediante *Google Forms*, previamente de le proporcionó al equipo Scrum una presentación para que conocieran el modelo de comunicación propuesto [Presentación para conocer el modelo de comunicación](#).

Las preguntas analizadas de la encuesta se muestran en la Tabla 4.9

Tabla 4.9 Preguntas al equipo Scrum para validar el modelo de comunicación propuesto

No.	Pregunta
P1	Durante los <i>Sprints</i> , los miembros del equipo expresaron sus ideas con claridad en las reuniones Scrum
P2	El <i>Product Owner</i> comunicó claramente los objetivos de cada <i>Sprint</i> mediante Comunica-Scrum
P3	El <i>Scrum Master</i> fomenta una comunicación abierta y comprensible durante los <i>Sprints</i>
P4	Los emisores adaptan su lenguaje técnico al nivel de comprensión de los receptores
P5	Las instrucciones y comentarios emitidos durante los <i>Sprints</i> fueron precisos y fáciles de seguir
P6	Los mensajes registrados en Comunica-Scrum durante los <i>Sprints</i> contenían la información necesaria para comprender las tareas
P7	La herramienta Comunica-Scrum permitió representar las historias de usuario de forma clara y sin ambigüedades
P8	Las decisiones acordadas durante las reuniones Scrum quedaron correctamente documentadas en Comunica-Scrum
P9	El propósito de cada mensaje compartido durante el proyecto fue claro y comprensible
P10	La información compartida en Comunica-Scrum durante los <i>Sprints</i> fue clara, relevante y sin sobrecarga
P11	Comunica-Scrum facilitó la colaboración del equipo en ceremonias como <i>el Sprint Planning, Daily Scrum, Sprint Review y Retrospective</i>

P12	El envío de la gráfica <i>burndown</i> posterior a las reuniones diarias permitió visualizar el avance de las tareas y de las historias de usuario, contribuyendo a que las actividades asignadas se realizarán en tiempo y forma.
P13	Comunica-Scrum fue un canal adecuado para compartir la información necesaria durante los Sprints
P14	La comunicación asincrónica mediante Comunica-Scrum fue efectiva y oportuna durante los Sprints
P15	Comunica-Scrum permitió mantener una coordinación fluida entre miembros del equipo que trabajan en diferentes ubicaciones
P16	Durante los <i>Sprints</i> , los mensajes enviados a través de Comunica-Scrum fueron comprendidos correctamente por los receptores
P17	Los receptores confirmaron la recepción o comprensión de los mensajes enviados a través de la herramienta Comunica-Scrum
P18	Los miembros del equipo demostraron disposición para escuchar, comprender y aportar durante las reuniones Scrum
P19	El <i>Product Owner</i> y el <i>Scrum Master</i> aseguraron que todo el equipo Scrum comprendieran los objetivos del sprint, mediante Comunica-Scrum
P20	Se promovió la participación y equitativa de los miembros del equipo en las reuniones
P21	La retroalimentación proporcionada mediante Comunica-Scrum contribuyó a mejorar la ejecución de las tareas
P22	Los comentarios registrados en la herramienta fueron constructivos y fomentaron la mejora continua
P23	Los miembros del equipo aplicaron la retroalimentación recibida para mejorar sus entregas
P24	La comunicación dentro del equipo se caracterizó por la colaboración y no por la jerarquía
P25	Durante los tres <i>Sprints</i> , la comunicación del equipo mejoró con el uso de Comunica-Scrum
P26	La estructura del modelo de comunicación facilitó la comprensión de la información compartida
P27	Se redujeron los malentendidos o retrabajos derivados de problemas de comunicación
P28	La coordinación entre los roles del equipo (<i>Scrum Master</i> , <i>Product Owner</i> y equipo de desarrollo) se volvió más fluida a lo largo de los <i>Sprints</i>
P29	El modelo adaptado mostró ser adecuado para entornos distribuidos de desarrollo de software

Confiabilidad

Alfa de Cronbach global = 0.921 (muy alta consistencia interna). Ningún ítem, al eliminarse, incrementó significativamente el alfa, lo que confirma la pertinencia de mantener todos los ítems, ver Figura 4.16.

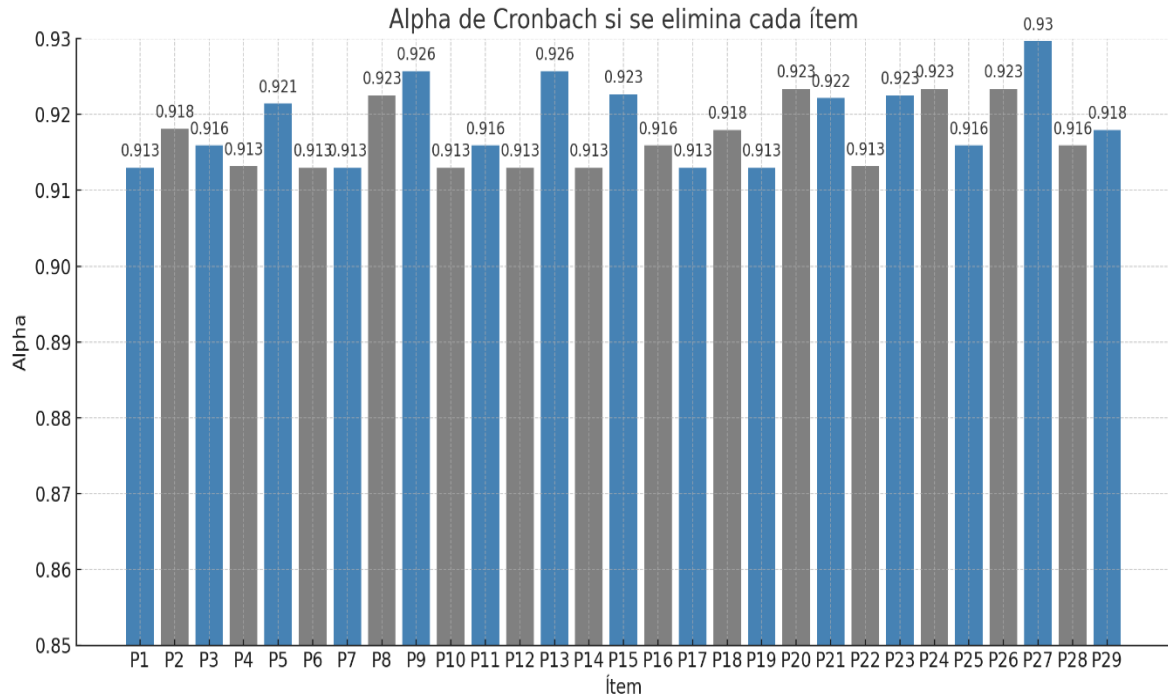


Figura 4.16. Alfa de Cronbach, equipo Scrum al excluir cada ítem individualmente

Estadísticos descriptivos

Se calcularon estadísticos descriptivos (media, desviación estándar, varianza, mediana, moda, mínimo y máximo) para cada ítem y dimensión, con el propósito de evaluar la percepción general de los participantes respecto a los elementos del modelo.

- Medias: 4.3 a 5.0 (máximo 5.0 en P21: “La retroalimentación proporcionada mediante Comunica-Scrum contribuyó a mejorar la ejecución de las tareas”), ver la Tabla 4.10.
- Desviaciones estándar: 0.0 a 0.548 (muy bajas → alto consenso).
- Mínimos: 4 (ningún participante puntuó por debajo de 4 en la mayoría de los ítems; mínimos aislados de 3).
- Moda y mediana: 5 en casi todos los ítems.

Los resultados obtenidos reflejan una percepción altamente favorable por parte de los participantes en todas las dimensiones del modelo evaluado —emisor, mensaje, canal, receptor, retroalimentación y efectividad global—, lo que respalda su utilidad práctica en entornos Scrum distribuidos.

Asimismo, el análisis estadístico realizado permitió validar empíricamente el modelo

propuesto, confirmando que las dimensiones teóricamente definidas se corresponden con la estructura observada en los datos, y que los ítems empleados presentan una alta fiabilidad y coherencia interna en la medición de los constructos establecidos.

Tabla 4.10 Estadísticos descriptivos de la encuesta al equipo Scrum

Pregunta	Encuestados	Media	Desviación estándar	Varianza	Mediana	Moda	Mínimo	Máximo
P1	5	4.6	0.548	0.3	5.0	5	4	5
P2	5	4.4	0.548	0.3	4.0	4	4	5
P3	5	4.8	0.447	0.2	5.0	5	4	5
P4	5	4.4	0.548	0.3	4.0	4	4	5
P5	5	4.8	0.447	0.2	5.0	5	4	5
P6	5	4.6	0.548	0.3	5.0	5	4	5
P7	5	4.6	0.548	0.3	5.0	5	4	5
P8	5	4.6	0.548	0.3	5.0	5	4	5
P9	5	4.8	0.447	0.2	5.0	5	4	5
P10	5	4.6	0.548	0.3	5.0	5	4	5
P11	5	4.8	0.447	0.2	5.0	5	4	5
P12	5	4.6	0.548	0.3	5.0	5	4	5
P13	5	4.8	0.447	0.2	5.0	5	4	5
P14	5	4.6	0.548	0.3	5.0	5	4	5
P15	5	4.4	0.548	0.3	4.0	4	4	5
P16	5	4.8	0.447	0.2	5.0	5	4	5
P17	5	4.6	0.548	0.3	5.0	5	4	5
P18	5	4.6	0.548	0.3	5.0	5	4	5
P19	5	4.6	0.548	0.3	5.0	5	4	5
P20	5	4.6	0.548	0.3	5.0	5	4	5
P21	5	5.0	0.0	0.0	5.0	5	5	5
P22	5	4.4	0.548	0.3	4.0	4	4	5
P23	5	4.6	0.548	0.3	5.0	5	4	5
P24	5	4.6	0.548	0.3	5.0	5	4	5
P25	5	4.8	0.447	0.2	5.0	5	4	5
P26	5	4.6	0.548	0.3	5.0	5	4	5
P27	5	4.8	0.447	0.2	5.0	5	4	5
P28	5	4.8	0.447	0.2	5.0	5	4	5
P29	5	4.6	0.548	0.3	5.0	5	4	5

Las medias obtenidas, que oscilan entre 4.3 y 5.0 en una escala Likert de cinco puntos, reflejan una percepción positiva y homogénea sobre la efectividad del modelo de comunicación propuesto y de la herramienta Comunica-Scrum como medio de soporte. Destaca particularmente el ítem P21, relacionado con la contribución de la retroalimentación a la mejora en la ejecución de las tareas, el cual alcanzó la puntuación máxima por parte de todos los participantes, lo que resalta la relevancia del componente de retroalimentación dentro del modelo adaptado.

Asimismo, las bajas desviaciones estándar registradas en la mayoría de los ítems, junto con valores de mediana y moda iguales a 5, indican un alto nivel de consenso entre los participantes y una percepción compartida sobre la claridad de los mensajes, la adecuación del canal de comunicación y la correcta comprensión de la información intercambiada durante los *Sprints*. La ausencia de puntuaciones mínimas extremas y la concentración de las respuestas en los valores superiores de la escala sugieren que el modelo favoreció una comunicación efectiva, redujo malentendidos y fortaleció la coordinación entre los distintos roles del equipo Scrum. En conjunto, estos resultados refuerzan la validez empírica del modelo propuesto y su aplicabilidad en entornos de desarrollo de software distribuidos y colocalizados.

Capítulo 5 Formalización del Modelo

Las historias de usuario constituyen los elementos centrales en el marco de trabajo Scrum para la comunicación efectiva y el entendimiento colaborativo en equipos distribuidos. Las historias de usuario están involucradas en todos los eventos asociados en dicho marco. En este sentido, se establece que, si el equipo de desarrollo implementa en tiempo y forma las historias de usuario y se obtiene la aprobación, basados en los criterios de aceptación, es un indicativo que se ha logrado una comunicación efectiva entre el equipo Scrum y las partes interesadas involucradas en el proceso, contribuyendo en los objetivos del proyecto de desarrollo de software.

En la presente investigación, se plantea la utilización de autómatas finitos y máquinas de estado con el propósito de representar, mediante un modelo formal, el flujo y los estados que atraviesan las historias de usuario concebidas como mensajes en este enfoque. Así mismo, se busca analizar su interacción en cada una de las ceremonias dentro del marco de trabajo Scrum.

Los autómatas finitos de Harel, propuestos por el científico de la computación David Harel [93], proporcionan una forma estructurada de modelar sistemas reactivos y concurrentes, con capacidad para expresar jerarquías y estados concurrentes, los autómatas finitos de Harel permiten capturar de manera intuitiva la complejidad inherente a sistemas dinámicos.

Los autómatas finitos de Harel y las máquinas de estado de UML son parte integral del desarrollo de la teoría de la computación y la ingeniería de software [94].

En esta investigación se utilizan las máquinas de estados para modelar la transición entre los diferentes estados por las que pasa una historia de usuario en proyectos ágiles, ver Figura 5.1.

La formalización matemática de una máquina de estados implica representar sus componentes, estados y transiciones de manera precisa utilizando conceptos matemáticos. Una máquina de estados se puede formalizar utilizando teoría de conjuntos y notación matemática, esta formalización ofrece una representación visual y semántica que facilita la comprensión y el análisis de la lógica subyacente de un sistema. [95].

Definición: Sea Q un conjunto que representa el conjunto de estados posibles de la máquina.

$Q = \{HUCapturada, HUCancelada, HURefinada, HUSprint, HUTareas, HUProgreso, HUCompletada, HUAceptada\}$

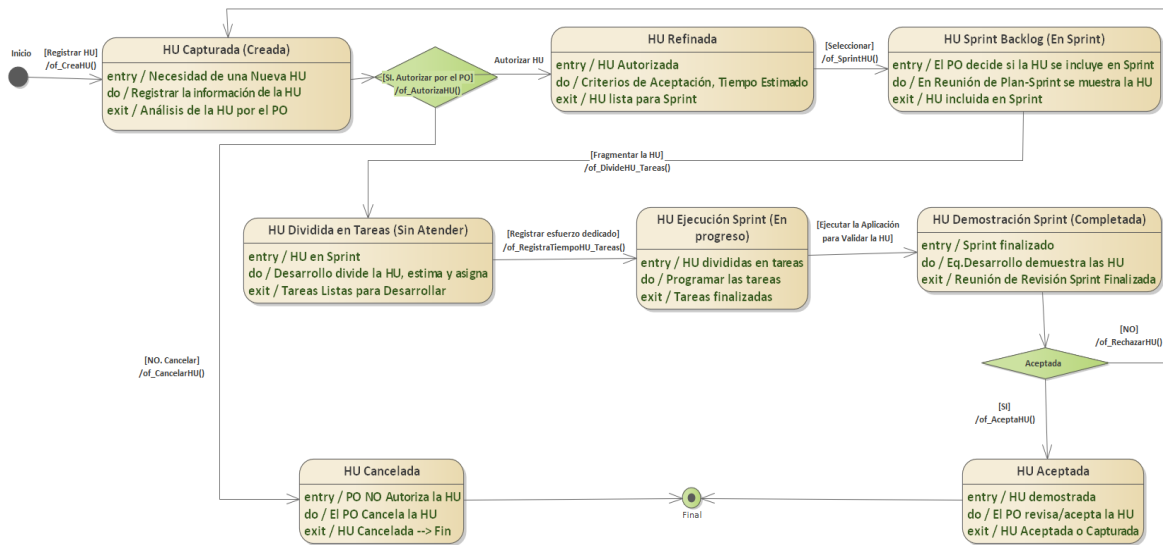


Figura 5.1. Diagrama de Máquina de Estados del flujo de un Historia de Usuario
Fuente: Elaboración propia

Pudiendo representarse por:

- $q_0 = HUCapturada$
- $q_1 = HUCancelada$
- $q_2 = HURefinada$
- $q_3 = HUSprint$
- $q_4 = HUTareas$
- $q_5 = HUProgreso$
- $q_6 = HUCompletada$
- $q_7 = HUAceptada$

Entonces $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

Definición: Sea Σ que representa el conjunto de símbolos de entrada (alfabeto), de la máquina de estado.

En este caso particular el alfabeto será representado por las acciones que hacen que la historia de usuario cambie de un estado a otro, estas serán ejecutadas por el equipo Scrum o Stakeholders.

$\Sigma = \{CancelarHU, AutorizarHU, SeleccionarSprintHU, DividirHU, RegistrarEsfuerzoHU, ValidarHU, AceptarHU, RechazarHU\}$

Si

- $a = \text{CancelarHU}$
- $b = \text{AutorizarHU}$
- $c = \text{SeleccionarSprintHU}$
- $d = \text{DividirHU}$
- $e = \text{RegistrarEsfuerzoHU}$
- $f = \text{ValidarHU}$
- $g = \text{AceptarHU}$
- $h = \text{RechazarHU}$

Entonces $\Sigma = \{a, b, c, d, e, f, g, h\}$

Definición: Sea δ una función de transición $\delta: Q \times \Sigma \rightarrow Q$ que describe cómo la máquina cambia de un estado a otro en respuesta a una entrada.

Ejemplo: $\delta(q, a) = q'$ indica que, si la máquina está en el estado q y recibe la entrada a , entonces la máquina transitará al estado q' .

Definición del Estado Inicial:

Especifica un estado inicial $q_0 \in Q$ desde el cual comienza la máquina de estados,

$q_0 = \text{HUCapturada}$

Definición de Estados de Aceptación:

Si la máquina de estados tiene estados de aceptación, $F \subseteq Q$ de estados de aceptación, $F = \{q_1, q_7\} = \{\text{HUCancelada}, \text{HUAceptada}\}$

La formulación matemática del diagrama de la máquina de estado de la Figura 5.1 estaría representado por:

El autómata finito determinista (AFD) A , el cual está definido por la quintupla

$A = (Q, \Sigma, \delta, q_0, F)$, donde

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$, conjunto de estados posibles de una HU

$\Sigma = \{a, b, c, d, e, f, g, h\}$, conjunto de símbolos de entrada, equivalentes a acciones o eventos que provocan cambios de estado.

δ = Función de transición que está dada por la

Tabla 5.1 y la Figura 5.2

q_0 = Estado inicial, correspondiente a la captura de una HU

$F = \{q_1, q_7\}$, que representan la cancelación definitiva o la aceptación formal de la HU.

Tabla 5.1 Tabla de transiciones de un AFD

	a	b	c	d	e	f	g	h
$\rightarrow q_0$	q_1	q_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$*q_1$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\emptyset$	q_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\emptyset$	q_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	q_5	$\emptyset$	$\emptyset$	$\emptyset$
q_5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	q_6	$\emptyset$	$\emptyset$
q_6	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	q_7	q_0
$*q_7$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

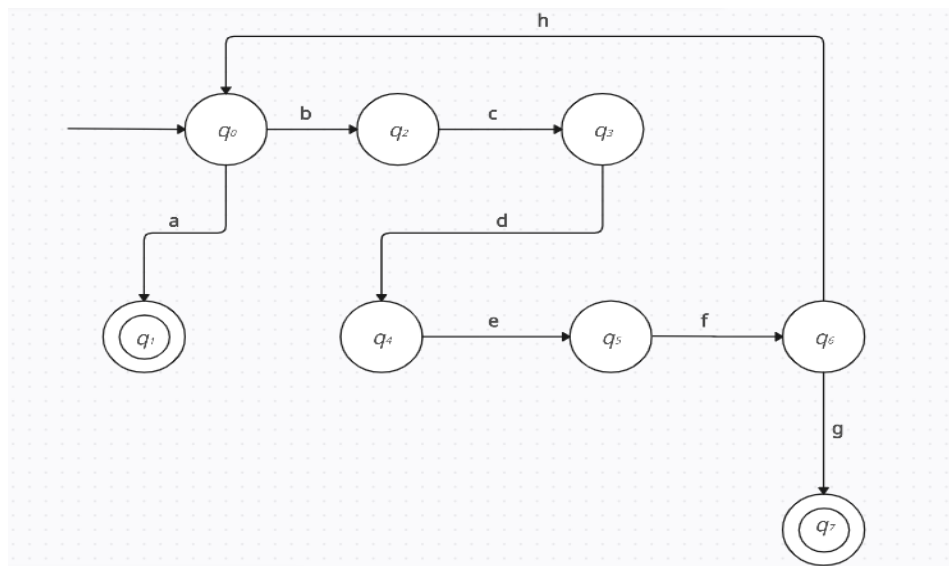


Figura 5.2. ADF que representa el flujo del ciclo de vida de una HU en Scrum
Fuente: Elaboración propia

En la Tabla 5.2 se presenta la correspondencia entre los estados del autómata y las etapas clave en el flujo de una HU en Scrum, así como las ceremonias asociadas.

Este AFD permite representar de manera formal y visual el flujo de trabajo de una historia de usuario en proyectos Scrum, facilitando la comprensión de su evolución a lo largo del ciclo de desarrollo y permitiendo, además, identificar puntos críticos, como rechazos, bucles

de refinamiento o cancelaciones.

Tabla 5.2 Correspondencia entre los estados y el flujo de una HU

Estado del AFD	Descripción	Etapas en Scrum	Evento relacionado
q0	HUCapturada	Incorporación al backlog	
q1	HUCancelada	Eliminación de HU no prioritaria	Refinamiento del backlog
q2	HURefinada	Detalle y evaluación técnica	Refinamiento del backlog
q3	HUSprint (seleccionada)	Planificación del Sprint	Sprint Planning
q4	HUTareas (dividida)	División en tareas técnicas	Fin del Sprint Planning
q5	HUProgreso (en desarrollo)	Implementación	Ejecución del Sprint
q6	HUCompletada (técnicamente)	Finalización técnica	Fin del Sprint
q7	HUAceptada	Validación por el PO	Sprint Review

Capítulo 6 Resultados

La evaluación multietapa del modelo de comunicación adaptado al marco Scrum permitió obtener evidencias sólidas sobre su validez, confiabilidad y efectividad en entornos reales de desarrollo de software. Los principales hallazgos se presentan a continuación.

Validez de contenido (Método Delphi)

La validación por expertos, realizada mediante el método *Delphi* con la participación de diez especialistas, alcanzó consenso en dos rondas consecutivas. Los valores finales del coeficiente *V de Aiken* fueron superiores a 0.80 en todos los *ítems*. Estos resultados confirman la pertinencia y claridad del instrumento de evaluación, consistente en una encuesta aplicada a los expertos para valorar el modelo de comunicación adaptado propuesto en esta investigación.

Confiabilidad en población profesional

La aplicación del instrumento a una muestra internacional de 44 profesionales con experiencia en Scrum arrojó un **Alfa de Cronbach de 0.862**, indicador de una alta consistencia interna y de la solidez del instrumento para evaluar el modelo en equipos distribuidos.

Validación empírica (Caso Grupo Funerario)

En el estudio aplicado a un entorno real, el instrumento mostró un **Alfa de Cronbach de 0.921**, valor que evidencia una excelente consistencia interna en condiciones operativas. La percepción del equipo reflejó resultados altamente favorables, con medias comprendidas entre **4.3 y 5.0**, y **22 de los 29 ítems** presentaron mediana y moda igual a 5. El *ítem* con mejor valoración fue **P21**, relacionado con la mejora de tareas a través de la retroalimentación, con una calificación de **5.0**, sin variabilidad (desviación estándar = 0.0), esto es un indicador fuerte de que la dimensión de retroalimentación es un aspecto del modelo muy pertinente. La varianza baja en los ítems indica que los participantes tienen una opinión muy similar, lo cual es positivo ya que indica que los constructos medidos son estables.

Cumplimiento de indicadores del proyecto

Los resultados cuantitativos del caso demostraron un desempeño destacado:

- **93.8 %** de historias de usuario completadas y aceptadas, superando el umbral esperado ($\geq 90 \%$).
- Únicamente una historia de usuario quedó incompleta debido a un factor externo (enfermedad de un desarrollador), sin retrabajos asociados ni problemas derivados de fallas de comunicación.
- Se generaron **32 retroalimentaciones**, todas atendidas en su totalidad, lo que refleja una **comprensión del 100 %** de los mensajes.
- Se observó una reducción significativa de malentendidos (P27: media 4.8) y una mejora notable en la comunicación del equipo (P25: media 4.8).
- El modelo mostró adecuación a entornos distribuidos (P29: media 4.6).

Cumplimiento de los objetivos específicos

- **La recepción efectiva de los mensajes** fue del 100 %, dado que la totalidad de los mensajes enviados fueron recibidos correctamente por los 5 integrantes del equipo (285 mensajes recibidos frente a 57 enviados), según los registros de la herramienta Comunica-Scrum. Este resultado se ve respaldado por las respuestas positivas obtenidas en las preguntas **P2, P8, P11, P12, P13, P14, P15, P17 y P19** del instrumento aplicado al equipo Scrum, las cuales evalúan aspectos relacionados con la transmisión, recepción y verificación de la información durante los *Sprints*.
- **Comprensión verificada:** el 100 % de los mensajes fueron confirmados y comprendidos por los destinatarios. Este resultado se sustenta en las respuestas favorables obtenidas en las preguntas **P1, P3, P4, P5, P6, P7, P9, P10, P16, P18, P20, P26 y P29** del instrumento aplicado al equipo Scrum, las cuales evalúan la claridad, precisión, adecuación y entendimiento de los mensajes transmitidos durante los *Sprints*.
- **Retroalimentación oportuna y efectiva:** el 100 % de los casos fue atendido y resuelto mediante la herramienta Comunica-Scrum. Este resultado se respalda con las respuestas favorables registradas en las preguntas **P17, P21, P22, P23, P24, P25, P27 y P28** del instrumento aplicado al equipo Scrum, las cuales evalúan la calidad, oportunidad, utilidad y efecto de la retroalimentación en el desarrollo de las tareas y en la coordinación del equipo.

Los resultados obtenidos en las tres fases, validación por expertos, aplicación a profesionales y caso de estudio, demuestran de manera consistente que el modelo de comunicación adaptado de Berlo al marco Scrum, implementado a través de la aplicación Comunica-Scrum, mejora la efectividad comunicativa en equipos distribuidos, reduce de forma notable los malentendidos, incrementa la eficiencia en el cumplimiento de historias de usuario y contribuye directamente al éxito del proyecto. En conjunto, los indicadores muestran que el modelo alcanza los objetivos planteados, consolidándose como una herramienta sólida y funcional para entornos ágiles distribuidos.

Capítulo 7 Conclusiones, Aportaciones y trabajos futuros

Conclusiones

La versión adaptada del modelo de comunicación de Berlo, cuando se pone en práctica dentro del marco Scrum, ha mostrado ser eficaz no solo en proyectos de desarrollo global de software (DGS), sino también en grupos que laboran presencialmente o colocalizados. Las altas puntuaciones promedio (medias superiores a 4) indican que los profesionales en desarrollo de software consideran que el modelo facilita una comunicación efectiva dentro de los equipos de desarrollo. Esto sugiere que los componentes del modelo de Berlo (emisor, mensaje, canal y receptor) están bien adaptados para abordar las necesidades de comunicación en un entorno de desarrollo de software bajo el marco de trabajo Scrum.

Los objetivos específicos se cumplieron de la siguiente forma:

1. Asegurar que los mensajes generados en cada una de las ceremonias de Scrum sean recibidos por los involucrados a través de herramientas de comunicación. Cumplido al 100 %. En el estudio de caso Grupo Funerario se registraron 285 recepciones correctas frente a 57 mensajes enviados. En la encuesta a 44 profesionales y en la validación por expertos se obtuvo una percepción promedio superior a 4.5/5 en ítems relacionados con la recepción efectiva y oportuna de la información.
2. Verificar que los mensajes recibidos sean comprendidos en su estructura y contenido utilizando herramientas de comunicación. Cumplido al 100 %. Inicialmente solo el 11.22 % de los mensajes requirió aclaración; tras el mecanismo de retroalimentación propuesto, la comprensión alcanzó el 100 %. Las medias de los ítems relacionados con claridad, precisión y comprensión del mensaje oscilaron entre 4.6 y 5.0 tanto en la encuesta internacional como en la aplicación real.
3. Retroalimentar la comprensión de los mensajes analizados por los participantes involucrados mediante herramientas de comunicación. Cumplido al 100 %. Se generaron 32 retroalimentaciones durante los tres Sprints del caso de estudio y todas fueron atendidas y resueltas. El ítem P21 ("La retroalimentación proporcionada mediante Comunica-Scrum contribuyó a mejorar la ejecución de las

tareas”) obtuvo la máxima puntuación posible (media = 5.0, desviación estándar = 0.0).

Los resultados obtenidos en las tres etapas de evaluación, (a) validación de contenido por expertos mediante Método Delphi (V de Aiken > 0.80 en todos los ítems tras segunda ronda), (b) aplicación del instrumento a 44 profesionales internacionales de siete países (Alfa de Cronbach = 0.862 y medias > 4.1 en todos los ítems), y (c) validación empírica en caso de estudio real con la aplicación Comunica-Scrum (Alfa de Cronbach = 0.921 y 22 de 29 ítems con mediana/moda = 5), demuestran de forma consistente y robusta que la adaptación del modelo SMCR de David Berlo al marco Scrum, complementado con la herramienta tecnológica Comunica-Scrum, constituye una solución efectiva, replicable y escalable para los problemas de comunicación en equipos Scrum distribuidos y colocalizados.

Se comprobó que la estructuración explícita de los elementos Emisor-Mensaje-Canal-Receptor, junto con el mecanismo obligatorio de retroalimentación bidireccional después de cada ceremonia Scrum, reduce significativamente los malentendidos, incrementa la trazabilidad de acuerdos y decisiones, y mejora la alineación del equipo con los objetivos del producto, incluso en condiciones de dispersión geográfica, diferencias horarias e idiomas distintos.

Aportaciones

Este trabajo ofrece evidencia sólida acerca de la utilidad y efectividad del modelo de comunicación de Berlo cuando se aplica dentro del marco Scrum, tanto en entornos de desarrollo global de software como en equipos que comparten ubicación física.

El objetivo principal de la investigación ha sido mejorar la comunicación y la colaboración entre los miembros del equipo Scrum y las partes interesadas, asegurando que todos mantengan un seguimiento claro y actualizado del avance del proyecto, lo que a su vez favorece el cumplimiento de los requisitos definidos desde el inicio.

La investigación destaca aspectos específicos del modelo de comunicación que fueron bien recibidos por los profesionales en el área de desarrollo de software.

La aplicación Comunica-Scrum representa una herramienta para implementar el modelo de comunicación propuesto en el marco de esta investigación. Esta plataforma está diseñada

específicamente para facilitar la gestión eficaz de la comunicación dentro de equipos que emplean el marco de trabajo Scrum. Comunica-Scrum facilita la coordinación fluida de mensajes clave durante las ceremonias y actividades del ciclo de desarrollo Scrum. Esta herramienta asegura la transmisión y comprensión efectiva de la información por parte de todos los miembros del equipo. Comunica-Scrum se basa en principios de agilidad y colaboración, lo que optimiza la interacción y la retroalimentación entre los participantes.

Trabajos futuros

Los resultados y el enfoque metodológico de esta investigación pueden servir como base para estudios futuros que busquen evaluar la efectividad de diferentes modelos de comunicación en otros contextos ágiles.

Se propone la conversión de mensajes de texto a voz y viceversa en las ceremonias Scrum.

Además, se sugiere incorporar un canal de comunicación en la aplicación Comunica-Scrum mediante videoconferencias, lo cual facilitará las reuniones del equipo de trabajo remoto, permitiendo también el uso compartido de pantalla y presentaciones.

La realización de estudios de caso con equipos de mayor tamaño y durante periodos de tiempo más prolongados permitirá obtener evidencia adicional sobre la escalabilidad y el comportamiento del modelo en contextos más complejos.

Referencias

- [1] N. K. Kamaruddin, N. H. Arshad, y A. Mohamed, “Chaos issues on communication in Agile Global Software Development”, en *2012 IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC)*, Kuala Lumpur, Malaysia: IEEE, abr. 2012, pp. 394–398. doi: 10.1109/BEIAC.2012.6226091.
- [2] J. Cho, “Issues and Challenges of Agile Software Development with Scrum”, *IACIS*, vol. 9, núm. 2, Art. núm. 2, 2008, doi: 10.48009/2_iis_2008_188-195.
- [3] K. Beck *et al.*, “Manifiesto for Agile Software Development”. Consultado: el 1 de diciembre de 2021. [En línea]. Disponible en: <http://agilemanifesto.org/iso/en/manifesto.html>
- [4] V. Holzmann y I. Panizel, “Communications management in scrum projects”, *7th European Conference on Information Management and Evaluation, ECIME 2013*, pp. 67–74, ene. 2013.
- [5] S. H. T. Editores, “Proyectos de TI afectados por mala comunicación: Praxis”, *InfoChannel*. diciembre de 2014. Consultado: el 3 de julio de 2021. [En línea]. Disponible en: <https://infochannel.info/proyectos-de-ti-afectados-por-mala-comunicacion-praxis/>
- [6] G. Gbegnedji, “Gestión de las Comunicaciones del Proyecto”, *Project Management* \textbar Gladys Gbegnedji. enero de 2016. Consultado: el 3 de julio de 2021. [En línea]. Disponible en: <https://www.gladysgbegnedji.com/gestion-de-las-comunicaciones-del-proyecto/>
- [7] R. S. Pressman y B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, Ninth. McGraw-Hill, 2019.
- [8] I. Jacobson, “A Resounding ‘Yes’ to Agile Processes—But Also More”, *Cutter IT Journal*, vol. 15, pp. 18–24, 2002.
- [9] I. Sommerville, *Software Engineering*, Tenth. Harlow, England: Pearson, 2015.
- [10] A. Vardy, *Scrum. The Ultimate Beginners Guide To Mastering Scrum To Boost Productivity & Beat Deadlines*, 1st ed. Adam Vardy, 2015.
- [11] R. S. Kenneth, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, 1st ed. Addison-Wesley, 2013.

-
- [12] H. Oktaba, “Desarrollo Global de Software”, SG Buzz. Consultado: el 16 de diciembre de 2021. [En línea]. Disponible en: <https://sg.com.mx/revista/30/desarrollo-global-software>
- [13] PMI, *A Guide to the Project Management Body of Knowledge (PMBOK)*, 7th Edition. Project Management Institute, 2021.
- [14] P. A. Mayab, “Comunicación efectiva: ¿Cómo aplicarla con tu equipo de trabajo?” Consultado: el 7 de junio de 2023. [En línea]. Disponible en: <https://merida.anahuac.mx/posgrado/blog/comunicacion-efectiva>
- [15] E. Aie y M. J. Harrison, *Comunicación Efectiva: 5 Consejos y Ejercicios Esenciales para Mejorar la Forma en que se Comunica en este Mundo Dividido, ¡Incluso si se Trata de Política, Raza o Género!* Independently Published, 2019.
- [16] Slack, “Team Collaboration Tool”, Slack. Consultado: el 22 de enero de 2026. [En línea]. Disponible en: <https://slack.com/team-collaboration-tool>
- [17] Microsoft Teams, “Overview of teams and channels in Microsoft Teams - Microsoft Teams”. Consultado: el 22 de enero de 2026. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/microsoftteams/teams-channels-overview>
- [18] Atlassian, “Jira \textbar Software de seguimiento de proyectos e incidencias”, *Atlassian*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>
- [19] Trello, “Trello”. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://trello.com>
- [20] J. C. Serrano y F. A. Zapico, “La fórmula de comunicación de Lasswell como método para implementar bases de datos documentales en los medios audiovisuales”, *Investigación Bibliotecológica: archivonomía, bibliotecología e información*, vol. 18, núm. 37, Art. núm. 37, jul. 2004, doi: 10.22201/iibi.0187358xp.2004.37.4048.
- [21] D. Baecker, “Teorías sistémicas de la comunicación”, *Revista Mad. Revista del Magíster en Análisis Sistémico Aplicado a la Sociedad*, núm. 37, pp. 1–20, 2017.
- [22] D. K. Berlo, *El proceso de la COMUNICACIÓN, Introducción a la teoría y la práctica*, 14°. EL ATENEO, 1984.
- [23] R. Jakobson, *Lingüística y Poética*. 1993.
- [24] W. Schramm, *El proceso y efectos de la comunicación de masas*. 1970.

-
- [25] R. Braddock, "An extension of the Lasswell Formula", *Journal of Communication*, 1958.
- [26] G. Maletzke, *Psicología de la Comunicación Social*, 5a. QUIPUS, 1992.
- [27] T. Sablatzky, "The Delphi Method", *Hypothesis: Research Journal for Health Information Professionals*, vol. 34, núm. 1, Art. núm. 1, ago. 2022, doi: 10.18060/26224.
- [28] M. Niederberger y J. Spranger, "Delphi Technique in Health Sciences: A Map", *Front Public Health*, vol. 8, p. 457, 2020, doi: 10.3389/fpubh.2020.00457.
- [29] J. C. Almenara y A. I. Moro, "Empleo del método Delphi y su empleo en la investigación en comunicación y educación", *Edutec, Revista Electrónica de Tecnología Educativa*, núm. 48, Art. núm. 48, jun. 2014, doi: 10.21556/edutec.2014.48.187.
- [30] L. R. Aiken, "Content Validity and Reliability of Single Items or Questionnaires", *Educational and Psychological Measurement*, vol. 40, núm. 4, pp. 955–959, dic. 1980, doi: 10.1177/001316448004000419.
- [31] M. Tavakol y R. Dennick, "Making sense of Cronbach's alpha", *Int J Med Educ*, vol. 2, pp. 53–55, jun. 2011, doi: 10.5116/ijme.4dfb.8dfd.
- [32] J. F. DeFranco y P. A. Laplante, "Review and Analysis of Software Development Team Communication Research", *IEEE Transactions on Professional Communication*, vol. 60, núm. 2, pp. 165–182, jun. 2017, doi: 10.1109/TPC.2017.2656626.
- [33] D. Kostin y D. Strode, "Effective Communication in Globally Distributed Scrum Teams", 2022.
- [34] Y. I. Alzoubi, A. Q. Gill, y A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review", *Information & Management*, vol. 53, núm. 1, pp. 22–37, ene. 2016, doi: 10.1016/j.im.2015.08.003.
- [35] M. Shameem, R. R. Kumar, M. Nadeem, y A. A. Khan, "Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process", *Applied Soft Computing*, vol. 90, p. 106122, may 2020, doi: 10.1016/j.asoc.2020.106122.

-
- [36] D. Šmite, N. B. Moe, y J. Gonzalez-Huerta, "Overcoming cultural barriers to being agile in distributed teams", *Information and Software Technology*, vol. 138, p. 106612, oct. 2021, doi: 10.1016/j.infsof.2021.106612.
- [37] M. Niazi *et al.*, "Challenges of project management in global software development: A client-vendor analysis", *Information and Software Technology*, vol. 80, pp. 1–19, dic. 2016, doi: 10.1016/j.infsof.2016.08.002.
- [38] S. Esquivel, "Communication Issues in Agile Software Development", 2016. Consultado: el 3 de julio de 2021. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/56630>
- [39] P. Walimbe, "To Overcome Communication Challenges in Distributed/Virtual Scrum Teams", *Dissertations and Theses*, dic. 2016, [En línea]. Disponible en: https://digitalcommons.harrisburgu.edu/pmgt_dandt/4
- [40] M. Korkala y P. Abrahamsson, "Communication in Distributed Agile Development: A Case Study", en *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)*, ago. 2007, pp. 203–210. doi: 10.1109/EUROMICRO.2007.23.
- [41] A. Yagüe, J. Garbajosa, J. Díaz, y E. González, "An exploratory study in communication in Agile Global Software Development", *Computer Standards & Interfaces*, vol. 48, pp. 184–197, nov. 2016, doi: 10.1016/j.csi.2016.06.002.
- [42] V. Stray, N. B. Moe, y M. Noroozi, "Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams", en *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, may 2019, pp. 111–121. doi: 10.1109/ICGSE.2019.00031.
- [43] V. Stray, J. H. Gundelsby, R. Ulfesnes, y N. Brede Moe, "How agile teams make Objectives and Key Results (OKRs) work", en *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, en ICSSP '22. New York, NY, USA: Association for Computing Machinery, may 2022, pp. 104–109. doi: 10.1145/3529320.3529332.
- [44] M. Paasivaara y C. Lassenius, "Scaling Scrum in a Large Globally Distributed Organization: A Case Study", en *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, ago. 2016, pp. 74–83. doi: 10.1109/ICGSE.2016.34.
- [45] J. Bablo, B. Marcinkowski, y A. Przybylek, "Overcoming Challenges of Virtual Scrum Teams: Lessons Learned Through an Action Research Study", en *Agile*

Processes in Software Engineering and Extreme Programming, C. J. Stettina, J. Garbajosa, y P. Kruchten, Eds., Cham: Springer Nature Switzerland, 2023, pp. 34–49. doi: 10.1007/978-3-031-33976-9_3.

[46] E. Bjarnason, B. Gislason Bern, y L. Svedberg, “Inter-team communication in large-scale co-located software engineering: a case study”, *Empir Software Eng*, vol. 27, núm. 2, p. 36, ene. 2022, doi: 10.1007/s10664-021-10027-z.

[47] E. Bjarnason, K. Smolander, E. Engström, y P. Runeson, “A theory of distances in software engineering”, *Information and Software Technology*, vol. 70, pp. 204–219, feb. 2016, doi: 10.1016/j.infsof.2015.05.004.

[48] P. Hohl *et al.*, “Back to the future: origins and directions of the ‘Agile Manifesto’ – views of the originators”, *Journal of Software Engineering Research and Development*, vol. 6, núm. 1, p. 15, nov. 2018, doi: 10.1186/s40411-018-0059-z.

[49] A. Cockburn, *Agile Software Development: The Cooperative Game*, 2a ed. Pearson Education, 2006.

[50] Y. Alzoubi, A. Gill, y A. Al-Ani, “Distributed Agile Development Communication: An Agile Architecture Driven Framework”, *Journal of Software*, vol. 10, pp. 681–694, jun. 2015, doi: 10.17706/jsw.10.6.681-694.

[51] A. A. Khan, S. Basri, y P. D. D. Dominc, “A Proposed Framework for Communication Risks During RCM in GSD”, *Procedia - Social and Behavioral Sciences*, vol. 129, pp. 496–503, may 2014, doi: 10.1016/j.sbspro.2014.03.706.

[52] E. S. Hidalgo, “Adapting the scrum framework for agile project management in science: case study of a distributed research initiative”, *Heliyon*, vol. 5, núm. 3, Art. núm. 3, mar. 2019, doi: 10.1016/j.heliyon.2019.e01447.

[53] H. Noor *et al.*, “Identifying Communication Issues Contributing to the Formation of Chaotic Situation: An AGSD View”, *International Journal of Advanced Computer Science and Applications*, vol. 12, ene. 2021, doi: 10.14569/IJACSA.2021.0120268.

[54] T. El-Najar, I. Ahmad, y M. Alkandari, “Easycomm - A Framework and Tool to Solve Client Communication Problem in Agile Development”, *IAENG International Journal of Computer Science*, vol. 46, feb. 2019.

[55] R. D. S. Santos, P. Ralph, A. Arshad, y K.-J. Stol, “Distributed Scrum: A Case Meta-analysis”, *ACM Comput. Surv.*, vol. 56, núm. 4, p. 100:1-100:37, nov. 2023, doi: 10.1145/3626519.

-
- [56] J. Noll, S. Beecham, y I. Richardson, "Global software development and collaboration: barriers and solutions", *ACM Inroads*, vol. 1, núm. 3, pp. 66–78, sep. 2011, doi: 10.1145/1835428.1835445.
- [57] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, y B. Fitzgerald, "Global software development: where are the benefits?", *Commun. ACM*, vol. 52, núm. 8, pp. 127–131, ago. 2009, doi: 10.1145/1536616.1536648.
- [58] S. ul Haq, M. Raza, A. Zia, y M. N. A. Khan, "Issues in Global Software Development: A Critical Review", *Journal of Software Engineering and Applications*, vol. 4, núm. 10, Art. núm. 10, oct. 2011, doi: 10.4236/jsea.2011.410069.
- [59] G. Lampropoulos y K. Siakas, *Communication in Distributed Agile Software Development: Impact of Social Media – Social Networking*. 2018.
- [60] P. Ciancarini, S. Ergasheva, I. Gizzatullina, V. Ivanov, S. Masyagin, y G. Succi, "Empirical Research on Customer Communication Challenges in the Companies Adopting Agile Practices", dic. 2021, pp. 139–150. Consultado: el 1 de diciembre de 2021. [En línea]. Disponible en: <https://www.scitepress.org/Link.aspx?doi=10.5220/0010526401390150>
- [61] K. Schwaber y J. Sutherland, "La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego". 2020.
- [62] G. Ammad, U. Iqbal Janjua, T. M. Madni, M. F. Cheema, y A. R. Shahid, "An Empirical Study to Investigate the Impact of Communication Issues in GSD in Pakistan's IT Industry", *IEEE Access*, vol. 7, pp. 171648–171672, 2019, doi: 10.1109/ACCESS.2019.2953008.
- [63] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, y A. Ebdrup, "From Scrum to Agile: a journey to tackle the challenges of distributed development in an Agile team", en *Proceedings of the 2018 International Conference on Software and System Process*, en ICSSP '18. New York, NY, USA: Association for Computing Machinery, may 2018, pp. 11–20. doi: 10.1145/3202710.3203149.
- [64] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, y A. Ebdrup, "From Scrum to Agile: a journey to tackle the challenges of distributed development in an Agile team", en *Proceedings of the 2018 International Conference on Software and System Process*, en ICSSP '18. New York, NY, USA: Association for Computing Machinery, may 2018, pp. 11–20. doi: 10.1145/3202710.3203149.

-
- [65] M. Wyrich, I. Bogicevic, y S. Wagner, “Improving Communication in Scrum Teams”, en *Product-Focused Software Process Improvement*, M. Felderer, D. Méndez Fernández, B. Turhan, M. Kalinowski, F. Sarro, y D. Winkler, Eds., en *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2017, pp. 602–605. doi: 10.1007/978-3-319-69926-4_50.
- [66] P. Srivastava y S. Jain, “A leadership framework for distributed self-organized scrum teams”, *Team Performance Management: An International Journal*, vol. 23, núm. 5/6, pp. 293–314, ene. 2017, doi: 10.1108/TPM-06-2016-0033.
- [67] F. R. Maulana y T. Raharjo, “Identification of Challenges, Critical Success Factors, and Best Practices of Scrum Implementation: An Indonesia Telecommunication Company Case Study”, *J. Phys.: Conf. Ser.*, vol. 1811, núm. 1, p. 012120, mar. 2021, doi: 10.1088/1742-6596/1811/1/012120.
- [68] R. A. Khan *et al.*, “An Evaluation Framework for Communication and Coordination Processes in Offshore Software Development Outsourcing Relationship: Using Fuzzy Methods”, *IEEE Access*, vol. 7, pp. 112879–112906, 2019, doi: 10.1109/ACCESS.2019.2924404.
- [69] H. Amar, P. M. Rafi-Ul-Shan, y A. Adegbile, *Towards A 5c Theory of Communication for Scrum-Based Distributed Projects*. 2019.
- [70] R. Jain y U. Suman, “A Project Management Framework for Global Software Development”, *SIGSOFT Softw. Eng. Notes*, vol. 43, núm. 1, Art. núm. 1, mar. 2018, doi: 10.1145/3178315.3178329.
- [71] R. S. Pressman y B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, Ninth. McGraw-Hill, 2019.
- [72] M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal, y N. Cerpa, “Toward successful project management in global software development”, *International Journal of Project Management*, vol. 34, núm. 8, pp. 1553–1567, nov. 2016, doi: 10.1016/j.ijproman.2016.08.008.
- [73] J. Sutherland, G. Schoonheim, y M. Rijk, “Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams”, feb. 2009, pp. 1–8. doi: 10.1109/HICSS.2009.225.
- [74] I. H. de Farias Junior, H. P. de Moura, y S. Marczak, “Towards a Communication Maturity Model for Distributed Software Development”, en *2013 IEEE*

8th International Conference on Global Software Engineering Workshops, ago. 2013, pp. 81–83. doi: 10.1109/ICGSEW.2013.18.

[75] Ivaldir de Farias Junior, S. Marczak, R. Santos, y H. Moura, “Communication in Distributed Software Development: A Preliminary Maturity Model”, en *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, ago. 2016, pp. 164–173. doi: 10.1109/ICGSE.2016.31.

[76] M. W. Bhatti y A. Ahsan, “Effective Communication among Globally Distributed Software Development Teams: Development of an ‘Effective Communication’ Scale”, *Journal of Global Information Management (JGIM)*, vol. 25, núm. 3, pp. 40–62, 2017, doi: 10.4018/JGIM.2017070103.

[77] N. G. de Sá Leitão Júnior, I. H. de Farias Junior, S. Marczak, R. Santos, F. Furtado, y H. P. de Moura, “Evaluation of a Preliminary Assessment Method for Identifying the Maturity of Communication in Distributed Software Development”, en *Proceedings of the 2nd Workshop on Social, Human, and Economic Aspects of Software*, en WASHES '17. New York, NY, USA: Association for Computing Machinery, may 2017, pp. 12–18. doi: 10.1145/3098322.3098326.

[78] A. Abrar, “Distributed Stakeholder’s Communication Challenges in Distributed Agile Software Development - A systematic Literature Review”, feb. 2019.

[79] A. L’Erario, J. A. Gonçalves, J. A. Fabri, T. Pagotto, y R. H. Cunha Palácios, “CFDSD: a Communication Framework for Distributed Software Development”, *Journal of the Brazilian Computer Society*, vol. 26, núm. 1, p. 7, ago. 2020, doi: 10.1186/s13173-020-00101-7.

[80] S. Qureshi, S. U. R. Khan, Inayat-Ur-Rehman, Y. Javed, S. Saleem, y A. Iqbal, “A Conceptual Model to Address the Communication and Coordination Challenges During Requirements Change Management in Global Software Development”, *IEEE Access*, vol. 9, pp. 102290–102303, 2021, doi: 10.1109/ACCESS.2021.3091603.

[81] Y. Alzoubi y A. Gill, “The Critical Communication Challenges Between Geographically Distributed Agile Development Teams: Empirical Findings”, *IEEE Transactions on Professional Communication*, vol. 64, núm. 4, pp. 322–337, dic. 2021, doi: 10.1109/TPC.2021.3110396.

[82] K. Bundhun y R. K. Sungkur, “Developing a framework to overcome communication challenges in agile distributed teams – Case study of a Mauritian-

-
- based IT service delivery centre”, *Global Transitions Proceedings*, vol. 2, núm. 2, pp. 315–322, nov. 2021, doi: 10.1016/j.gltp.2021.08.006.
- [83] A. K. Koczewski Witoslaw, “Kunagi - Agile project management based on Scrum”, *Kunagi*. 2021. Consultado: el 6 de julio de 2021. [En línea]. Disponible en: <http://kunagi.org>
- [84] Atlassian, “Jira \textbar Software de seguimiento de proyectos e incidencias”, *Atlassian*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>
- [85] iceScrum, “iceScrum”, *iceScrum*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://www.icescrum.com/>
- [86] Sprintometer, “Sprintometer - Scrum & XP project tracking \textbar Website of Scrum tool Sprintometer”. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://sprintometer.com/>
- [87] ScrumDO, “ScrumDo - Software Scrum y Kanban en línea”. 2021. Consultado: el 6 de julio de 2021. [En línea]. Disponible en: <https://www.scrumdo.com/>
- [88] Trello, “Trello”. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://trello.com>
- [89] Asana, “Gestiona en línea el trabajo, los proyectos y las tareas de tu equipo • Asana”, *Asana*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://asana.com/es>
- [90] Slack, “Slack”, *Slack*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://slack.com/intl/es-la/>
- [91] Discord, “Discord \textbar Tu sitio para hablar y pasar el rato”, *Discord*. 2021. Consultado: el 7 de julio de 2021. [En línea]. Disponible en: <https://discord.com:2087/>
- [92] P. Runeson y M. Höst, “Guidelines for conducting and reporting case study research in software engineering”, *Empir Software Eng*, vol. 14, núm. 2, pp. 131–164, abr. 2009, doi: 10.1007/s10664-008-9102-8.
- [93] D. Harel, “Statecharts: a visual formalism for complex systems”, *Science of Computer Programming*, vol. 8, núm. 3, pp. 231–274, jun. 1987, doi: 10.1016/0167-6423(87)90035-9.

-
- [94] D. Drusinsky, *Modeling and Verification Using UML Statecharts A Working Guide to Reactive System Design, Runtime Monitoring and Execution-Based Model Checking*. Elsevier, 2006.
- [95] J. E. Hopcroft, R. Motwani, y J. D. Ullman, *Introducción a la teoría de autómatas lenguajes y computación*, Tercera. Madrid, España: Pearson, Addison Wesley, 2007.
- [96] “Asana para métodos Agile y Scrum”, Asana Help Center. Consultado: el 12 de diciembre de 2023. [En línea]. Disponible en: <https://help.asana.com/hc/es/articles/17964916348187-Asana-para-m%C3%A9todos-Agile-y-Scrum>

Capítulo 8 Anexos

Anexo A Divulgación del Proyecto de Investigación

8.1 Presentación del Artículo en un Congreso Internacional

Se elaboró y se presentó el artículo “Propuesta para Adaptar de un modelo de comunicación estratégico aplicable al marco de trabajo Scrum en un entorno DGS”, en el Congreso CIMPS 2022, 11th International Conference on Software Process Improvement, en la Ciudad de Acapulco, Gro., celebrado del 19 al 21 de octubre del 2022.

En la Figura 8.1 se muestra la página de la portada de las memorias del congreso CIMP 2022.



Figura 8.1. Página de la Portada Memorias del Congreso

En la Figura 8.2 y la Figura 8.3 se muestra el índice de las memorias del congreso.

Content

MODELS AND STANDARDS

1. **Mapeo Sistemático de Literatura: Prácticas de Gobierno y Gestión de TI en las organizaciones, bajo el marco de referencia COBIT 2019.**
Systematic Mapping of Literature: Governance and IT Management Practices in organizations, under the COBIT 2019 reference framework.
María Natalia Bohórquez Lobo, Torcoroma Velásquez Pérez, Dewar Rico-Bautista.....1
2. **Aplicación de las primeras fases de la ingeniería de software para el diseño de un sistema basado en una interfaz cerebro-computadora.**
Applying the first phases of software engineering for designing a brain-computer interface-based system.
Sonia López, Francisco López, José-Antonio Cervantes, Adriana Mexicano, Jesús Carlos Carmona.....10
3. **Diseño de una aplicación móvil para evaluar el estrés laboral implementando los cuestionarios MBI Y OIT-OMS empleando la norma NOM-035-STPS-2018.**
Design of a mobile application to assess work stress by implementing the MBI and OIT-OMS questionnaires using the NOM-035-STPS-2018 standard.
Roberto Lidio Garcés Tello, Luis Ángel Reyes Hernández, Celia Romero Torres, Ignacio López Martínez, Asdrúbal López Chau.....19
4. **Implementación de Azure DevOps para automatizar los procesos del estándar ISO/IEC 29110 un Caso de Estudio.**
Implementation of Azure DevOps to automate the processes of the ISO/IEC 29110 standard a Case Study.
Josefina García, J Jesús Minero, Elvia Lara29
5. **MyFEPS - Evaluación de Productos Software: Un análisis reflexivo del pasado y futuro.**
Software Product Evaluation - MyFEPS: a past and future reflective analysis.
Abraham Dávila, Paula Angeleri.....37
6. **Desafíos en la enseñanza de estándares internacionales en instituciones de educación superior.**
Challenges toward teaching international standards in higher education institutions.
Mirna Munoz.....46
7. **Metodología Scrumban-Ex Aplicado al Desarrollo de Herramientas de Visión Artificial para la Generación de Repositorios: Caso Poses Humanas.**
Scrumban-Ex Methodology Applied to the Development of Artificial Vision Tools for Repository Generation: Human Pose Case
Alberto Nolasco, Salvador Cervantes, José-Antonio Cervantes Adriana Mexicano.....52

8. Diseño de una API basada en el estándar HL7 para la interoperabilidad de sistemas de información médicos.

API Design based on HL7 standard to interoperability of medical information systems

I. Aguilar Juárez, L. Dávila Nicanor, J. Ayala de la Vega, J.E. Martínez Hernández.....60

9. Clasificación automática de audio obtenido de la percusión de frutas: hacia la estimación no destructiva de calidad.

Automatic classification of audio through fruit percussion: toward no destructive estimation of quality

Francisco Javier Becerra Sánchez, Humberto Pérez Espinosa, Marco Antonio Meza Aguilar, Maria Guadalupe Sánchez Acevedo.....69

10. Propuesta para adaptar un modelo de comunicación estratégico al marco de trabajo Scrum en entornos DGS.

Proposal for an adaptation of a Strategic communication model applicable to the Scrum framework in a GSD environment

Juan Miguel Hernández Bravo, Blanca Dina Valenzuela Robles, René Santaolaya Salgado, Juan Carlos Rojas Pérez, Olivia Graciela Fragoso Díaz, Noé Alejandro Castro Sánchez and María Clara Gómez Álvarez.....74

KNOWLEDGE MANAGEMENT

11. Modelado de Bases de Datos para la Nube: Una Revisión Sistemática de Literatura.

Database Modeling for the Cloud: A Systematic Review of the Literature

Identifying Activities for Enhancing Software Quality in DevOps Settings

Naomi García, Raúl A. Aguilar, Julio C. Díaz, Brenda L. Flores.....85

12. Modelo para la clasificación de señales de tráfico mexicanas dañadas y vandalizadas.

Model for the classification of damaged and vandalized Mexican traffic signs

José Antonio Rivas Navarrete, Humberto Pérez-Espinosa, Carlos Alberto Aguilar-Lazcano, Ansel Y. Rodríguez and Arley Magnolia Aquino Garcia.....95

13. Estrategia de un lago de datos para flujos de trabajo en ciencia de datos.

Data Lake Strategy for Data Science Workflows

Edgar Oswaldo Díaz.....101

14. Smart Contract Application on Blockchain Technology in the Software Industry

Wilson Soto.....106

En la Figura 8.4 se muestra la página inicial del artículo.

Propuesta para adaptar un modelo de comunicación estratégico al marco de trabajo Scrum en entornos DGS

Proposal for an adaptation of a Strategic communication model applicable to the Scrum framework in a GSD environment

Juan Miguel Hernández-Bravo, Blanca Dina Valenzuela-Robles,
René Santaolaya-Salgado, Juan Carlos Rojas-Pérez, Olivia
Graciela Frigoso-Díaz, Noé Alejandro Castro-Sánchez
Centro Nacional de Investigación y Desarrollo Tecnológico
Tecnológico Nacional de México (TecNM)/CENIDET
Cuernavaca, Mor., México
{d21ce091; blanca.vr; rene.ss; juan.rp; olivia.fd;
noe.cs}@cenidet.tecnm.mx

María Clara Gómez-Álvarez
Universidad Nacional de Colombia
Medellín, Colombia
mcgomez@udecmedellin.edu.co

Resumen — La comunicación efectiva es un factor de éxito al desarrollar proyectos de software en un entorno tanto local como distribuido geográficamente (Desarrollo Global de Software, DGS). Los problemas de comunicación que se presentan entre el equipo de desarrollo y el *product owner*, así como en los eventos *sprint planning*, *daily scrum*, *sprint review* y *sprint retrospective* trae como consecuencia problemas de ambigüedad en la determinación de requisitos. La poca participación del cliente, baja definición de la incertidumbre o identificación de riesgos afectan de manera directa el alcance, el presupuesto y el tiempo de entrega de un producto de software. Asimismo, la distancia, la diferencias entre zonas horarias y la cultura son algunos desafíos actuales que presentan problemas de comunicación entre los miembros del equipo de trabajo, lo cual tiene una gran repercusión en la gestión de proyectos. En este trabajo se describe brevemente una propuesta para adaptar un modelo de comunicación estratégico que se adapte al marco de trabajo Scrum y que sea adecuado tanto para entornos locales como distribuidos.

Palabras Clave — Scrum, DGS, comunicación.

Abstract — Effective communication is a success factor when developing software projects in both a local and geographically distributed environment (Global Software Development, GSD). The communication problems that arise between the development team and the product owner, as well as in the sprint planning, daily scrum, sprint review and sprint retrospective events, result in problems of ambiguity in determining requirements. Low customer involvement, low uncertainty definition, or risk identification directly affect the scope, budget, and delivery time of a software product. Likewise, distance, differences between time zones and culture are some current challenges that present communication problems between the members of the work team, which have a great impact on project management. This paper briefly describes a proposal to adapt a strategic communication

model that fits the Scrum framework and is suitable for both local and distributed environments.

Keywords — Scrum, GSD, communication

I. INTRODUCCIÓN

Los acontecimientos imprevistos de la actual crisis pandémica del COVID, han traído como consecuencia cambios culturales y organizacionales en todo el mundo. La comunicación entre los miembros de los equipos se ve afectada por diversos factores y uno de ellos es la dispersión geográfica de los equipos de desarrollo. Esto ha hecho que la forma de trabajar y la forma de comunicarse sea diferente, y con ello el crecimiento acelerado de nuevas aplicaciones que dan soporte a estas necesidades. El uso de la tecnología y la conectividad han hecho posible mitigar la propagación del virus y la actualización de aplicaciones no se ha hecho esperar.

Si bien es cierto que con los avances tecnológicos es posible reducir la brecha comunicacional, aún se presentan barreras en el proceso comunicativo entre equipos de desarrollo del enfoque ágil Scrum [1].

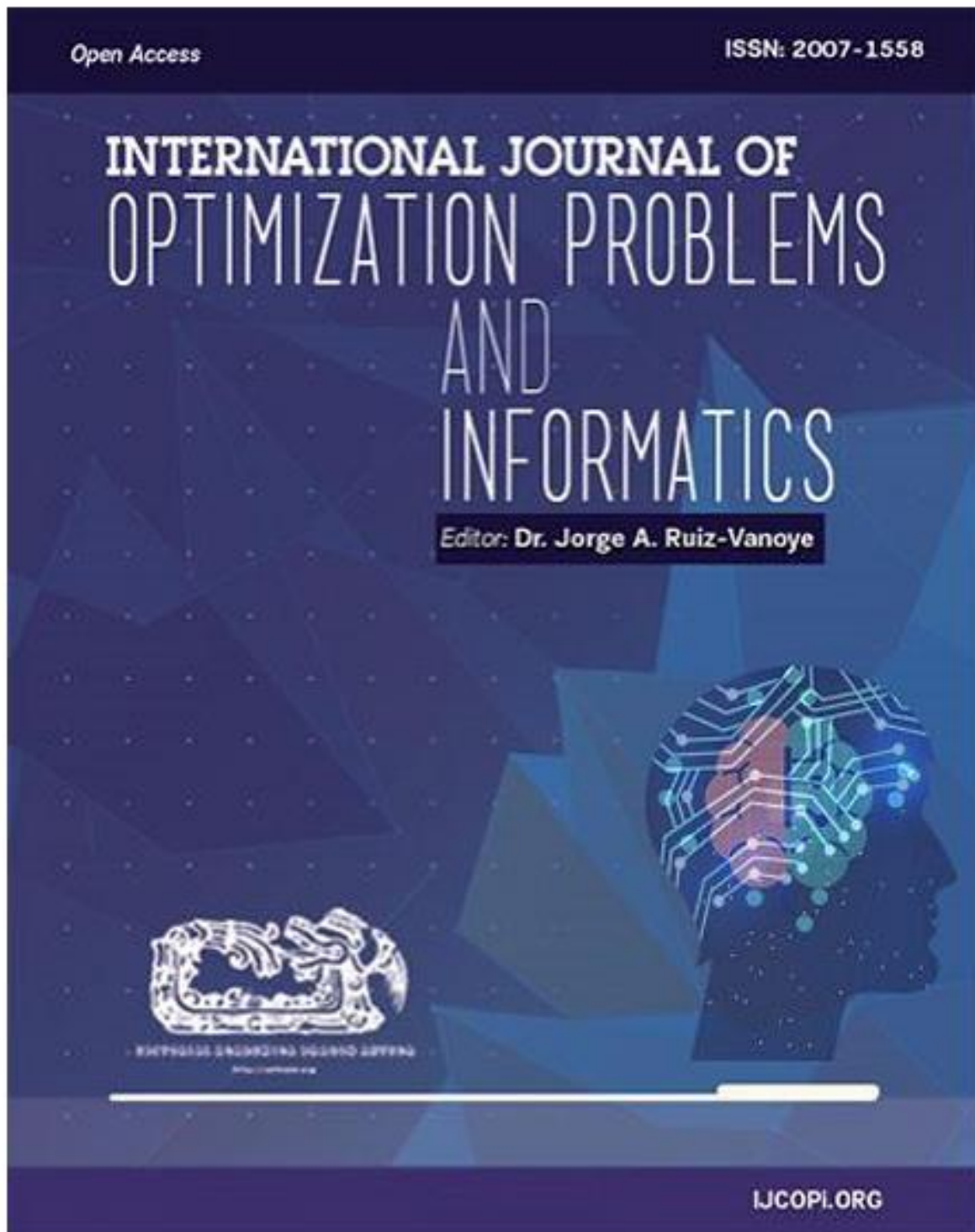
La falla en la comunicación en un entorno de Desarrollo Global del Software (DGS) incide directamente en la productividad del personal, originando ambientes negativos en la empresa y por lo tanto origina en gran parte problemas en el proceso de desarrollo de software.

La comunicación efectiva es un elemento fundamental en el DGS, específicamente en entornos ágiles distribuidos, por lo que la aplicación de un modelo de comunicación adecuada en dichos entornos permitirá en gran medida el éxito del proyecto.

El objetivo de este artículo es describir brevemente una propuesta para adaptar un modelo de comunicación estratégico que conviva con el marco de trabajo Scrum, adecuado tanto

8.2 Artículo JCR (Emerging Sources Citation Index) y Revista CONACYT

Se publicó en octubre de 2025 el artículo en inglés en la revista *International Journal of Optimization Problems and Informatics (IJCOP)*, con el título “*Application of Berlo's communication model to agile software development with Scrum in distributed teams*”, En la Figura 8.5 se muestra la página de la revista IJCOP.



Published: 2025-10-12

En la Figura 8.6 se muestra el índice de la revista IJCOPI Vol. 16 No. 4 (2025).

• Application of Berlo's Communication Model to Agile Software Development with Scrum in Distributed Teams
Juan Miguel Hernández Bravo, Blanca Dina Valenzuela Robles, René Santaolaya Salgado, Juan Gabriel González Serna, Noé Alejandro Castro Sánchez, María Clara Gómez Álvarez, Iliana Lizbeth Alvarado Lara
176-193
◦ PDF
• An Experimental Study on Grouping Mutation Operators within the GGA-CGT Applied to the One-Dimensional Bin Packing Problem
Alejandro Barojas-Vázquez, Marcela Quiroz-Castellanos, Guadalupe Carmona-Arroyo
194-211
◦ PDF
• Adaptive Traffic Congestion Prediction Using LSTM Networks and Physics-Based Models
Ernesto de la Cruz-Nicolás, Hugo Estrada-Esquivel, Alicia Martínez-Rebollar, Odette Pliego-Martínez, Eddie Clemente Torres
212-232
◦ PDF
• Using the Zipf Distribution to Mitigate the Matthew Effect and Improve Fairness in Bias-SVD Algorithm Recommendations
Cupertino Lucero Alvarez, Perfecto Malaquias Quintero Flores, David Eduardo Pinto Avendaño, Brian Manuel Gonzáles Contreras, María del Rocio Ochoa Montiel, Ever Juarez Guerra
233-249
◦ PDF
• Classification of Soil Type in Buildings Using Pseudo Spectral Acceleration Readings from Earthquake Events
Erick Rosete Beas, Laura Margarita Rodríguez Peralta, Eduardo Ismael Hernandez
250-258
◦ PDF
• Nonlinear Systems Modeling Using Polynomial Neural Networks: Comparison
Guillermo Ronquillo Lomeli

En la Figura 8.7 se muestra la página inicial del artículo de la revista IJCOP.



International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 176-193. ISSN: 2007-1558.
<https://doi.org/10.61467/2007.1558.2025.v16i4.1001>
www.editada.org

Application of Berlo's Communication Model to Agile Software Development with Scrum in Distributed Teams

Juan Miguel Hernández-Bravo¹, Blanca Dina Valenzuela-Robles¹, René Santaolaya-Salgado¹, Gabriel González-Serna¹, Noé Alejandro Castro-Sánchez¹, María Clara Gómez-Álvarez², Iliana Lizbeth Alvarado-Lara¹

¹ TecNM Centro Nacional de Investigación y Desarrollo Tecnológico, México
² Universidad Nacional de Colombia, Colombia
{d21ce091, blanca.vr, rene.ss, gabriel.gs, noe.cs, d23ce054}@cenidet.tecnm.mx, mcgomez@unal.edu.co

Abstract. This study examines communication challenges in distributed teams operating within the Scrum framework. To enhance communication effectiveness, Berlo's model was adapted. An instrument was developed and validated by experts through surveys and interviews with professionals experienced in agile project management. The instrument was subsequently applied to 44 practitioners across seven countries, assessing its capacity to improve communication and optimise the flow of information between teams and stakeholders. The findings confirmed that the adaptation of Berlo's model is effective in addressing communication challenges in distributed teams. In conclusion, the implementation of Berlo's model in Scrum teams reinforces clarity and cohesion in communication, thereby contributing to the success of agile projects in global environments.

Keywords: Communication challenges, Berlo, communication model, Scrum, global software development.

Article Info
Received March 13, 2025
Accepted July 8, 2025

1 Introduction

Globalization and technological advancement allow software development teams to be distributed in various geographical locations, which increases the need for effective communication models to coordinate their work. The agile framework Scrum emerges as one of the most popular solutions for managing software projects due to its flexibility and iterative approach (Beck et al., 2001). However, the application of Scrum in distributed teams faces several challenges, particularly with regard to communication, a fundamental aspect to ensure the successful delivery of quality software products (Hossain et al., 2011; Luz et al., 2009). Global Software Development (GSD) is a growing trend in software engineering due to its multiple benefits, such as cost reduction, access to skilled labor and the possibility of continuous work. Many organizations combine GSD with agile methodologies, thus achieving faster development and better project control. This combination makes it possible to optimize resources, guarantee higher quality products and take advantage of the benefits of having teams distributed in different locations (Ammad et al., 2019; Bannerman et al., 2012; Bundhun & Sungkur, 2021; Shafiq et al., 2019).

Although several studies highlight the importance of communication in distributed teams, few address the adaptation of theoretical communication models to the specific context of distributed Scrum. Berlo's SMCR model, which emphasizes the sender, message, channel, and receiver, provides a robust framework for improving communication in distributed Scrum teams. The message component includes code, elements, structure, content, and processing, influenced by the participants' communication skills, attitudes, knowledge, and sociocultural context. However, empirical validation of this model in distributed Scrum environments remains limited, as noted by Ammad et al. (2019), who highlight the need for further research to adapt and validate communication models in such contexts.

Adapting Berlo's communication model to a Scrum environment distributes messages effectively, improving communication and contribution to the fulfillment of requirements in projects.

© Editorial Académica Dragón Azteca S. de R.L. de C.V. (EDITADA.ORG). All rights reserved.

Figura 8.7. Página Inicial del Artículo IJCOP

8.3 Ponencia en el Instituto Tecnológico de Zacatepec

Se participó con una conferencia en Instituto Tecnológico de Zacatepec con el título “Mejorando la Comunicación e Scrum en un Ambiente Distribuido” en el Simposio de la Red Multidisciplinaria de Investigación y Tecnología PRODEP, dentro de las Jornadas Académicas de Innovación, Tecnología, Liderazgo y Sostenibilidad 2023, ver Figura 8.8.



Figura 8.8. Ponencia en el IT Zacatepec

Anexo B Adaptación del Modelo de Comunicación de los Eventos Scrum

El evento de la Reunión de la Refinamiento del Product Backlog se muestra en diagrama de la Figura 8.9, y en Figura 8.12 se muestra un ejemplo de un mensaje generado de esta ceremonia.

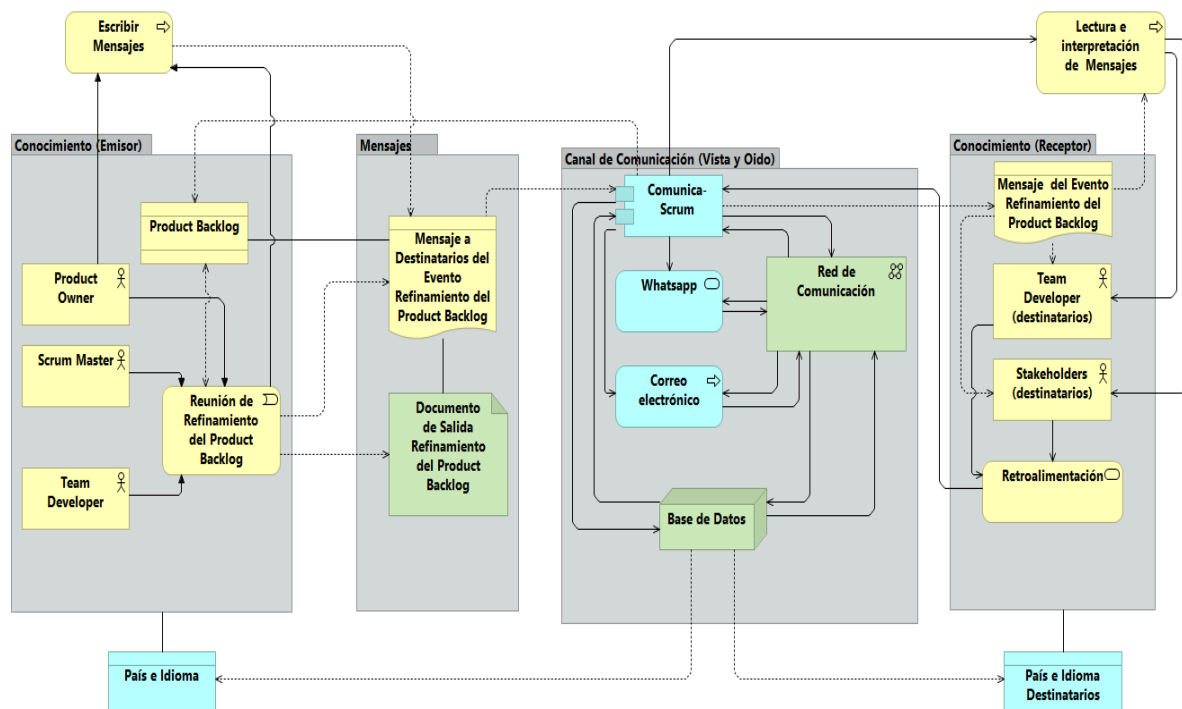


Figura 8.9. Adaptación del modelo comunicativo (Reunión de Refinamiento del Product Backlog)

En la Tabla 8.1 se describen los elementos del Mensaje de la Reunión de Refinamiento del *Product Backlog* del *Sprint* y en la Figura 8.11 se muestra un ejemplo del Mensaje de la Retroalimentación de la Reunión de Refinamiento del *Product Backlog*, y en la Tabla 8.2 se muestra la descripción de los elementos que componen el mensaje de la Retroalimentación de la Reunión de Refinamiento del *Product Backlog*.

Reporte de Reunión de Refinamiento del Product Backlog				
Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Fecha y hora	29 de Septiembre de 2025 a las 09:00			
Reunión de refinamiento del Product Backlog				
Idioma	Español			
País	Mexico			
Asistentes:	Nombre	Rol	Status	Tipo de Asistencia
	Jmiguél	Product Owner	Obligatorio	Síncrona
	Lorenzo	Developers	Obligatorio	Síncrona
	Brayan	Developers	Obligatorio	Síncrona
	Samuel	Developers	Obligatorio	Síncrona
	Jovany	Scrum Master	Obligatorio	Síncrona
	Yesenia	Product Owner	Obligatorio	Síncrona
HU	Nombre			
HU012	Cálculo de tiempos estimados de servicio			
	Tiempo estimado:	75 hrs		
	Priondad:	Alta		
	Descripción:	Como sistema, quiero calcular tiempos estimados de traslado y servicio para informar al cliente y coordinar al equipo.		
	Criterio de aceptación:	El sistema muestra todos los mensajes pendientes de aprobación. Se puede aprobar, editar o rechazar un mensaje. Se notifica al remitente si su mensaje es rechazado.		
HU013	Sistema de alertas internas e incidencias			
	Tiempo estimado:	75 hrs		
	Priondad:	Alta		
	Descripción:	Como sistema, quiero enviar alertas internas cuando se detecten retrasos o incidencias para tomar acciones correctivas.		
	Criterio de aceptación:	Cada venta se almacena con detalle de productos, extras y precio. Reporte exportable en Excel o PDF. Control de inventario se actualiza automáticamente.		
HU014	Generación de reportes de desempeño			
	Tiempo estimado:	60 hrs		
	Priondad:	Alta		
	Descripción:	Como supervisor, quiero generar reportes de desempeño por tipo de servicio y zona para optimizar recursos.		
	Criterio de aceptación:	El sistema muestra listado filtrable y ordenable de pedidos. Incluye buscador por nombre de cliente o folio. Permite exportar los datos.		

Figura 8.10. Ejemplo Mensaje de la Reunión de Refinamiento del Product Backlog

Tabla 8.1 Descripción del Mensaje de la Reunión de Refinamiento del Product Backlog

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Programación de la reunión de demostración del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Asistentes	Listado de los asistentes a la reunión:
	Nombre: Nombre de la persona, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.

Historias de usuario	Rol: Rol del asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
	Status: Indica si su asistencia es obligatoria u opcional, valor numérico.
	Tipo de asistencia: Valores permitidos Síncrona o Asíncrona.
	Listado de las historias de usuario del <i>Sprint Backlog</i> .
	Clave: Clave de la historia de usuario, formato HUddd, ddd significa que son tres dígitos.
	Nombre: Nombre de la historia de usuario, longitud máxima 80
	Tiempo Estimado: Horas estimadas, valor numérico entero
	Descripción: Descripción detallada de la Historia de Usuario, longitud máxima de 500 caracteres.
	Criterios de Aceptación: Especificación de los criterios de aceptación de la Historia de Usuario, los cuales los deben validar el <i>Product Owner</i> y/o los <i>Stakeholders</i> , longitud máxima de 1000 caracteres.

Proyecto	Creación de una plataforma WEB					
Fecha y hora	04/03/2023 10:00					
Retroalimentación de la Reunión de Refinamiento del Product Backlog, Sprint n						
País	Estados Unidos					
Idioma	Inglés					
Emisor	Pedro Páramo	Developer Team				
Destinatarios	Nombre	Rol				
	Jorge Hernández	Product Owner				
	Miguel López	Scrum Master				
	Juan Pérez	Developer Team				
Retroalimentación	Comentarios y/o dudas					

Figura 8.11. Ejemplo de Mensaje de la Retroalimentación Reunión de Refinamiento del Product Backlog

Tabla 8.2 Atributos del mensaje de retroalimentación generado en la ceremonia de refinamiento del Product Backlog

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Fecha y hora de la retroalimentación por parte de algunos de los destinatarios del mensaje de la reunión de refinamiento el product backlog del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Emisor	Nombre: Nombre del emisor del mensaje, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.

	Rol: Rol del emisor, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Destinatarios	Listado de los destinatarios a los que se les dirige el mensaje:
	Nombre: Nombre del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Retroalimentación	Comentarios y/o dudas del receptor del mensaje de la reunión de refinamiento el product backlog del <i>Sprint</i> , longitud máxima de 2000 caracteres.

El evento de la Reunión de la Planificación del Sprint se muestra en el diagrama de la Figura 8.12, y en Figura 8.13 se muestra un ejemplo de un mensaje generado de esta ceremonia, también en la Figura 8.14 se muestra las historias de usuario, dividida en tareas.

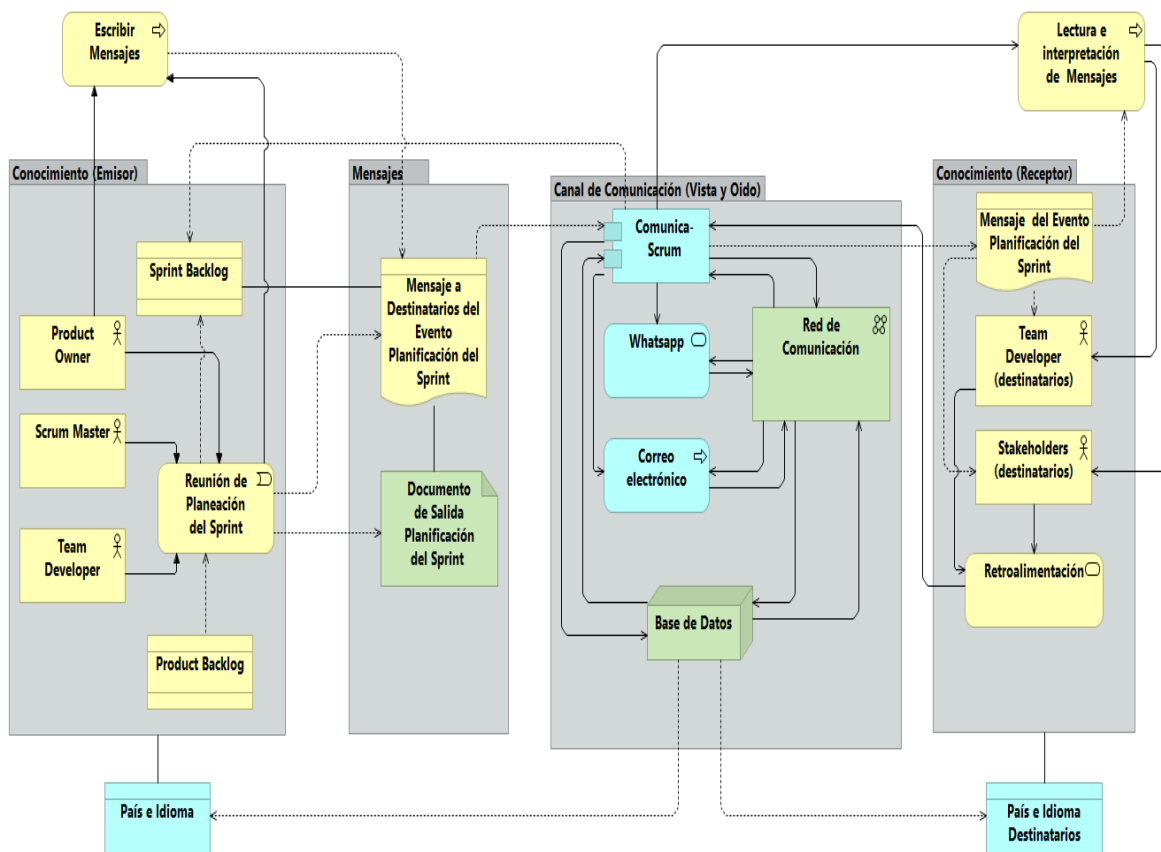


Figura 8.12. Adaptación del modelo de comunicación (Reunión de planificación del Sprint)

Reunión de Planeación del Sprint				
Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Fecha y hora	29 de Septiembre de 2025 a las 11:00			
Planificación del Sprint 3				
Idioma	Español			
País	México			
Asistentes:	Nombre	Rol	Status	Tipo de Asistencia
	Jmiguél	Product Owner	Obligatorio	Síncrona
	Jorenzo	Developers	Obligatorio	Síncrona
	Brayan	Developers	Obligatorio	Síncrona
	Samuel	Developers	Obligatorio	Síncrona
	Jovany	Scrum Master	Obligatorio	Síncrona
	Yessenia	Product Owner	Obligatorio	Síncrona
Sprint Backlog	HU	Nombre	Prioridad	Estimación (hrs)
	HU012	Cálculo de tiempos estimados de servicio	Alta	75
	HU013	Sistema de alertas internas e incidencias	Alta	75
	HU014	Generación de reportes de desempeño	Alta	60
	HU015	Notificaciones automáticas de compra y seguimiento	Alta	75
	HU016	Libro de visitas con mensajes y recuerdos	Alta	75
			Horas:	360
			Días:	45.00
Objetivo del sprint	Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.			

Figura 8.13. Ejemplo de una plantilla de la Ceremonia de Planeación del Sprint

Historias de usuario dividida en tareas			
HU	Nombre	Estimación (hrs)	Developer
HU012	Cálculo de tiempos estimados de servicio	75	
	Diseñar y construir la interfaz de usuario principal	10	Jorenzo
	Implementar la lógica de negocio central del módulo	25	Jorenzo
	Modelar los casos de uso del sistema	5	Jorenzo
	Diseñar las clases y sus relaciones principales	5	Jorenzo
	Crear el diagrama de secuencia del flujo funcional	5	Jorenzo
	Desarrollar pruebas unitarias del componente	5	Jorenzo
	Realizar pruebas de integración entre módulos	10	Jorenzo
	Coordinar pruebas de aceptación del usuario (UAT)	10	Jorenzo
HU013	Sistema de alertas internas e incidencias	75	
	Diseñar y construir la interfaz de usuario principal	10	Jorenzo
	Implementar la lógica de negocio central del módulo	25	Jorenzo
	Modelar los casos de uso del sistema	5	Jorenzo
	Diseñar las clases y sus relaciones principales	5	Jorenzo
	Crear el diagrama de secuencia del flujo funcional	5	Brayan
	Desarrollar pruebas unitarias del componente	5	Brayan
	Realizar pruebas de integración entre módulos	10	Brayan
	Coordinar pruebas de aceptación del usuario (UAT)	10	Brayan

Figura 8.14. Ejemplo de una plantilla de la Ceremonia de Planeación del Sprint, HU Divididas

En la Tabla 8.3 se describe cada uno de los elementos del Mensaje de la Reunión de Planeación del *Sprint* y en la Figura 8.15 se muestra un ejemplo del Mensaje de la Retroalimentación de la Reunión de Planificación del *Sprint*, y en la Tabla 8.4 se muestra la descripción de cada uno de los elementos que componen el mensaje de la Retroalimentación de la Reunión de Planeación del *Sprint*.

Tabla 8.3 Descripción de la Plantilla de la Ceremonia de Planeación del *Sprint*

Atributo	Descripción	
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.	
Fecha y hora	Programación de la reunión de demostración del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm	
Fecha Inicio	Fecha en que iniciará el proyecto, formato: dd/mm/yyyy	
Fecha Final	Fecha planeada en que terminará el proyecto, formato: dd/mm/yyyy	
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30	
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.	
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.	
Asistentes	Listado de los asistentes a la reunión:	
	Nombre:	Nombre de la persona, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol:	Rol del asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
	Status:	Indica si su asistencia es obligatoria u opcional, valor numérico.
	Tipo de asistencia:	Valores permitidos Síncrona o Asíncrona.
Historias de usuario	Listado de las historias de usuario del <i>Sprint Backlog</i> .	
	Clave: Clave de la historia de usuario, formato HUddd, ddd significa que son tres dígitos.	
	Nombre: Nombre de la historia de usuario, longitud máxima 80	Nombre Tarea: Nombre de la tarea, longitud máxima 80
		Estimación: Estimación en horas de la tarea.
		Desarrollador: Nombre del desarrollador responsable de realizar la tarea.
	Prioridad: Indica la prioridad, valor numérico, rango permitido de 1 a 10, el valor 1 tiene la máxima prioridad.	
	Estimación: Horas estimadas, valor numérico entero	
Porcentaje de avance del proyecto	Valor numérico en formato de porcentaje que indique el avance real del proyecto al finalizar el <i>Sprint</i> en revisión.	
Objetivo del <i>Sprint</i>	Es la descripción del objetivo del <i>Sprint</i> , este se establece al iniciar el <i>Sprint</i> , longitud máxima de 500 caracteres.	
Comentarios	Comentarios surgidos durante la reunión por parte de los asistentes, longitud máxima de 2000 caracteres.	

Proyecto	Creación de una plataforma WEB							
Fecha y hora	04/03/2023 10:00							
Retroalimentación de la Planeación del <i>Sprint</i> n								
País	Estados Unidos							
Idioma	Inglés							
Emisor	Pedro Páramo	Developer Team						
Destinatarios	Nombre	Rol						
	Jorge Hernández	Product Owner						
	Miguel López	Scrum Master						
	Juan Pérez	Developer Team						
Retroalimentación	Comentarios y/o dudas							

Figura 8.15. Ejemplo de un Plantilla de Retroalimentación de la Ceremonia de Planeación del *Sprint*

Tabla 8.4 Descripción de la Plantilla de la Retroalimentación de la Ceremonia de Planificación del *Sprint*

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Fecha y hora de la retroalimentación por parte de algunos de los destinatarios del mensaje de la reunión de planificación del <i>Sprint</i> n, formato: dd/mm/yyyy hh:mm
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Emisor	Nombre: Nombre del emisor del mensaje, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del emisor, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Destinatarios	Listado de los destinatarios a los que se les dirige el mensaje:
	Nombre: Nombre del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Retroalimentación	Comentarios y/o dudas del receptor del mensaje de la reunión de planeación del <i>Sprint</i> , longitud máxima de 2000 caracteres.

El evento de la Reunión Diaria se muestra en el diagrama de la Figura 8.16, y en Figura 8.17 se muestra un ejemplo de un mensaje generado de esta ceremonia.

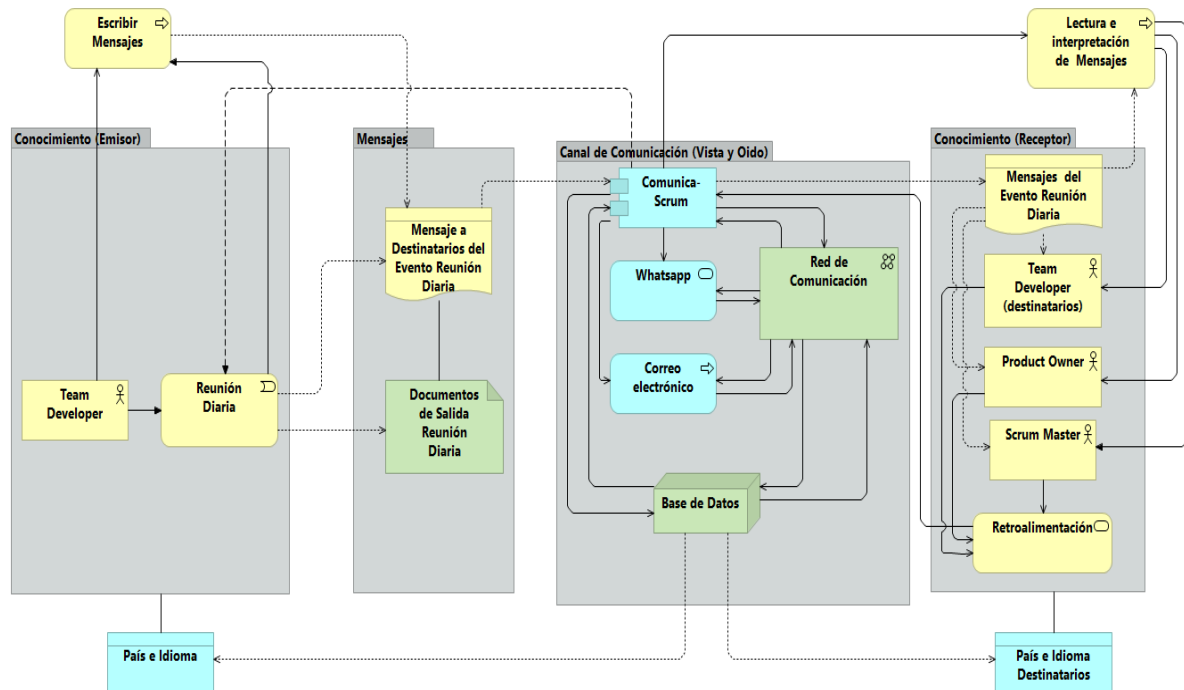


Figura 8.16. Adaptación del modelo de comunicación (Reunión Diaria)

Reporte de Reunión Diaria				
Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Fecha	17 de Octubre de 2025 a las 09:00			
Reunión Diaria, Sprint 3				
País	México			
Idioma	Español			
Asistentes:	Nombre	Rol	Tipo Asistencia	País
	Jmiguél	Product Owner	Sincrónica	México
	Lorenzo	Developers	Sincrónica	México
	Brayan	Developers	Sincrónica	México
	Samuel	Developers	Sincrónica	México
	Jovany	Scrum Master	Sincrónica	México
	Yesenia	Product Owner	Sincrónica	México
Luis Alberto Lorenzo Pérez				
Obstáculos del día anterior:	Nuevos requerimientos impactan validaciones existentes			
Trabajo realizado el día anterior:	Implementar la lógica de negocio central del módulo: implementación de casos principales y validaciones; Modelar los casos de uso del sistema: refinamiento de flujos y precondiciones/postcondiciones			
Planificación del día de hoy:	Modelar los casos de uso del sistema: refinamiento de flujos y precondiciones/postcondiciones; Diseñar las clases y sus relaciones principales: definición de entidades, repositorios y servicios			
Marcos Brayan Hernandez Mejía				
Obstáculos del día anterior:	Nuevos requerimientos impactan validaciones existentes			
Trabajo realizado el día anterior:	Implementar la lógica de negocio central del módulo: implementación de casos principales y validaciones			
Planificación del día de hoy:	Implementar la lógica de negocio central del módulo: implementación de casos principales y validaciones			
Samuel Peralta Aréchiga				
Obstáculos del día anterior:	Error de sincronización entre módulos refactorizados			
Trabajo realizado el día anterior:	Realizar pruebas de integración entre módulos: pruebas API y contratos entre módulos; Coordinar pruebas de aceptación del usuario (UAT): preparación de scripts de UAT y criterios			
Planificación del día de hoy:	Coordinar pruebas de aceptación del usuario (UAT): preparación de scripts de UAT y criterios			

Figura 8.17. Ejemplo de una Plantilla de la Reunión Diaria

En la Tabla 8.5 se describe cada uno de los elementos del Mensaje de la Reunión Diaria.

Tabla 8.5 Descripción del Plantilla de la Reunión Diaria

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Programación de la reunión diaria, formato: dd/mm/yyyy hh:mm
Número del Sprint	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Asistentes	Listado de los asistentes a la reunión: Nombre: Nombre de la persona, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60. Rol: Rol del asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30. Tipo de asistencia: Valores permitidos Síncrona o Asíncrona. País: Nombre del país al que pertenece el asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50
Listado de lo realizado por cada desarrollador	Trabajo realizado el día anterior: Es la descripción de lo realizado el día anterior por el desarrollador, longitud máxima de 500 caracteres. Obstáculos: Es la descripción de los obstáculos presentados durante el día de trabajo por el desarrollador, longitud máxima de 500 caracteres Planificación del siguiente día: Es la descripción de lo que va hacer el desarrollador el siguiente día de trabajo, longitud máxima de 500 caracteres

El evento de la **Ejecución del Sprint** se muestra en el diagrama de la Figura 8.18 y en Figura 8.19 se muestra un ejemplo de un mensaje generado de esta ceremonia.

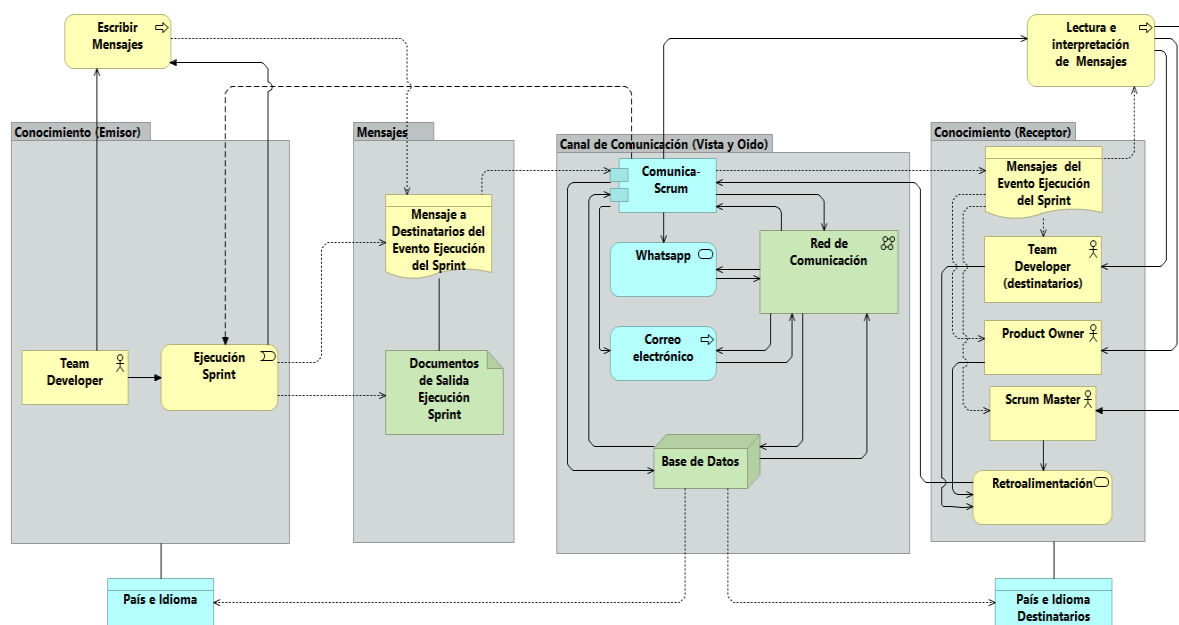


Figura 8.18. Adaptación del modelo de comunicación (Ejecución del Sprint)

Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)																												
Reporte de Reunión Diaria																												
Proyecto	Desarrollo de la plataforma Funeraria XYZ																											
Fecha	17 de Octubre de 2025 a las 09:00																											
Ejecucion del sprint 3																												
País	México																											
Idioma	Español																											
HU	Nombre				Septiembre					Octubre																		
HU012	Calculo de tiempos estimados de servicio	Developer	Horas Estimadas	Horas Reales	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17					
1	Diseñar y construir la interfaz de usuario principal	lorenzo	10	10	8/2	2/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
2	Implementar la logica de negocio central del módulo	lorenzo	25	25	0/0	6/19	8/11	8/3	3/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
3	Modelar los casos de uso del sistema	lorenzo	5	5	0/0	0/0	0/0	0/0	5/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
4	Diseñar las clases y sus relaciones principales	lorenzo	5	5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	5/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
5	Crear el diagrama de secuencia del flujo funcional	lorenzo	5	5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	3/2	2/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
6	Desarrollar pruebas unitarias del componente	lorenzo	5	6	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	6/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
7	Realizar pruebas de integración entre módulos	lorenzo	10	10	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	8/2	2/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
8	Coordinar pruebas de aceptación del usuario (UAT)	lorenzo	10	10	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	6/4	4/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
HU013	Sistema de alertas internas e incidencias	Developer	Horas Estimadas	Horas Reales	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17					
1	Diseñar y construir la interfaz de usuario principal	lorenzo	10	10	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	4/6	0/0	0/0	6/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
2	Implementar la logica de negocio central del módulo	lorenzo	25	24	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	8/17	8/9	8/0	0/0	0/0	0/0	0/0	0/0	0/0
3	Modelar los casos de uso del sistema	lorenzo	5	5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	5/0
4	Diseñar las clases y sus relaciones principales	lorenzo	5	5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	5/0
5	Crear el diagrama de secuencia del flujo funcional	Brayan	5	5	5/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
6	Desarrollar pruebas unitarias del componente	Brayan	5	5	3/2	2/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
7	Realizar pruebas de integración entre módulos	Brayan	10	10	0/0	6/4	4/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0

Figura 8.19. Ejemplo de una Plantilla de la Ejecución del Sprint

En la Tabla 8.6 se describe cada uno de los elementos del Plantilla de Ejecución del *Sprint*, en donde se indica el esfuerzo dedicado y pendiente de cada uno los días.

Tabla 8.6 Descripción de la Plantilla de la Ejecución del Sprint

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha	Ejecución del <i>Sprint</i> , formato: dd/mm/yyyy
Número del Sprint	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Historias de usuario	Listado de las historias de usuario del <i>Sprint Backlog</i> .
	Clave: Clave de la historia de usuario, formato HUddd, ddd significa que son tres dígitos.
	Nombre HU: Nombre de la historia de usuario, longitud máxima 80
	Clave de la Tarea: Clave de tarea, valor numérico.

	Desarrollador: Nombre del desarrollador, longitud máxima 80	Nombre de la Tarea: Nombre de la tarea, longitud máxima 40.
		Horas Estimadas: Horas estimadas de la tarea, valor numérico entero.
		Horas Reales: Esfuerzo real aplicado a la tarea, valor numérico entero.
		Listado del esfuerzo dedicado: Horas dedicadas y faltantes en un determinado día de una tarea, formato nn/nn.
	Horas Estimadas: Horas estimadas de la historia de usuario, valor numérico entero.	
	Horas Reales: Esfuerzo real aplicado a la historia de usuario durante el <i>Sprint</i> .	
	Listado de días del % de Avance: Muestra el porcentaje de avance por cada día del <i>Sprint</i> , valor en porcentaje.	

El evento de la **Reunión de Retrospectiva del Sprint** se muestra en el diagrama de la Figura 8.20 y en Figura 8.21 se muestra un ejemplo de un mensaje generado de esta ceremonia.

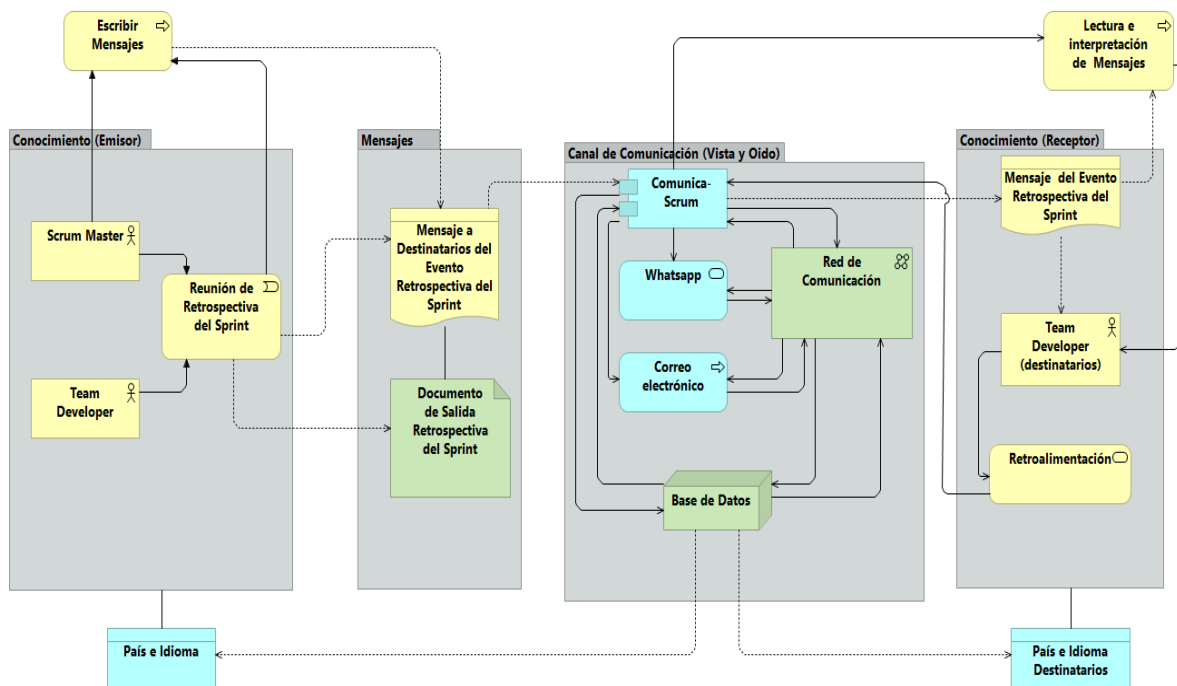


Figura 8.20. Adaptación del modelo de comunicación (Retrospectiva del Sprint).

Reunión de Retrospectiva del Sprint

Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Reunión de Restrospectiva del Sprint 3				
Fecha y hora reunión	17 de Octubre de 2025 a las 18:00			
País	México			
Idioma	Español			
Asistentes:	Nombre	Rol	Status	Tipo Asistencia
	Jmiguél	Product Owner	Obligatorio	Síncrona
	lorenzo	Developers	Obligatorio	Síncrona
	Brayan	Developers	Obligatorio	Síncrona
	Samuel	Developers	Obligatorio	Síncrona
	Jovany	Scrum Master	Obligatorio	Síncrona
	Yesenia	Product Owner	Obligatorio	Síncrona
Objetivo:	17/10/2025 18:00; Retrospectiva Sprint. Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.			
Reflexiones/Comentarios (lorenzo):	Logramos una integración visual consistente y la lógica de validación en tiempo real funcionó correctamente. Sin embargo, note que la revisión de UI con el Product Owner se agendó con poca anticipación, lo que redujo el margen de corrección antes del cierre del sprint.			
Oportunidades de mejora (lorenzo):	Propongo que en Comunica-Scrum programemos revisiones visuales semanales automáticas, vinculadas al tablero de tareas de diseño, para evitar retrasos en validaciones estéticas.			
Reflexiones/Comentarios (Brayan):	A pesar de haber tenido tres días de incapacidad, logré completar las pruebas de integración del sistema de incidencias y la generación de reportes. El único pendiente fue la conexión SMTP para las notificaciones, que no se pudo completar por falta de credenciales del servidor.			
Oportunidades de mejora (Brayan):	Sugiero agregar un entorno de pre-producción con servicios autenticados de prueba, para no depender del entorno final en tareas críticas como el correo y los reportes.			
Reflexiones/Comentarios (Samuel):	El libro de visitas funcionó correctamente y se integro sin errores. Sin embargo, en las pruebas de estres note lentitud al registrar mensajes simultáneos, lo que sugiere que necesitamos un proceso asincrono de escritura.			
Oportunidades de mejora (Samuel):	Propongo implementar un sistema de workers asincronos usando Celery y Redis para optimizar las operaciones de insercion masiva.			

Figura 8.21. Ejemplo de una Plantilla de la Ceremonia de Retrospectiva del Sprint

En la Tabla 8.7 se describe cada uno de los elementos del Mensaje de la Reunión de Retrospectiva del *Sprint*.

Tabla 8.7 Descripción de la Plantilla de la Ceremonia de Retrospectiva del Sprint

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Programación de la Reunión de Retrospectiva del <i>Sprint n</i> , formato: dd/mm/yyyy hh:mm
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Asistentes	Listado de los asistentes a la reunión: Nombre: Nombre de la persona, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.

	Rol: Rol del asistente, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
	Status: Indica si su asistencia es obligatoria u opcional, valor numérico.
	Tipo de asistencia: Valores permitidos Síncrona o Asíncrona.
Reflexiones/Comentarios	Comentarios surgidos durante la reunión por parte de los asistentes, longitud máxima de 2000 caracteres.
Oportunidades de Mejora	Es la descripción de las oportunidades de mejora que se deberán considerar en el siguiente <i>Sprint</i> , longitud máxima de 2000 caracteres.

En la Figura 8.22 se muestra un ejemplo del Mensaje de la Retroalimentación de la Reunión de Retrospectiva del *Sprint*, y en la Tabla 8.8 se muestra la descripción de cada uno de los elementos que componen el mensaje de la Retroalimentación de la Reunión de Retrospectiva del *Sprint*.

Proyecto	Creación de una plataforma WEB					
Fecha y hora	04/03/2023 10:00					
Retroalimentación de la Retrospectiva del <i>Sprint</i> n						
País	Estados Unidos					
Idioma	Inglés					
Emisor	Pedro Páramo	Developer				
Destinatarios	Nombre	Rol				
	Jorge Hernández	Scrum Master				
	Juan Pérez	Developer Team				
Retroalimentación	Comentarios y/o dudas					

Figura 8.22. Ejemplo de un Plantilla de Retroalimentación de la Ceremonia de Retrospectiva del *Sprint*

Tabla 8.8 Descripción de la Plantilla de la Retroalimentación de la Ceremonia de Retrospectiva del *Sprint*

Atributo	Descripción
Nombre del Proyecto	Nombre del proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 100.
Fecha y hora	Fecha y hora de la retroalimentación por parte de algunos de los destinatarios del mensaje de la reunión de retrospectiva del <i>Sprint</i> n, formato: dd/mm/yyyy hh:mm
Número del <i>Sprint</i>	Valor numérico entero, rango de 1 a 30
País	Nombre del país al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 50.
Idioma	Idioma al que pertenece el proyecto, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Emisor	Nombre: Nombre del emisor del mensaje, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del emisor, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.

Destinatarios	Listado de los destinatarios a los que se les dirige el mensaje:
	Nombre: Nombre del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 60.
	Rol: Rol del destinatario, acepta el conjunto de caracteres ASCII imprimibles, máximo de una longitud de 30.
Retroalimentación	Comentarios y/o dudas del receptor del mensaje de la reunión de retrospectiva del <i>Sprint</i> , longitud máxima de 2000 caracteres.

Anexo C Definición de términos que deben conocer todos los involucrados del proyecto

El desarrollo de software tanto en entornos de *Global Software Development* (GSD) como en equipos presenciales que siguen el marco Scrum exige el uso de una terminología y unas prácticas comunes que permitan mantener una comunicación fluida y una colaboración efectiva, incluso cuando los miembros están distribuidos en distintas ubicaciones geográficas.

Scrum: Marco ágil destinado a la gestión de proyectos y al desarrollo de software que promueve la colaboración constante, la transparencia y la capacidad de adaptación del equipo [61].

Team Developer (equipo de desarrollo): Grupo multifuncional responsable de entregar incrementos del producto de software [61].

Scrum Master: Facilita el proceso, elimina obstáculos y asegura que se sigan las prácticas de Scrum [61].

Product Owner (Propietario del producto): Representa los intereses del cliente y define las funcionalidades del producto [61].

Team Scrum (Equipo Scrum): Integrado por el *Product Owner*, *Scrum Master* y *Team Developer* [61].

Stakeholders (partes interesadas): Son todas las partes interesadas que tienen algún interés, influencia o participación en un proyecto, proceso o sistema. Estas partes interesadas pueden afectar o ser afectadas por las acciones, decisiones y resultados del proyecto, ejemplos (clientes, usuarios finales, patrocinadores, gerentes, etc.) [61].

User History (Historia de usuario): Es una técnica utilizada en el desarrollo de software y en metodologías ágiles, como Scrum, para describir una característica o funcionalidad del sistema desde la perspectiva del usuario final, es como un requisito, están inspiradas en las conversaciones con los usuarios finales, encapsulan las necesidades y expectativas del cliente en fragmentos manejables [61].

Task (Tarea): Es la una unidad más pequeña de trabajo que contribuye a la implementación de una historia de usuario específica, son acciones más detalladas que deben llevarse a cabo para completar la implementación de esa historia [96].

Estimación: Es una medida relativa al esfuerzo necesario para completar una tarea o una historia de usuario, puede medirse en puntos de historia o en horas [96].

Prioridad: Es un valor que se le asigna a una historia de usuario, para saber el orden en el que el equipo de desarrollo las atenderá, el *Product Owner* es el encargado de asignar la prioridad [96].

Epic (Épica): Una "épica" es una unidad de trabajo grande y compleja que generalmente no se puede completar en un solo *Sprint*. Las épicas son historias de usuario a una escala mayor y suelen describir funcionalidades o requerimientos a nivel de alto nivel. Estas historias son descompuestas en historias de usuario más pequeñas y manejables durante el proceso de planificación y ejecución [96].

Product Backlog (*Backlog* del producto): El *Product Backlog* es una lista emergente y ordenada de las historias de usuarios de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el *Scrum Team* [61].

Sprint: Es un periodo de tiempo fijo y regular durante el cual se lleva a cabo el trabajo de desarrollo y entrega de un incremento de producto de software potencialmente entregable. La duración típica de un sprint es de dos a cuatro semanas [61].

Sprint Backlog (*Backlog* del *Sprint*): El *Sprint Backlog* se compone del Objetivo del *Sprint* (por qué), el conjunto de elementos del *Product Backlog* seleccionados para el *Sprint* (qué), así como un plan de acción para entregar el incremento (cómo) [61].

Backlog Grooming Meeting (Reunión de refinamiento del *backlog* del producto): El refinamiento del *Product Backlog* es el acto de dividir y definir aún más los elementos del *Product Backlog* en elementos más pequeños y precisos [96].

Sprint Planning Meeting (Reunión de planificación del *Sprint*): La *Sprint Planning* inicia el *Sprint* al establecer el trabajo que se realizará para el *Sprint*. El *Scrum Team* crea este plan resultante mediante trabajo colaborativo. El *Product Owner* propone cómo el producto podría *incrementar* su valor y utilidad en el *Sprint* actual, los *Developers* seleccionan elementos del *Product Backlog* para incluirlos en el *Sprint* actual y planifican el trabajo necesario para crear un *incremento* que cumpla con la Definición de Terminado [61].

Daily Scrum (reunión diaria Scrum): La *Daily Scrum* es un evento de 15 minutos para los *Developers* del *Scrum Team*, estos pueden seleccionar la estructura y las técnicas que deseen, siempre que su *Daily Scrum* se centre en el progreso hacia el Objetivo del *Sprint* y produzca un plan viable para el siguiente día de trabajo [61].

Sprint Execution (Ejecución del *Sprint*): Es la fase en la que el equipo de desarrollo trabaja en la implementación de las historias de usuario y tareas seleccionadas durante la planificación del sprint, para entregar un incremento de producto al final del sprint [61].

Sprint Review Meeting (Reunión de demostración del *Sprint*): El propósito de la *Sprint Review* es inspeccionar el resultado del *Sprint* y determinar futuras adaptaciones. El *Scrum Team* presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto. En esta reunión se revisa si las historias de usuario que fueron asignadas al Sprint en la planeación fueron completadas y aceptadas por parte del *Product Owner* [61].

Product Increment (Incremento de producto): Se refiere al resultado tangible y potencialmente entregable al final de cada sprint, al final de cada iteración (*Sprint*), el producto es mejorado y ampliado, y puede ser presentado a los *Stakeholders* o incluso puesto en producción si así se decide [96].

Iterative and Incremental (Iterativo e Incremental): Son términos que describen enfoques de desarrollo de software que se centran en la entrega continua de mejoras al producto a lo largo del tiempo, en lugar de tratar de entregar todo el producto al final del desarrollo [96].

Feedback (Retroalimentación): Se refiere al proceso de proporcionar y recibir comentarios sobre el trabajo realizado. Esta retroalimentación puede provenir de diferentes fuentes, como el cliente, los usuarios finales, otros miembros del equipo, o cualquier otra persona interesada en el proyecto, esta retroalimentación se da normalmente en las diferentes ceremonias de Scrum [96].

Sprint Retrospective Meeting (Reunión de retrospectiva del *Sprint*): El *Scrum Team* analiza qué salió bien durante el *Sprint*, qué problemas encontró y cómo se resolvieron (o no) esos problemas. El *Scrum Team* identifica los cambios más útiles para mejorar su efectividad. Las mejoras más impactantes se abordan lo antes posible. Incluso se pueden agregar al *Sprint Backlog* para el próximo *Sprint* [61].

Definition of Done (DoD) (Definición de hecho): Es un concepto fundamental en Scrum y otras metodologías ágiles. La DoD establece los criterios que una historia de usuario o cualquier elemento de trabajo debe cumplir para considerarse completado y listo para la entrega. Es una lista de verificación que ayuda a garantizar que el trabajo realizado cumple con los estándares de calidad y las expectativas del equipo [96].

Burn-down Chart (Gráfica de trabajo pendiente): Es una herramienta visual utilizada en Scrum y otras metodologías ágiles para representar el trabajo restante en un sprint o proyecto a lo largo del tiempo. Esta gráfica proporciona una visión rápida del progreso del equipo y ayuda a prever si se cumplirán los objetivos del sprint o proyecto en el tiempo planificado [96].

Burn-ups Chart (Gráfica de trabajo completado): Es herramienta visual utilizada en Scrum y otras metodologías ágiles para representar el progreso del trabajo a lo largo del tiempo. A diferencia de la *Burn-down Chart*, que se centra en el trabajo restante, la *Burn-up Chart* muestra la acumulación del trabajo completado a medida que avanza el tiempo [96].

Anexo D Aplicación Comunica-Scrum

Gestión de proyectos Scrum

A continuación, se muestran algunas pantallas de la aplicación Comunica-Scrum.

En la Figura 8.23 muestra la pantalla de inicio de sesión, donde el usuario deberá capturar sus credenciales.

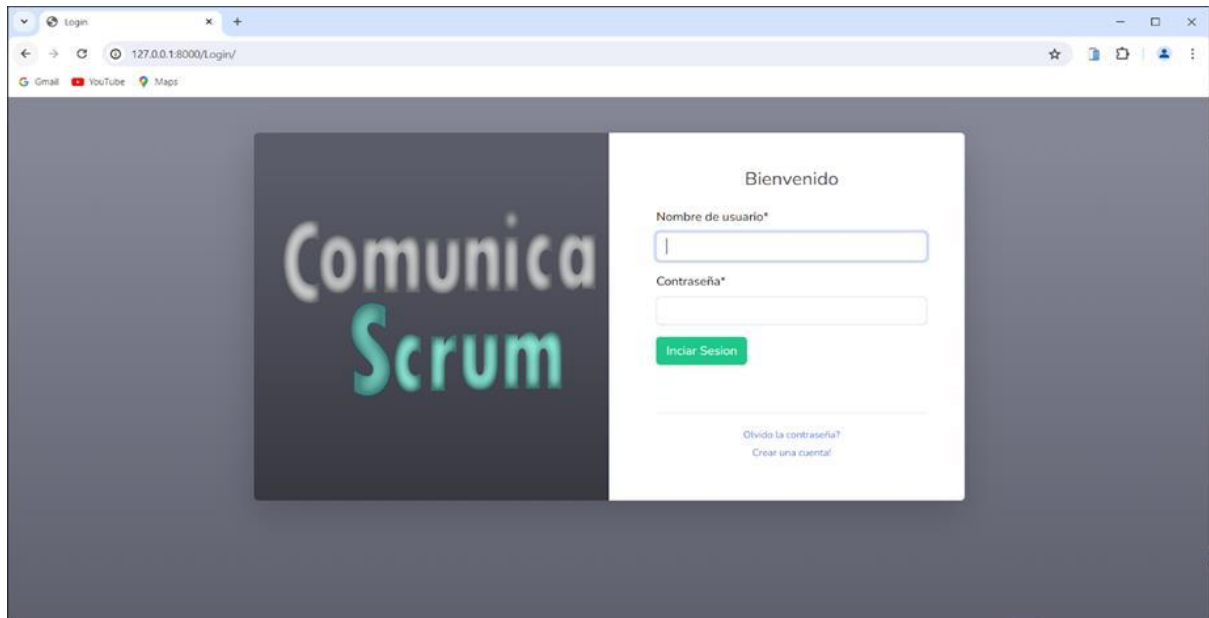


Figura 8.23. Pantalla de inicio se sesión

El registro de nuevos usuarios se hará desde la pantalla de inicio se sesión, en el botón Crear Cuenta, ver Figura 8.24

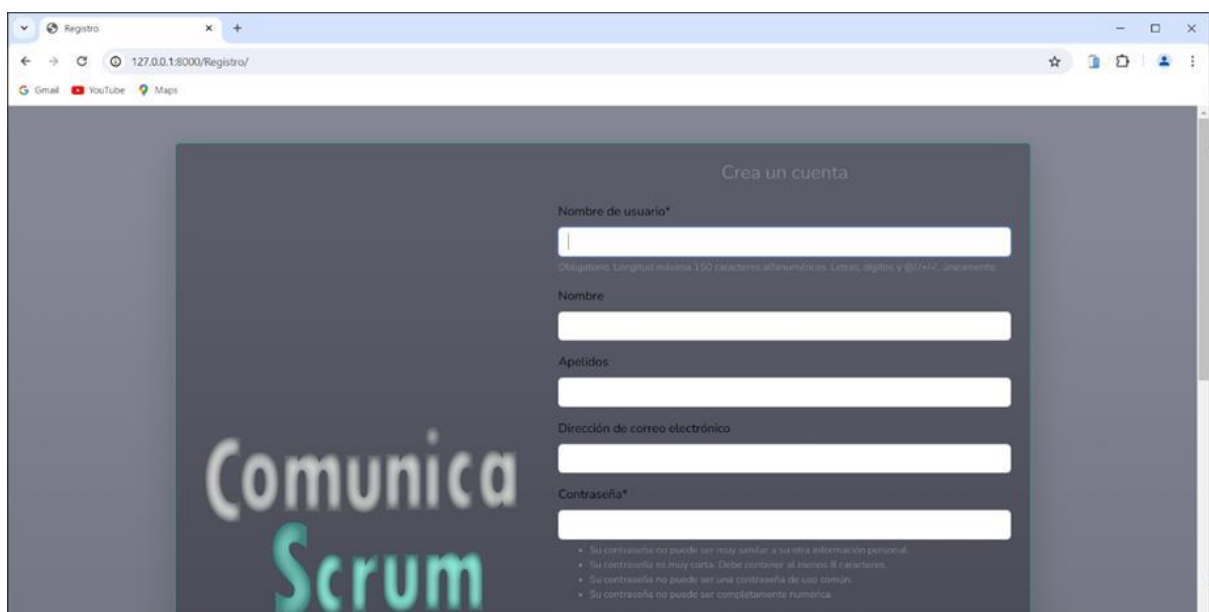


Figura 8.24. Formulario de registros de usuarios

En la Figura 8.25 se muestra como agregar nuevos proyectos en la aplicación.

Comunica Scrum

Buscar...

Hola jefe

Menu

Proyectos

INICIO DE PROYECTOS

Ver proyectos

Registrar un proyecto

COMPLEMENTOS

Avances

Seguimiento

Agregar proyecto

Nombre de Proyecto*

Ingrese el Nombre del Proyecto

Fecha de Inicio

dd/mm/aaaa

Fecha de Conclusión

dd/mm/aaaa

Objetivo

Ingrese el Objetivo

Impacto

Ingrese el Impacto

Figura 8.25. Formulario de proyectos

Después de agregar un proyecto, este se reflejará en el formulario de listado de proyectos, como se ve en la Figura 8.26.

Comunica Scrum

Buscar...

Hola jefe

Menu

Proyectos

INICIO DE PROYECTOS

Ver proyectos

Registrar un proyecto

COMPLEMENTOS

Avances

Seguimiento

BIENVENIDO A LA SECCIÓN DE PROYECTO

#	Nombre	Fecha de inicio	Fecha de conclusión	Objetivo	Impacto	Homologacion de visión	Opciones
1	Desarrollo de una aplicación web con ASP.NET	11 de Abril de 2024	27 de Abril de 2024	El marco ASP.NET amplía la plataforma para desarrolladores de .NET con el objetivo de crear aplicaciones basadas en web.	.NET ofrece la ventaja de un entorno específico de lenguaje a lo largo del proceso de desarrollo de código	NET es una plataforma de código abierto para crear aplicaciones de escritorio, web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo.	
2	Api rest	11 de Abril de 2024	24 de Abril de 2024	Intercambiar información de manera segura a través de Internet.	Una API Rest permite que la aplicación acceda a bases de datos desde diferentes servidores, lo que a menudo es importante para el desarrollo en aplicaciones grandes. Por lo tanto, su uso garantiza una mayor visibilidad y credibilidad a la hora de utilizar estos recursos.	Arquitectura cliente-servidor compuesta de clientes, servidores y recursos, con la gestión de solicitudes a través de HTTP.	
3	Aplicación móvil	11 de Abril de	20 de Abril de 2024	Incluir estrategias para fomentar la participación continua, como notificaciones relevantes y	Una de las ventajas más significativas de tener una aplicación móvil es la accesibilidad	Con una aplicación móvil, pueden acceder a la información de la empresa en	

Figura 8.26. Listado de proyectos

En la Figura 8.27 se muestra como agregar una historia de usuario.

Agregar Historia de Usuario

Nombre*

Ingreso o supresión de roles

Fecha de creacion*

25/05/2024

Fecha de actualizacion

25/05/2024

Descripcion*

Ingreso o supresión de roles

Horas Estimadas

9

Prioridad

1

Criterios de Aceptacion

2

Figura 8.27. Registro de historia de usuario

Después de agregar una historia de usuario, esta se mostrará en el listado correspondiente, como se ve en el Figura 8.28.

Comunica Scrum

Buscar...

Hola jefe

Menu

Proyectos

Historias de Usuario

Sprints

Historias de Usuarios

#	Nombre	Fecha de creacion	Fecha de actualizacion	Descripcion	Asignar	Opciones
1	Pantalla de acceso	14 de Mayo de 2024	None	Pantalla de acceso	incompleta	  
2	Memoria técnica	14 de Mayo de 2024	None	memoria	incompleta	  
3	Ingreso o supresión de roles	25 de Mayo de 2024	25 de Mayo de 2024	Ingreso o supresión de roles	incompleta	  

Figura 8.28. Listado de historias de usuario

Para agregar una tarea de utiliza el formulario que se muestra en la Figura 8.29.

Crear de Tarea

Nombre:

Tarea 1 - Establecer expectativas claras

Clasificación de tarea:

Product Owner

Nivel de prioridad:

Alta

Fecha de inicio planeado:

25/05/2024

Fecha de terminacion planeada:

26/05/2024

Horas estimadas:

5

Cancelar

Confirmar

Figura 8.29. Creación de tarea para una historia de usuario

En la Figura 8.30 se muestra como agregar un Sprint.

Agregar Sprint

Nombre*

Sprint 1

Fecha de Inicio*

05/04/2024

Fecha de Finalizacion

10/04/2024

Numero de Sprint*

1

Ojetivo de Sprint*

Construir aplicaciones web dinámicas y escalables

Cancelar

Confirmar

Figura 8.30. Registro de sprint

El listado de *Sprints* se muestra en la Figura 8.31.

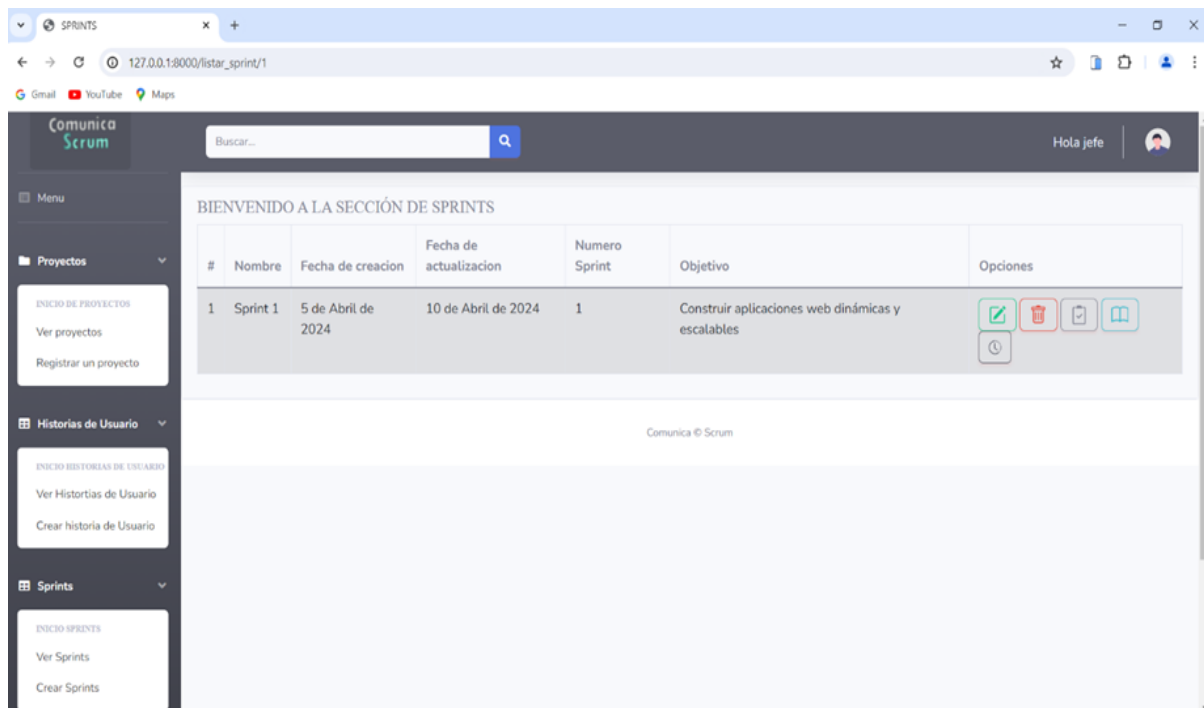


Figura 8.31. Listado de Sprints

Ceremonias Scrum

En la pantalla principal del sistema se muestran las diferentes ceremonias Scrum que contempla la aplicación, ver Figura 8.32.



Figura 8.32. Ceremonias

En este documento se incluirán, a manera de ejemplo, las pantallas asociadas a **la ceremonia de planeación del Sprint**.

En la Figura 8.33 se muestra la pantalla para iniciar la ceremonia de reunión de planeación del Sprint.

REUNIÓN DE PLANEACIÓN DEL SPRINT

Sin registros

Agendar una reunion

El objetivo de la planificación de sprints es definir lo que se puede entregar en el sprint y cómo se conseguirá ese trabajo.

Figura 8.33. Reunión de planeación de sprint sin registros

Al seleccionar el botón “Agendar” se mostrará la pantalla de la Figura 8.34, con el listado de los *Sprints* disponibles.

Sprints disponibles					
#	Proyecto	Sprint	Objetivo	Status	Opciones
1	Desarrollo de la plataforma Funeraria XYZ	Sprint 1	Implementar las funciones principales que permitan gestionar los servicios funerarios personalizados desde la plataforma, incluyendo la moderación de contenido, el registro y control de ventas, la personalización del servicio, el seguimiento de pedidos mediante folios y la validación de cobertura geográfica para los domicilios de los clientes.	Creado	
2	Desarrollo de la plataforma Funeraria XYZ	Sprint 2	Desarrollar las funcionalidades que permitan completar el proceso de contratación del servicio funerario, desde la selección de combos y el cálculo automático de precios, hasta el registro logístico, el procesamiento del pago y la generación de páginas conmemorativas digitales.	Creado	
3	Desarrollo de la plataforma Funeraria XYZ	Sprint 3	Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.	Creado	

Figura 8.34. Listado de Sprints para reunión de planeación del sprint

Una vez seleccionado el Sprint, se enviará al formulario, que se muestra en la Figura 8.35, donde el único dato que se solicitará será el de la fecha y hora en que se llevará a cabo para la reunión de planeación del *Sprint*.

Agendar datos para Reunión de Planeación del Sprint

Emisor Product_Owner

Fecha y Hora (aaaa-mm-dd hh:mm)*

22/02/2024 02:10:10

Figura 8.35. Formulario para agendar reunión de planeación de Sprint

Una vez que se agenda la reunión de planeación del *Sprint*, esta se mostrará en la Figura 8.36.

Reunión de Planeación del Sprint							
Agendar una reunión El objetivo de la planificación de sprints es definir lo que se puede entregar en el sprint y cómo se conseguirá ese trabajo.				Guardar documentos Utilice esta sección para guardar las plantillas generadas o cualquier documento que desee.			
Agendar				Guardar			
#	Proyecto	Fecha	Emisor	Descripción	Status	Sprint	Opciones
3	Desarrollo de la plataforma Funeraria XYZ	18 de Agosto de 2025 a las 13:00	Yesenia	18/08/2025 13:00, Planeación Sprint. Implementar las funciones principales que permitan gestionar los servicios funerarios personalizados desde la plataforma, incluyendo la moderación de contenido, el registro y control de ventas, la personalización del servicio, el seguimiento de pedidos mediante folios y la validación de cobertura geográfica para los domicilios de los clientes.	Comprendido	Sprint 1	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
3	Desarrollo de la plataforma Funeraria XYZ	8 de Septiembre de 2025 a las 11:00	Yesenia	08/09/2025 11:00, Planeación Sprint. Desarrollar las funcionalidades que permitan completar el proceso de contratación del servicio funerario, desde la selección de combos y el cálculo automático de precios, hasta el registro logístico, el procesamiento del pago y la generación de páginas conmemorativas digitales.	Comprendido	Sprint 2	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
3	Desarrollo de la plataforma Funeraria XYZ	29 de Septiembre de 2025 a las 11:00	Yesenia	29/09/2025 11:00, Planeación Sprint. Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.	Comprendido	Sprint 3	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Figura 8.36. Listado de reuniones de planeación del Sprint

Después de generar la ceremonia de planeación del Sprint se podrán agregar los asistentes a la reunión, como se muestra en la Figura 8.37.

Lista de Asistentes Planeacion Sprint						
Agregar Asistente +						
#	Proyecto	Fecha de creacion	Evento Scrum	Usuario	Rol	Opciones
1	Aplicación movil	25 de Mayo de 2024 a las 20:47	Reunión de Planeación del Sprint	tigre	Developers	☐ ☐
2	Aplicación movil	25 de Mayo de 2024 a las 20:47	Reunión de Planeación del Sprint	empleado	Developers	☐ ☐
3	Aplicación movil	25 de Mayo de 2024 a las 20:47	Reunión de Planeación del Sprint	empleado2	Scrum Master	☐ ☐

Figura 8.37. Listado de asistentes para reunión de planeación del Sprint

Asimismo, se podrán seleccionar las historias de usuario que serán parte del *Sprint*, ver Figura 8.38.

Historias de usuario - Planeación Sprint					
#	Nombre	Descripción	Estatus	Horas estimadas	Opciones
1	Seguridad web	Implementar medidas de seguridad web	Capturada	8	☐ ☐ ☐

Historias agregadas					
#	Nombre	Descripción	Estatus	Horas estimadas	Opciones
1	Cambiar de dirección de envío	Cambiar de dirección de envío	En Sprint	6	☐ ☐
2	Ingreso o supresión de roles	Ingreso o supresión de roles	En Sprint	9	☐ ☐

Figura 8.38. Listado de historias de usuario para reunión de planeación del Sprint

Una vez agregados los asistentes y las historias de usuario, se genera el archivo PDF, que será el mensaje que se enviará a los asistentes programados de la reunión de planeación del Sprint, como se muestra en la Figura 8.39.

Reunión de Planeación del Sprint				
Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Fecha y hora	29 de Septiembre de 2025 a las 11:00			
Planificación del Sprint 3				
Idioma	Español			
País	México			
Asistentes:	Nombre	Rol	Status	Tipo de Asistencia
	Jmiguél	Product Owner	Obligatorio	Síncrona
	Jorenzo	Developers	Obligatorio	Síncrona
	Brayan	Developers	Obligatorio	Síncrona
	Samuel	Developers	Obligatorio	Síncrona
	Jovany	Scrum Master	Obligatorio	Síncrona
	Yesenia	Product Owner	Obligatorio	Síncrona
Sprint Backlog	HU	Nombre	Prioridad	Estimación (hrs)
	HU012	Cálculo de tiempos estimados de servicio	Alta	75
	HU013	Sistema de alertas internas e incidencias	Alta	75
	HU014	Generación de reportes de desempeño	Alta	60
	HU015	Notificaciones automáticas de compra y seguimiento	Alta	75
	HU016	Libro de visitas con mensajes y recuerdos	Alta	75
			Horas:	360
			Días:	45.00
Objetivo del sprint	Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.			

Figura 8.39. Plantilla de reunión de planeación del sprint

Los asistentes podrán hacer comentarios durante la reunión, esto se muestra en la Figura 8.40.

Reunión de Planeación del Sprint				
Proyecto	Desarrollo de la plataforma Funeraria XYZ			
Fecha y hora	29 de Septiembre de 2025 a las 11:00			
Planificación del Sprint 3				
Idioma	Español			
País	México			
Asistentes:	Nombre	Rol	Status	Tipo de Asistencia
	Imiguel	Product Owner	Obligatorio	Síncrona
	Lorenzo	Developers	Obligatorio	Síncrona
	Brayan	Developers	Obligatorio	Síncrona
	Samuel	Developers	Obligatorio	Síncrona
	Jovany	Scrum Master	Obligatorio	Síncrona
	Yesenia	Product Owner	Obligatorio	Síncrona
Sprint Backlog	HU	Nombre	Prioridad	Estimación (hrs)
	HU012	Cálculo de tiempos estimados de servicio	Alta	75
	HU013	Sistema de alertas internas e incidencias	Alta	75
	HU014	Generación de reportes de desempeño	Alta	60
	HU015	Notificaciones automáticas de compra y seguimiento	Alta	75
	HU016	Libro de visitas con mensajes y recuerdos	Alta	75
			Horas:	360
			Días:	45.00
Objetivo del sprint	Implementar funcionalidades que fortalezcan la gestión operativa y la interacción automatizada en la plataforma funeraria, incluyendo el cálculo de tiempos estimados de servicio, el registro de incidencias internas, la generación de reportes de desempeño, las notificaciones automáticas al cliente y un libro de visitas digital con mensajes conmemorativos.			
Comentario de Lorenzo:	Sugiero que el cálculo de tiempos (HU012) esté vinculado con las notificaciones automáticas (HU015), para que el cliente reciba actualizaciones más precisas según el avance del servicio.			
Comentario de Brayan:	En el sistema de alertas (HU013) sería útil que Comunica-Scrum registre automáticamente las incidencias detectadas, para mantener trazabilidad y generar reportes en la HU014.			
Comentario de Samuel:	La funcionalidad del libro de visitas (HU016) podría integrarse con la página conmemorativa creada en el sprint anterior, reutilizando el módulo multimedia.			
Comentario de Jovany:	Propongo dividir las HU012 y HU013 en tareas más pequeñas dentro de Comunica-Scrum, ya que ambas implican cálculos automáticos e integración con otros módulos. Así evitaremos cuellos de botella durante el desarrollo.			

Figura 8.40. Reunión de planeación del sprint con comentarios

En la Figura 8.41, se muestra la opción para enviar el mensaje a los asistentes que se agregaron a la reunión de planeación del sprint.

Enviar Mensaje

Nota: Asegurese de que la información sea correcta antes de enviar el mensaje.

Informacion del Mensaje

Proyecto: Aplicación movil

Emisor: Product_Owner

Descripción: Monitorización y gestión de procesos, así como para planificación y coordinación de estrategias y actividades clave para las mismas

Fecha: 22 de Febrero de 2024 a las 02:10

Sprint: Sprint 3

No hay archivos en este mensaje

Guardar documentos

Destinatarios

Asistente: tigre

Asistente: empleado

Asistente: empleado2

Enviar

Figura 8.41. Envío del mensaje para reunión de planeación del Sprint

Si alguno de los asistentes envió un mensaje de retroalimentación, el Product Owner podrá ver sus opiniones o sugerencias respecto al tema, como se muestra en la Figura 8.42.

Retroalimentación de la Reunión de Planeación del Sprint						
#	Fecha de envío	Proyecto	Mensaje	Emisor	Descripción	Opciones
1	26 de Mayo de 2024 a las 21:02	Aplicación móvil	Monitorización y gestión de procesos, así como para planificación y coordinación de estrategias y actividades clave para las mismas	empleado2	Después de revisar detenidamente el progreso y las funcionalidades implementadas hasta ahora, me gustaría proporcionarles unas sugerencia para mejorar este proyecto: Revisar el servicio de notificaciones y asegurar que esté correctamente integrado y optimizado. Considerar el uso de un servicio de terceros confiable si es necesario.	

Figura 8.42. Mensajes de retroalimentación para reunión de planeación del Sprint

El Product Owner dispondrá de una opción que le servirá para responder al mensaje de retroalimentación al asistente emisor, ver Figura 8.43.

Mensaje de Retroalimentación

Contestación

En general, estoy muy satisfecho con el progreso del proyecto y el estado actual de la aplicación. Estas sugerencias están destinadas a ayudarnos a perfeccionar la aplicación y garantizar que ofrezca la mejor experiencia posible a nuestros usuarios. Estoy seguro de que, con estas sugerencias, podremos llevar nuestro producto al siguiente nivel.

Enviar

Figura 8.43. Respuesta del Product Owner

Anexo E Preguntas de la encuesta aplicada al panel de expertos.

N°	Pregunta
1	Nombre de la empresa donde labora
2	País de origen de la empresa
3	Idioma con el que comunica en la empresa
4	Número de trabajadores en su empresa
5	Número de empleados dedicados al desarrollo de software, en su empresa
6	¿Cuál es su cargo actual en la empresa?
7	¿Utiliza el marco de trabajo Scrum?
8	¿Cuál es su rol actual en la empresa, según Scrum?
9	Años de experiencia en el desarrollo de software
10	¿En qué tipo de proyectos de software trabaja actualmente? (Seleccione las que apliquen, distribuidos y/o locales)?
11	¿Está familiarizado con el modelo de comunicación de Berlo o SMCR (Emisor, Mensaje, Canal y Receptor)?
12	¿En qué medida crees que el modelo de comunicación adaptado pueda mejorar la comunicación en tu equipo?
13	La información generada en cada una de las ceremonias se enviará a cada uno de los involucrados, a pesar de que no asistan a las reuniones por las diferencias de horario o alguna otra situación, ¿en qué medida cree que esto mejore la comunicación entre ellos?
14	La información generada en cada una de las ceremonias se enviará a cada uno de los involucrados en el idioma de cada uno de los destinatarios, ¿cree que esto mejore la comunicación entre ellos?
15	¿Cree que haya algún beneficio de llevar un control de los mensajes enviados, indicando si ya fue comprendido o no por parte de los destinatarios?
16	La implementación del modelo de comunicación adaptado, a través de una aplicación, ¿podría contribuir a que los mensajes transmitidos en las ceremonias Scrum lleguen de manera oportuna y completa a todos los participantes, optimizando la comunicación en su equipo?
17	¿La frecuencia de los mensajes enviados y recibidos puede promover una retroalimentación para mejorar la comunicación entre el personal de los equipos distribuidos?
18	¿Considera que la claridad de los mensajes se puede mejorar, una vez que se utilice el modelo de comunicación propuesto, en equipos distribuidos?
19	¿Considera que la comprensión de los mensajes puede mejorar, una vez que se utilice el modelo de comunicación propuesto?
20	¿Considera que la colaboración entre el equipo Scrum se puede mejorar, una vez que se utilice el modelo de comunicación propuesto, en equipos distribuidos?
21	¿Considera que el acceso a la información de forma oportuna se puede mejorar, una vez que se utilice el modelo de comunicación propuesto, en equipos distribuidos?
22	¿Considera que la retroalimentación se puede mejorar, una vez que se utilice el modelo de comunicación propuesto, en equipos distribuidos?
23	¿Considera que el uso de plantillas para el envío de mensajes y la retroalimentación se puede mejorar, una vez que se utilice el modelo de comunicación propuesto, en equipos distribuidos?
24	Los canales de comunicación utilizados, Comunica-Scrum, Correo Electrónico y una Aplicación de Mensajería Instantánea, ¿son adecuados para el tipo de información que se comparte?
25	¿Recomendarías el modelo de comunicación adaptado a otros equipos Scrum?
26	Por favor, comparta cualquier comentario o sugerencia que tenga sobre el modelo SMCR adaptado o su implementación en proyectos de software en un entorno GSD

En el siguiente enlace está el diseño del cuestionario en un formulario de Google Forms
<https://forms.gle/EsTYJVttMdKE55mD8>.

Anexo F Clases del diseño de Comunica-Scrum.

Las clases con atributos y métodos para la gestión de los proyectos Scrum se muestran en la Figura 8.44, no se muestran las relaciones para una mejor legibilidad.

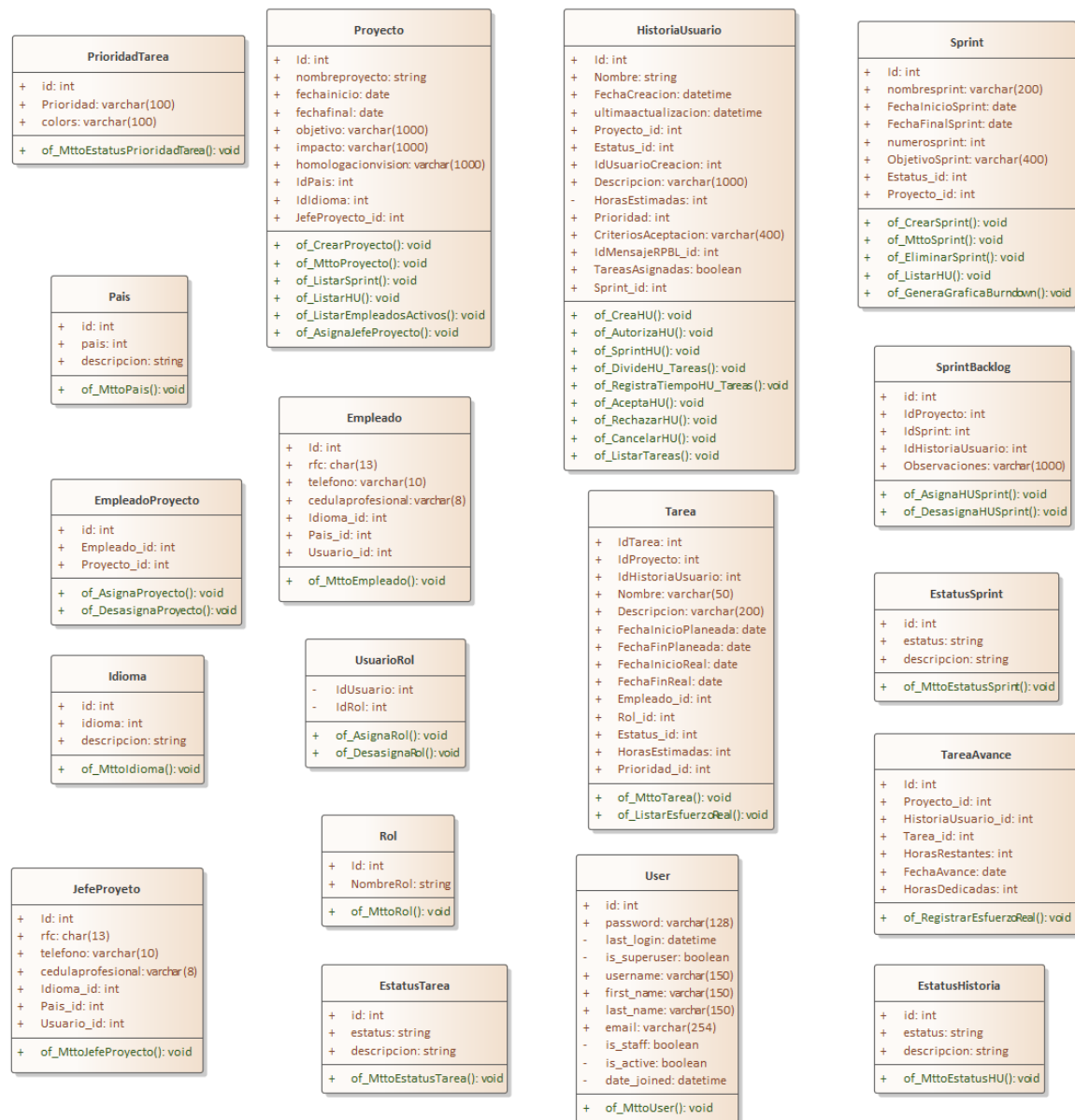


Figura 8.44. Clases para la gestión de proyectos con Scrum

Las clases con atributos y métodos para la comunicación entre emisor y receptor se muestran en la Figura 8.45, no se muestran las relaciones para una mejor legibilidad.

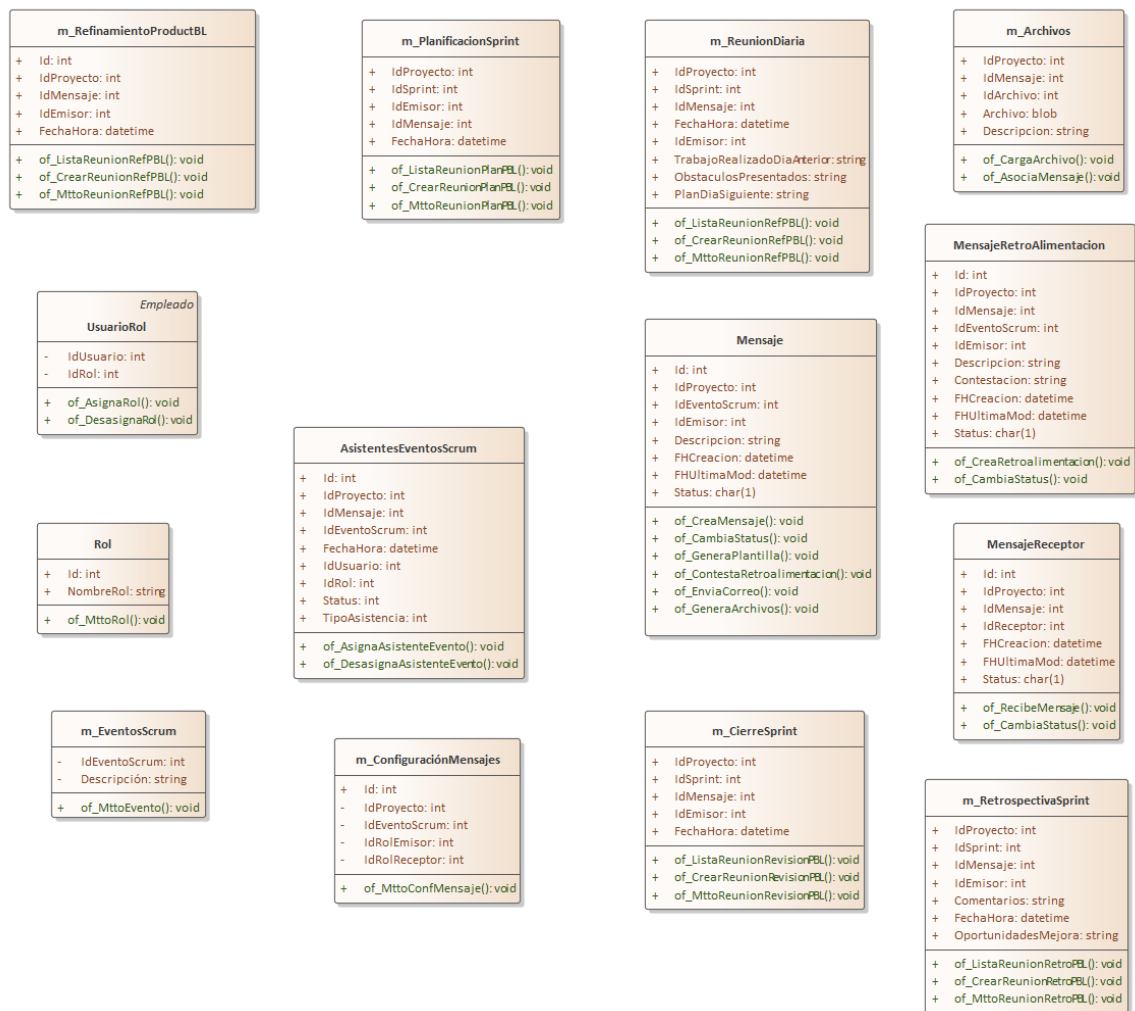


Figura 8.45. Clases que representan la comunicación entre emisor y receptor