

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

DESARROLLO DE UNA GUÍA METODOLÓGICA PARA MEJORAR LA DOCUMENTACIÓN DEL CAPÍTULO TÉCNICO DE LOS TRABAJOS DE GRADO DE LA CARRERA DE SOFTWARE QUE UTILICEN LA METODOLOGÍA SCRUM

Trabajo de grado previo a la obtención del título de Ingeniero de Software
presentado ante la ilustre Universidad Técnica del Norte.

Autor:

Ruth Alejandra Huertas Burgos

Director:

PhD. José Antonio Quiña Mera Msc.

Ibarra, 2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004347710		
APELLIDOS Y NOMBRES:	HUERTAS BURGOS RUTH ALEJANDRA		
DIRECCIÓN:	ALPARGATE Y RIO CHINCHIPE		
EMAIL:	alejahuertas1417@gmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0967849487

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UNA GUÍA METODOLÓGICA PARA MEJORAR LA DOCUMENTACIÓN DEL CAPÍTULO TÉCNICO DE LOS TRABAJOS DE GRADO DE LA CARRERA DE SOFTWARE QUE UTILICEN LA METODOLOGÍA SCRUM
AUTOR(ES):	HUERTAS BURGOS RUTH ALEJANDRA
FECHA:	07/03/2024
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA DE SOFTWARE
DIRECTOR:	PhD. ANTONIO QUIÑA MSc.
ASESOR 1:	Ing. MAURICIO REA MSc.
ASESOR 2:	PhD. CATHY GUEVARA MSc.

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 07 días del mes de marzo de 2024

EL AUTOR:



ESTUDIANTE

Ruth Alejandra Huertas Burgos

C.I 1004347710

CERTIFICACIÓN DIRECTOR

Ibarra 04 de marzo del 2024

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo Msc. Antonio Quiña Mera, certifico que la Srta. Ruth Alejandra Huertas Burgos portador de la cedula de ciudadanía número 1004347710, ha trabajado en el desarrollo del proyecto de grado **“Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de software que utilicen la metodología Scrum”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente

PhD. Antonio Quiña Mera Msc.

DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

El presente trabajo se lo dedico a mis padres Blanca Burgos y Guillermo Alejandro Huertas por haberme guiado y apoyado a lo largo de mi vida. Sin su amor, comprensión y aliento, este logro no habría sido posible y espero poder retribuir todo lo que me han brindado.

A mi hermana Ligia Huertas quien ha sido mi mayor apoyo en mi carrera universitaria, por su amor y preocupación que me brinda todos los días.

Ruth Alejandra Huertas Burgos

AGRADECIMIENTO

Agradezco a la Universidad Técnica del Norte por haber sido un segundo hogar donde pasé buenos momentos y supieron enseñarme grandes lecciones de vida. A los docentes de la carrera de software por su conocimiento, consejos y amistad compartidos en las aulas.

Agradezco a mi tutor de trabajo de titulación PhD. Antonio Quiña Mera MSc. Y a mis asesores Ing. Mauricio Rea y PhD. Cathy Guevara MSc. Por haberme brindado su apoyo y conocimiento para culminar mi carrera universitaria.

A los amigos que hice durante mi carrera universitaria, haciendo que sean buenos años junto a ellos.

Ruth Alejandra Huertas Burgos

TABLA DE CONTENIDOS

DEDICATORIA	3
AGRADECIMIENTO	4
ÍNDICE DE FIGURAS.....	8
ÍNDICE DE TABLAS	10
RESUMEN	11
ABSTRACT	12
INTRODUCCIÓN.....	13
Tema	13
Antecedentes.....	13
Situación Actual.....	13
Planteamiento del problema.....	13
Objetivos.....	14
Objetivo General.....	14
Objetivos Específicos	14
Alcance.....	15
Justificación.....	15
Justificación Metodológica	15
Justificación Técnica	16
CAPÍTULO 1	17
1. Marco Teórico.....	17
1.1. Conceptos Generales	17
1.1.1. Guía Metodológica.....	17
1.1.2. Documentación	17
1.1.3. Trabajo de Grado.....	17

1.2.	Guías de documentación.....	18
1.2.1.	Guías de práctica.....	18
1.2.2.	Guías de usuario.....	18
1.3.	Buenas prácticas en la documentación del software.....	18
1.3.1.	Requisitos	18
1.3.2.	Arquitectura/Diseño	18
1.3.3.	Técnico.....	19
1.3.4.	Usuario final	19
1.4.	Marco de trabajo Scrum	20
1.4.1.	Teoría de Scrum	20
1.4.2.	Fases de Scrum.....	21
1.4.3.	Equipo Scrum.....	22
1.4.4.	Eventos de Scrum.....	23
1.4.5.	Artefactos de Scrum	24
CAPÍTULO 2		26
2.	Desarrollo del proyecto	26
2.1.	Metodología para la guía metodológica	26
2.1.2	Objetivos, Alcance y Audiencia.....	27
2.1.3	Recopilación de la información	28
2.1.4	Aprobación	28
2.1.5	Edición y Diseño.....	28
2.1.6	Difusión.....	28
2.2.	Diseño de la guía metodológica	28
2.2.1	Pre-Juego	29
2.2.3	Juego	32
2.2.4	Post-Juego	33
2.3.	Desarrollo de la guía metodológica.....	33

2.3.1	Introducción	33
2.3.2	Objetivo.....	33
2.3.3	Alcance	34
2.3.4	Audiencia	34
2.3.5	Desarrollo del capítulo 3 “Marco Metodológico”	34
2.3.7	Desarrollo del capítulo 4 “Resultados y análisis”	34
2.4	Ejecución de fases Scrum	35
2.4.1	Fase 1. Pre-juego.....	35
2.4.2	Fase 2. Juego.....	43
2.4.3	Fase 3. Post-juego	50
CAPÍTULO 3		55
3.	Validación de resultados.....	55
3.1	Validación de la prueba de concepto	55
3.1.1	Fase 1. Pre-Juego.....	55
3.1.2	Fase 2. Juego.....	60
3.1.3.	Fase 3. Post-Juego	63
3.2	Resultados de la prueba de concepto	64
3.2.1	Análisis de resultados	65
3.2.2	Evaluación de los resultados	66
CONCLUSIONES		68
RECOMENDACIONES.....		69
BIBLIOGRAFÍA		70
ANEXOS		73

ÍNDICE DE FIGURAS

Figura 1 Árbol de problemas	14
Figura 2 Alcance del proyecto	15
Figura 3 Documentación arquitectónica y diseño	19
Figura 4 Marco Scrum.....	20
Figura 5 Ciclo de un Sprint	23
Figura 6 Proceso Scrum.....	24
Figura 7 Contenido del capítulo.....	26
Figura 8 Componentes de la guía metodológica	26
Figura 9 Vida del proyecto desde TI.....	30
Figura 10 Formato estructura del capítulo	35
Figura 11 Ejemplo de Arquitectura del proyecto	42
Figura 12 Diagrama de la base de datos.....	43
Figura 13 Ejemplo de acta de entrega.....	54
Figura 14 Roles del equipo de la prueba de concepto.....	55
Figura 15 Historia de usuario 1 de la prueba de concepto.....	56
Figura 16 Historia de usuario 2 de la prueba de concepto.....	56
Figura 17 Historia de usuario 3 de la prueba de concepto.....	57
Figura 18 Estimación de la prueba de concepto.....	58
Figura 19 Product backlog de la prueba de concepto.....	58
Figura 20 Sprint 0 de la prueba de concepto	58
Figura 21 Arquitectura del software de la prueba de concepto	59
Figura 22 Diagrama de la base de datos de la prueba de concepto	59
Figura 23 Planificación Sprint de la prueba de concepto	60

Figura 24 Sprint 1 Backlog de la prueba de concepto 61

Figura 25 Revisión Sprint 1 de la prueba de concepto 62

Figura 26 Retrospectiva del sprint 1 de la prueba de concepto 62

Figura 27 Prueba de aceptación de la prueba de concepto 63

Figura 28 Acta de entrega/recepción de la prueba de concepto 63

Figura 29 Niveles de calidad 67

Figura 30 Formato estructura del capítulo 79

Figura 31 Ejemplo de Arquitectura del proyecto 87

Figura 32 Ejemplo del diagrama de la base de datos 88

ÍNDICE DE TABLAS

Tabla 1 Diseño de la guía metodológica	29
Tabla 2 Formato de asignación de roles	35
Tabla 3 Ejemplo de asignación de roles	36
Tabla 4 Formato de Historia de Usuario	37
Tabla 5 Ejemplo de historia de usuario	38
Tabla 6 Formato de Product Backlog.....	39
Tabla 7 Ejemplo de Product Backlog	39
Tabla 8 Planificación Sprint 0	41
Tabla 9 Ejemplo de planificación Sprint 0	41
Tabla 10 Formato de Sprint Backlog.....	46
Tabla 11 Ejemplo Sprint Backlog.....	46
Tabla 12 Revisión de Sprint.....	47
Tabla 13 Ejemplo de Revisión de Sprint	48
Tabla 14 Formato de Retrospectiva del Sprint.....	49
Tabla 15 Formato de Pruebas de aceptación	50
Tabla 16 Formato de despliegue	51
Tabla 17 Ejemplo de formato de despliegue	51
Tabla 18 Acta de entrega/recepción	52
Tabla 20 Completitud Funcional	65
Tabla 21 Evaluación de la guía mediante la medida de completitud funcional	66

RESUMEN

Los trabajos de titulación permiten plasmar el conocimiento de cada estudiante universitario y es un trabajo que se debe realizar para obtener su título como ingeniero, debido a esto se presenta una guía metodológica que permita ayudar a que se finalice este trabajo, esta guía está centrada en el marco de trabajo Scrum por lo cual todo trabajo de titulación que utilice esta metodología podrá hacer uso de esta guía.

En primer lugar, se presenta todo el marco teórico, temas como el marco de trabajo de scrum, reglamento de los trabajos de titulación de la carrera, las buenas prácticas en la documentación en el software.

Luego se detalla el desarrollo del proyecto, la selección del diseño y metodología que tendrá la guía. Las fases que son empleadas en la guía metodológica (Pre-Juego, Juego y Post-Juego). En cada fase se encuentran elaboradas las plantillas que servirán a los estudiantes de trabajo de titulación para su documentación.

Finalmente, mediante una prueba de concepto se validó la guía metodológica en el cual se detallan la parte del análisis de resultados y la interpretación de resultados.

Palabras claves: Scrum, Guía metodológica, Documentación, Trabajos de titulación.

ABSTRACT

Thesis work allows each university student to express their knowledge and is a requirement to obtain their title as an engineer. Because of this, a methodological guide is presented to help students complete this work. This guide is centered around the Scrum framework, so any thesis work that uses this methodology can make use of this guide.

First, the entire theoretical framework is presented, covering topics such as the Scrum framework, regulations for thesis work in the program, and best practices for software documentation.

Next, the project development is detailed, including the selection of the design and methodology for the guide. The phases used in the methodological guide (Pre-Game, Game, and Post-Game) are outlined. Templates for thesis students to use in their documentation are provided for each phase.

Finally, a proof of concept was used to validate the methodological guide, detailing the analysis and interpretation of results.

Keywords: Scrum, Methodological Guide, Documentation, Thesis Work

INTRODUCCIÓN

Tema

Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de software que utilicen la metodología Scrum.

Antecedentes

La documentación es importante ya que permite que los clientes o usuarios finales conozcan cómo aprovechar al máximo un producto, además es parte de la relación con los clientes (Knott, 2020). Todos los productos de desarrollo de software necesitan de cierta documentación relacionada ya sea que desarrollen por equipos pequeños o gigantescas organizaciones. Asimismo, los errores que se presenten en la documentación pueden generar brechas dando como resultado una solución propuesta que no cumplirá con las expectativas de las partes involucradas, es por eso por lo que se debe prestar atención a la calidad de la documentación (altexsoft, 2023).

Uno de los mayores problemas con la documentación es no tener una idea clara de lo que se debe hacer, un alcance claro sin dudas sobre lo que debería incluirse o qué estilo adoptar (Procida, 2020). El repositorio de la Universidad Técnica del Norte cuenta con 504 trabajos de tesis publicados de la carrera de Ingeniería en Sistemas Computacionales y en los últimos 5 años el 30% de trabajos de titulación están realizados con el marco de trabajo Scrum

Situación Actual

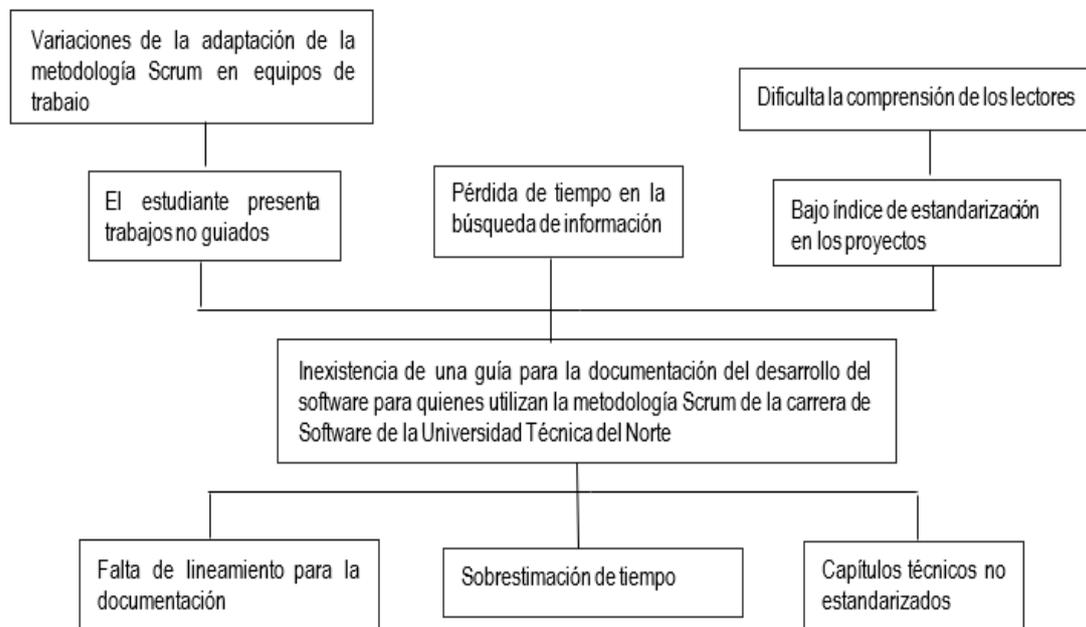
Los estudiantes de la Universidad Técnica del Norte de la carrera de Ingeniería en Sistemas Computacionales y Software deben realizar trabajo de grado, como requisito para poder culminar la carrera, además existe un alto porcentaje de trabajos de tesis que desarrollan software utilizando la metodología Scrum, donde presentan su trabajo de forma diferente, dado este escenario se plante desarrollar una guía para documentar el capítulo técnico, para que lleven a cabo de una manera guiada y estandarizada y facilitar el trabajo de redacción a los autores de la tesis.

Planteamiento del problema

La documentación aumenta el rendimiento de los ingenieros con respecto al análisis de impacto y comprensión del sistema (Satish & Anand, 2016). Un alto porcentaje de trabajos son realizados mediante metodologías ágiles una de ellas es Scrum, que se basa en el trabajo

de equipo y este enfoque ágil plantea que no se requiere una documentación completa, los valores del manifiesto ágil sugiere “software funcional sobre documentación completa” (altexsoft, 2023). Con el fin que la documentación del capítulo técnico de la tesis de grado de los estudiantes de la carrera de Sistemas computacionales y de Software sea realizada de una manera guiada y estandarizada para una fácil comprensión de los lectores. Cuando utilicen la metodología Scrum, mediante una guía para la documentación para los estudiantes de la carrera de Sistemas computacionales y Software de la Universidad Técnica del Norte, ver Figura 1.

Figura 1 Árbol de problemas



Objetivos

Objetivo General.

Elaborar una guía metodológica para documentar el capítulo técnico de los trabajos de grado que desarrollen software utilizando el marco de trabajo Scrum.

Objetivos Específicos

- Elaborar un marco conceptual y tecnológico.
- Elaborar una guía metodológica para documentación
- Validar la guía metodológica mediante una prueba de concepto.

Alcance

El proyecto propuesto tiene como finalidad desarrollar una guía para la documentación de trabajo de grado que desarrollen software para estudiantes de la carrera de Sistemas computacionales y Software de la Universidad Técnica del Norte y así despejar dudas y tener una idea clara de lo que se debe hacer. Las guías facilitan el desarrollo, la corrección y futuro mantenimiento del proyecto de una forma más sencilla (Alvarez, 2022). El objetivo de esta guía es ayudar al escritor a redactar de manera adecuada el capítulo técnico del trabajo de grado cuando realizan desarrollo de software usando el marco de trabajo Scrum. Se validará la guía mediante una prueba de concepto.

La estructura de la guía metodológica estará compuesta por:

- Planificación
- Recolección de información
- Marco teórico y tecnológico
- Análisis de los problemas hallados
- Validación de la guía metodológica

Figura 2 Alcance del proyecto



Justificación

Justificación Metodológica

El no contar con una guía para la documentación del capítulo técnico para los trabajos de grado que desarrollan software origina problemas a la hora de documentar el capítulo

técnico, de esta forma se plantea el desarrollo de la guía que permita a los estudiantes de la carrera de software mejorar la documentación para una fácil comprensión de los lectores.

El proyecto se enfoca en el objetivo 9 Industria, Innovación e Infraestructura de los 17 Objetivos de Desarrollo Sostenible, en el cual se menciona las siguientes metas:

Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países. (ODS, 2023)

Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo. (ODS, 2023)

El desarrollo de una guía metodológica permitirá que los estudiantes universitarios cuenten con material de apoyo para redactar su trabajo.

Justificación Técnica

La realización de la guía permitirá mejorar la productividad en la redacción del documento de trabajo de grado de los estudiantes que desarrollen software con el marco de trabajo Scrum.

CAPÍTULO 1

1. Marco Teórico

En este capítulo se determinó un marco de trabajo, el cual sirvió para realizar la fundamentación teórica de los aspectos para el Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de software que utilicen la metodología Scrum.

1.1. Conceptos Generales

1.1.1. Guía Metodológica

Es un documento técnico detallado de procesos, actividades o prácticas. Se describe diferentes operaciones o pasos en su secuencia lógica. Una guía metodológica debe basarse en una experiencia probada y debe incorporar las claves del éxito para su implementación (Meza, 2014).

1.1.2. Documentación

La documentación es una actividad que busca la mejora continua de la empresa, permite registrar todos sus pasos, de manera que las personas que tengan que realizarlos tengan la idea clara de cómo, cuándo y que hacer (Beneyto Carlos, 2022).

Documentar es un mostrar procesos, comparte el aprendizaje, lo socializa, lo formaliza y lo abre, obteniendo un proceso contrastable, mejorable, imitable y escalable. Una buena documentación debería convertir en replicables los pasos necesarios para llegar al resultado alcanzado. Documentar es todo un arte (Lafuente et al., 2018).

1.1.3. Trabajo de Grado

Es el resultado de una investigación personal acerca de un tema previamente acordado entre el estudiante y el director, debe representar el resultado de una investigación, además, el estudiante debe evidenciar su capacidad para investigar y reflexionar sobre un tema, para luego estructurarlo y presentarlo (Alonso, n.d.).

1.2. Guías de documentación

1.2.1. Guías de práctica

En el área de Salud, existen las llamadas guías de prácticas las cuales ayuda a los clínicos tomen una decisión más acertada sobre los pacientes individuales. La documentación en las guías de práctica clínica permite que los médicos sepan cómo reaccionar ante casos individuales de los pacientes y saber cómo suministrar adecuados tratamientos y medicamentos (Casariego & Costa, 2022).

1.2.2. Guías de usuario

Es la primera fuente de información necesaria a la que deben recurrir las personas que utilizan la aplicación desarrollada. Funciona como elemento ahorrador de varios aspectos como: avanzar de mejor manera en la adopción del desarrollo generado, dispersar el conocimiento más rápido y ahorrar tiempo (GEOScéntricos, 2019).

1.3. Buenas prácticas en la documentación del software

La documentación contribuye a la conservación a largo plazo y a la puesta a disposición de todos los recursos prácticos necesarios para los futuros trabajos de desarrollo. Una buena documentación es sólo parte una de las mejores prácticas para crear excelentes productos de software (Distillery, 2023).

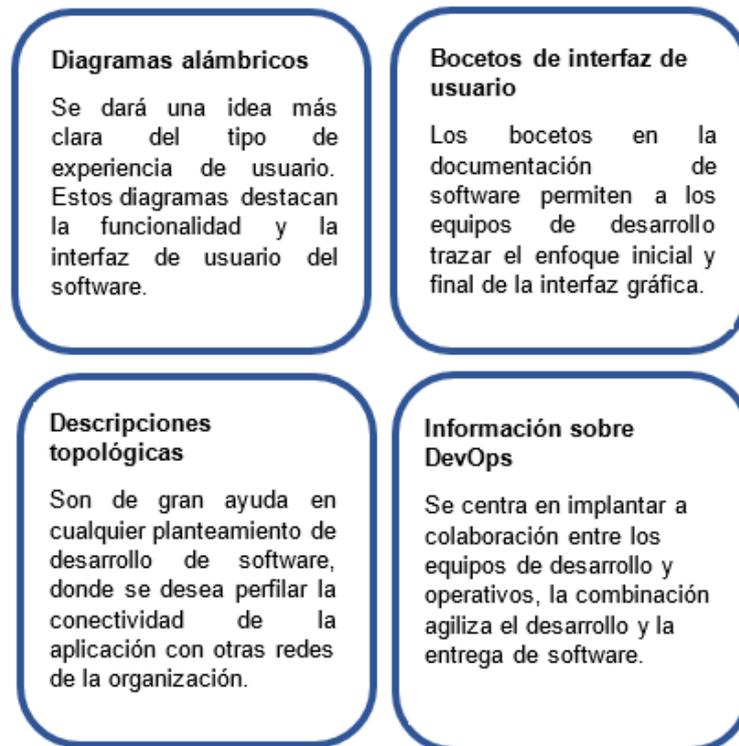
1.3.1. Requisitos

El inicio de un proyecto de desarrollo comienza con unos requisitos bien documentados, garantiza una cooperación eficaz, una comunicación clara y la comprensión precisa de cómo debe ser la tarea. La documentación de los requisitos debe permitir incluir cambios de forma rápida y sencilla durante el proceso de desarrollo del producto (Distillery, 2023).

1.3.2. Arquitectura/Diseño

Los documentos de arquitectura de software se usan para realizar un seguimiento de las decisiones de más alto nivel tomadas sobre la estructura y el comportamiento de sistema, además, deben hacer un seguimiento de cómo se aplican las soluciones. Se menciona algunos ejemplos de documentación arquitectónica y de diseño en la figura 3.

Figura 3 Documentación arquitectónica y diseño



Nota. Fuente (Distillery, 2023).

1.3.3. Técnico

La documentación técnica captura el código del programa, garantiza que las unidades de negocio asociadas establezcan los recursos adecuados para contribuir a que el producto de software alcance el logro completo. Proporciona información técnica y funcional que ayude a los usuarios a utilizar el producto (Bektas, 2023).

1.3.4. Usuario final

La documentación es un parte crucial del software, cuando se trata del usuario final la documentación suele ser manuales para los administradores del sistema y personal de apoyo (Bektas, 2023).

No tiene que ser una lectura agotadora al contrario mediante la creación de contenidos útiles y atractivos que se despliegan con el software, asegura que los usuarios finales tienen toda la información que les permite resolver por sí mismos los inconvenientes más comunes (Distillery, 2023).

1.4. Marco de trabajo Scrum

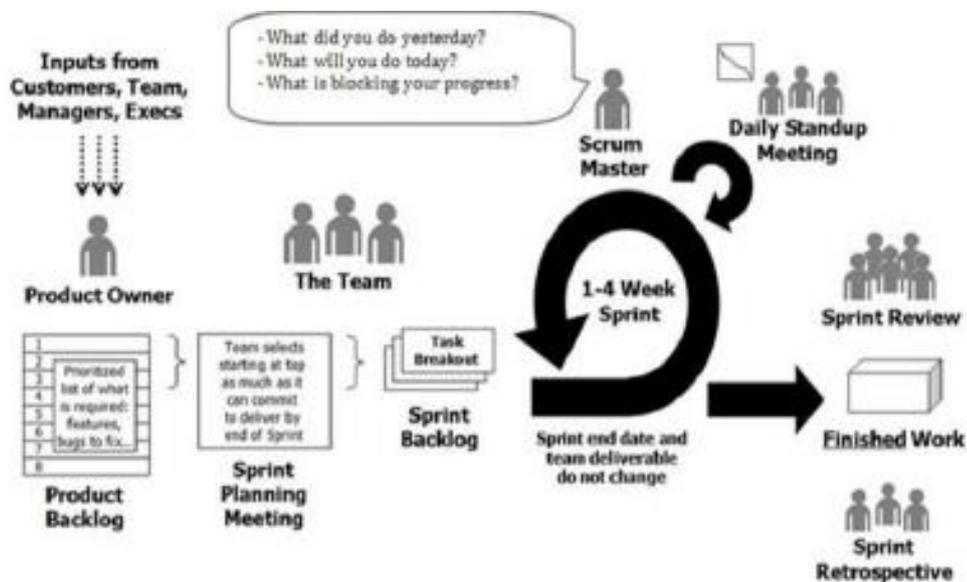
El marco de trabajo Scrum se caracteriza por agilizar a los equipos a desarrollar y cada vez más popular debido a su enfoque de promover una cultura centrada en el rendimiento y en generar valor (Rea-Peñafiel et al., 2023).

1.4.1. Teoría de Scrum

Scrum es un marco de trabajo que incentiva la colaboración en los equipos para lograr desarrollar productos complejos, reduce la complejidad en el desarrollo de productos para complacer las exigencias de los clientes (Humbachano, 2017).

Es un marco de trabajo ágil mediante el cual las personas pueden abordar problemas complejos y adaptativos, favoreciendo la entrega de productos con alto valor. Fomenta la colaboración entre equipos para alcanzar un trabajo de alto impacto (Martins, 2023).

Figura 4 Marco Scrum



Nota. Fue!

Scrum implementa un enfoque iterativo e incremental para mejorar la previsibilidad y mitigar el riesgo, combina cuatro eventos formales para la inspección y adaptación dentro de un evento el Sprint. Dichos eventos funcionan porque implementar los pilares empíricos de Scrum:

Transparencia. El proceso y el trabajo deben ser visibles para aquellos que hacen el trabajo, como para los que recién el mismo. Las decisiones importantes se basan en el estado percibido de sus tres artefactos formales. La transparencia permite la inspección, evaluar sin claridad implica engaños y pérdidas innecesarias (Schwaber & Sutherland, 2020).

Inspección. Deben ser inspeccionados los artefactos de Scrum y el progreso encaminados a los objetivos pactados con regularidad y diligentemente para que se pueda detectar varianzas o problemas no deseados. La inspección permite la adaptación (Schwaber & Sutherland, 2020).

Adaptación. Si algún aspecto de un proceso se desvía fuera de los límites aceptables o si el resultado es inaceptable, se debe ajustar el proceso o los materiales lo más antes posible para reducir la desviación adicional. Se desea que un equipo de Scrum se adapte en el momento en que aprenda algo nuevo por medio de la inspección (Schwaber & Sutherland, 2020).

1.4.2. Fases de Scrum

Están agrupadas en tres fases denominadas Pre-juego, Juego y Post-juego.

Pre-juego. Se define la visión del proyecto, que puede no estar clara inicialmente pero que puede perfeccionarse en sprints posteriores. Además, se prepara un plan que incluye la estimación de tiempos y costos con la fecha de entrega del producto final y el número de lanzamientos en los que se entregará el producto final. Algunas tareas importantes completadas en esta fase incluyen la evaluación de riesgos, la definición del equipo de desarrollo, la validación de las herramientas de desarrollo y la verificación de la aprobación y los recursos (Anwer et al., 2017).

Juego. Es la fase de desarrollo el cual se realiza pequeñas iteraciones llamadas sprints, esto es un período de desarrollo con un límite de tiempo que varía entre una semana a cuatro semanas dependiendo del nivel de complejidad y el riesgo. Estos sprints añaden actividades tales como desarrollar, finalizar, revisar y ajustar (Anwer et al., 2017).

Post-juego. Es la fase final luego de implementar las funciones deseadas durante la fase de desarrollo, se produce el despliegue final, se realizan pruebas de integración finales, manuales de usuario y se desarrolla materiales de capacitación para la versión definitiva (Anwer et al., 2017).

1.4.3. Equipo Scrum

El equipo Scrum es lo bastante pequeño para permanecer ágil y lo suficientemente grande para completar un trabajo significativo dentro de un sprint, por lo general 10 o menos personas. Todo el equipo de Scrum es responsable de crear un incremento valioso y útil en cada sprint. Se define tres responsabilidades dentro del equipo: los desarrolladores, el propietario del producto y el Scrum Master (Schwaber & Sutherland, 2020).

Desarrolladores

(Schwaber & Sutherland, 2020) asegura que son las personas encargadas de crear cualquier aspecto de un incremento útil en cada sprint. Son responsables de:

- Crear un plan para el Sprint, Sprint Backlog;
- Garantizar la calidad mediante el cumplimiento de una definición de Hecho;
- Ajustar su plan diariamente hacia el Objetivo Sprint;
- Fomentar la responsabilidad entre los miembros del equipo como profesionales

Propietario del producto (Product Owner)

La organización debe respetar sus decisiones, siendo visibles en el contenido y el orden del trabajo pendiente del producto y a través del incremento que se inspecciona en la revisión de Sprint (Schwaber & Sutherland, 2020).

El dueño del producto es una persona responsable de las ganancias o pérdidas del producto. Debe contar con una buena habilidad de comunicación debido a que son importantes para tratar con diferentes partes interesadas que tienen diferentes intereses (Anwer et al., 2017).

Scrum Master

Son líderes que ayudan al equipo Scrum y a toda la organización, es el encargado de establecer tal como se define en la Guía de Scrum, facilitando a todos a comprender la teoría y la práctica de Scrum, dentro y fuera de la organización (Schwaber & Sutherland, 2020).

Es un facilitador de equipo, se asegura de que los miembros del equipo sigan las prácticas, reglas y valores de Scrum para conseguir valor comercial. Es responsable de proteger el equipo de la intervención externa y brinda circunstancias favorables para trabajar (Anwer et al., 2017).

1.4.4. Eventos de Scrum

Scrum se basa en sprints que tienen una duración determinada la cual el equipo desarrolla el producto de acuerdo con el trabajo pendiente. En la figura 4 se observa las siguientes actividades se realizan durante el sprint:

Figura 5 Ciclo de un Sprint



Nota. Fuente (Hadida & Troilo, 2020).

El objetivo de estos es la transparencia, para proporcionar la oportunidad de inspeccionar o adaptar el producto.

Sprint Planning

Se basa en dos fases, en la primera fase se reúne el equipo de desarrollo para determinar de manera realista lo que se puede desarrollar la tarea hasta el final del sprint. El equipo de desarrollo debe comprometerse a completar el trabajo teniendo presente el objetivo del sprint, la cual afecta la funcionalidad y el uso de la tecnología (Gonçalves, 2018).

La segunda fase el dueño del producto analiza el objetivo de todos los miembros del equipo que desean lograr terminado el sprint, así como los elementos del product backlog que encaminan hacia ahí (Gonçalves, 2018).

Daily Scrum

Los miembros del equipo realizan una rápida reunión de 15 minutos diarios para revisar el progreso del proyecto, cada miembro del equipo responderá tres preguntas (Lee & Chen, 2023);

- ¿Qué se ha hecho desde la última reunión?
- ¿Qué se hará antes de la próxima reunión?
- ¿Cuáles son los inconvenientes en el proceso?

Esta reunión permite a los miembros del equipo reconocer la causa de cualquier interrupción y tomar las medidas correctivas correspondientes (Anwer et al., 2017).

Sprint Development

Se lleva a cabo las actividades de diseño, desarrollo y pruebas para cada tarea del sprint backlog. Estas tareas se implementan dependiendo de la prioridad definida por el dueño del producto (Anwer et al., 2017).

Sprint Review

Finalizado el sprint se realiza una revisión del producto realizado, dura aproximadamente cuatro horas si fuera el caso de un sprint de un mes. Además de revisar el producto, se revisa el trabajo que queda de anteriores sprint reviews y lo que aún se quiere para finalizar el sprint actual dentro del plazo definido (Gonçalves, 2018).

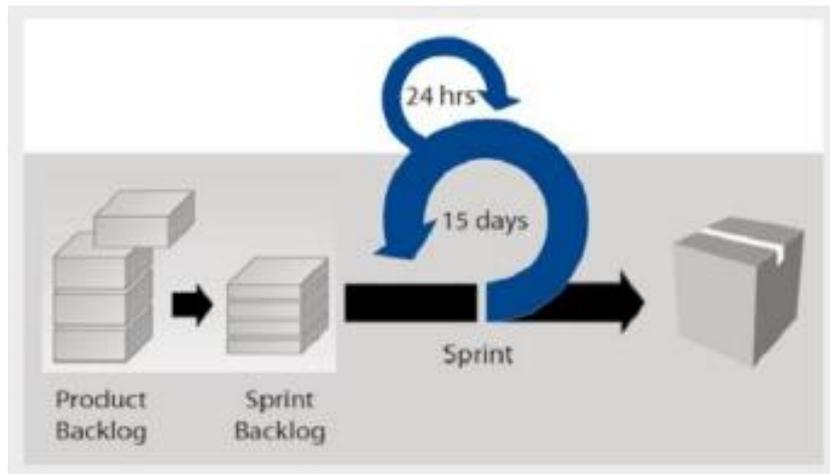
Sprint Retrospective

Es una reunión que dura dependiendo del sprint, se da cada vez que se completa la revisión del sprint. Esta retrospectiva permite al equipo la posibilidad de inspeccionar lo que se ha hecho y planificar un plan para mejorar sus procedimientos de trabajo para el próximo sprint (Gonçalves, 2018).

1.4.5. Artefactos de Scrum

Estos artefactos están diseñados para maximizar la transparencia de la información clave. Representan trabajo o valor. La figura 6 representa el proceso del marco Scrum.

Figura 6 *Proceso Scrum*



Nota. Fuente (Gonçalves, 2018).

Product Backlog

Los elementos de trabajo pendiente del producto, que pueden ser incluidos por el equipo de Scrum durante un Sprint, son considerados aptos para su inclusión en un evento de planificación de Sprint. Este nivel de claridad comúnmente se alcanza después de llevar a cabo las actividades de refinamiento (Schwaber & Sutherland, 2020).

Sprint Backlog

El trabajo pendiente de Sprint se compone del objetivo sprint (¿Por qué?), el conjunto de elementos de trabajo pendiente de producto seleccionados para el Sprint (¿Qué?), además de un plan práctico para proporcionar el incremento (¿Cómo?) (Schwaber & Sutherland, 2020).

Increment

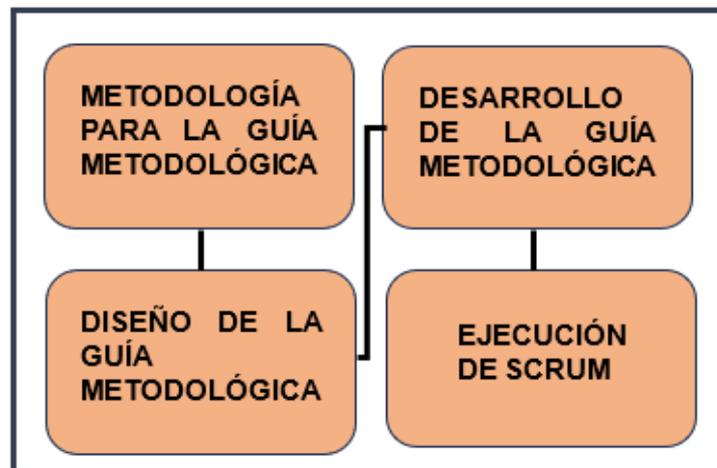
Es un paso de hormigón encaminado al objetivo del producto, cada incremento es aditivo a todos los anteriores incrementos y verificado a fondo, asegurando que todos los incrementos funcionen en conjunto. El incremento es utilizable cuando se proporciona valor (Schwaber & Sutherland, 2020).

CAPÍTULO 2

2. Desarrollo del proyecto

En este capítulo se introduce un marco de trabajo para el desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico que utilicen la metodología Scrum. La figura 7 representa la estructura del capítulo.

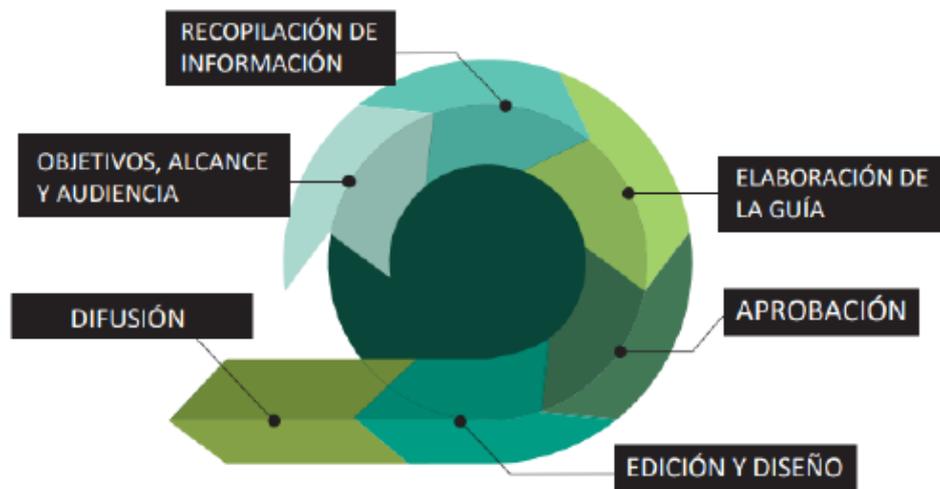
Figura 7 Contenido del capítulo



2.1. Metodología para la guía metodológica

El presente tema de investigación tiene por objetivo realizar una guía metodológica para documentar el capítulo técnico de los trabajos de grado con el marco de trabajo Scrum, para lo cual se planteó un diseño con el que contará la guía tomando en cuenta los trabajos detallados anteriormente. En la Figura 8 se muestra el diseño para realizar la guía.

Figura 8 Componentes de la guía metodológica



Nota. Fuente (Robles Belmonte, 2017).

A continuación, se describe las seis etapas propuestas por (Robles Belmonte, 2017) para el desarrollo de una guía:

2.1.2 Objetivos, Alcance y Audiencia

En la primera etapa se debe dar respuesta a las siguientes preguntas:

Objetivo: ¿Qué queremos conseguir con la guía?

Es importante especificar el objetivo principal que se persigue, ejemplo: documentar el capítulo técnico de los trabajos de grado que desarrollen software con el marco de trabajo Scrum (Robles Belmonte, 2017).

Audiencia: ¿Cuál es la audiencia objetivo?

Es esencial identificar a quién va dirigida la guía y que se espera de ellos. Ejemplo: de manera que el receptor pueda solventar problemas relacionado al desarrollo de software (Robles Belmonte, 2017).

Alcance: ¿Qué conocimiento queremos sistematizar y diseminar?

Es necesario reconocer a alto nivel los contenidos y mensajes clave que se va a incluir en la guía metodológica, por ejemplo: correcta documentación en el desarrollo de software haciendo uso de la metodología Scrum (Robles Belmonte, 2017).

2.1.3 Recopilación de la información

La segunda etapa corresponde a la recopilación y al análisis del material que se empleará para la realización de la guía metodológica. (Robles Belmonte, 2017) enlista las posibles fuentes de información:

- Documentación del proyecto: informes de seguimiento de proyectos, informes de evaluación, líneas de base realizadas, información del sistema de seguimiento y evaluación del proyecto.
- Documentos relevantes sobre la temática del proyecto de fuentes externas, que puedan aportar información adicional.
- Documentos realizados anteriormente para el proyecto.

2.1.4 Aprobación

La guía metodológica requerirá la aprobación de la organización contratante, puede sugerir cambios en cuanto a los elementos de la guía, esta etapa puede llegar a repetirse hasta alcanzar cierta aceptación (Robles Belmonte, 2017).

2.1.5 Edición y Diseño

Finalmente aprobado la guía se procede con la revisión de posibles errores gramaticales y ortográficos, además de añadir un diseño según lo que el líder del proyecto considere (Robles Belmonte, 2017).

2.1.6 Difusión

La última fase del desarrollo de la guía consiste en garantizar que la guía llegue a su público objetivo. Por lo que es importante seleccionar y utilizar los canales de comunicación como: publicaciones en revistas científicas, talleres prácticos, entre otros (Robles Belmonte, 2017).

2.2. Diseño de la guía metodológica

La guía metodológica para mejorar la documentación está planteada de acuerdo con las fases del marco de trabajo Scrum. La tabla 1 indica el diseño establecido para el desarrollo de esta guía.

Tabla 1 Diseño de la guía metodológica

FASE	ACTIVIDAD SCRUM	TAREA SCRUM	Producto
PRE-JUEGO	Planteamiento	Definición de roles	Involucrados
		Definición de requisitos	Historias de Usuario
		Product Backlog	Lista priorizada
JUEGO	Sprint	Sprint Planning	Arquitectura
		Daily Scrum	No Aplica
		Revisión de Sprint	Lista de Observaciones
		Retrospectiva de Sprint	Lista de mejoras
POST-JUEGO	Entrega	Pruebas de integración/aceptación	Artefactos de validación
		Despliegue	No Aplica
		Acta entrega/recepción	Documento firmado

La aplicación de SCRUM como marco de trabajo se centra en la realización de la guía metodológica, por lo tanto, las actividades llevadas a cabo para la gestión del proceso de desarrollo de software fueron distribuidas en base a las fases descritas de SCRUM Pre-juego, Juego y Post-juego, que se describen a continuación:

2.2.1 Pre-Juego

Esta etapa da a conocer las actividades entre las que se listan: la visión, el presupuesto, el sprint backlog inicial, además de la arquitectura y diseños generales (Rivadeneira, 2012).

Se identifican más requerimientos y priorizan tareas para la primera iteración. Entre las actividades se tiene: planificación, diseño y prototipos (Rivadeneira, 2012).

Es la fase inicial del proyecto, inicia con la detección del problema a solucionar. Ante a un desafío, se procede a reunir a un equipo de individuos competentes para abordar y resolver la necesidad identificada. Este equipo debe tener asignado roles entre ellos, de acuerdo como define el marco de trabajo Scrum.

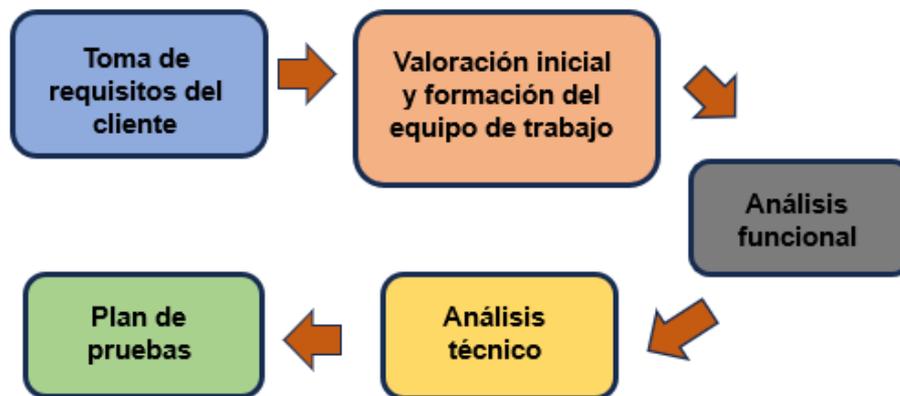
Dicho esto, una vez asignado los roles, el Product Owner será el encargado de obtener requisitos del usuario, la unión de varias listas de usuarios conocido como el Product Backlog dentro de Scrum.

Todos los requerimientos del Product Backlog deben ser comunicados a cada uno de los miembros del equipo, entre todos los miembros deben etiquetar o clasificar los requerimientos.

Las personas encargadas de asignar los pesos de dificultad correspondientes a cada requerimiento son los desarrolladores. Luego se procede a elaborar el Sprint, esto está conformado por algunos requerimientos que tiene un objetivo común, el Sprint va encaminado a cumplir un objetivo y estos deben ser notificados a cada miembro del equipo.

(Rodríguez, 2018) describe mediante fases para entregar la respectiva documentación luego de haber concluido el proyecto, lo denomina el ciclo de vida del desarrollo desde el punto de vista como jefe de proyecto de IT y de un desarrollador, lo cual se indica en la Figura 10.

Figura 9 Vida del proyecto desde TI



Nota. Fuente (Rodríguez, 2018).

Toma de requisitos del cliente: Es la parte donde se especifica como cliente lo que se desea del producto. La documentación que se debe presentar al cliente va a depender de lo que se especifica lo que quiere.

- Cláusula de responsabilidad/confidencialidad
- Gestión de calidad

- Visión actual del proyecto
- Requisitos Globales

Nota: No necesariamente se debe cumplir con todas, pero se debe detallar lo que el cliente demande de alguna forma.

Valoración inicial del proyecto y formación del equipo de trabajo: Se debe indicar el tiempo que se tardaría en el desarrollo y el coste aproximadamente se llevaría a cabo indicando que se tiene una idea global del proyecto.

- **Equipo Scrum**

El equipo debe tener especificado sus roles, además, indicar la experiencia que posee cada uno. Estos roles se asignan a todas las personas que intervienen en el proyecto. Este trabajo está orientado a trabajos de titulación.

Análisis funcional: Es una descripción de cómo funciona la aplicación siguiendo paso a paso lo que el usuario debe hacer para conseguir el funcionamiento total de la aplicación.

Análisis técnico: Se expone el modelo de programación que se va a realizar, detallando cada componente para que sea validado por este. Este tipo de documentación es esencial debido a que si la empresa contratante desearía mejorar la aplicación contratando a diferentes desarrolladores deben seguir el mismo orden y la nomenclatura del modelo.

- **Definición de requisitos**

Se utiliza historias de usuario de acuerdo con la metodología Scrum, (Angel et al., 2022) que brinda una detallada explicación sobre las historias de usuario. La plantilla de historias de usuario es una adaptación de Scrum Manager (Scrum Manager Bok, 2023).

Es una descripción de una funcionalidad que debe incorporar un sistema de software, la cual aporta valor al cliente. La estructura de una historia de usuario está formada por:

- Nombre breve y descriptivo
- Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que se desea realizar
- Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita (Scrum Manager Bok, 2023).

Plan de pruebas: El objetivo de realizar el plan de pruebas es de indicar el correcto funcionamiento de la aplicación de pruebas demostrando que funciona como se debe, tal como el cliente solicitó.

- **Product Backlog**

Basándose en las historias de usuario establecidas anteriormente, el Product Owner las prioriza y asigna un orden para el desarrollo del producto de software.

2.2.3 Juego

Se implementa un sistema terminado para entregar en una serie de iteraciones también denominado sprint. Es la fase dos del marco de trabajo Scrum, se organiza los product backlog, se procede a crear los Sprint, los cuales tienen objetivos y es asignado a un desarrollador como tarea.

A partir de los Product Backlogs, se procede a crear el Sprint Backlog. El equipo de trabajo debe asignar el tiempo de desarrollo que tiene cada sprint, el tiempo va a depender del grado de dificultad de cada sprint.

Terminado con el Sprint Backlog, se procede con la asignación de tareas a los desarrolladores hecho por el Scrum Master, esto se asigna dependiendo del nivel de experiencia en desarrollar software. Asignados la tarea, se realiza hasta el tiempo que se estableció al Sprint.

En el proceso de desarrollo pueden manifestarse dudas, por lo que el encargado de gestionar las dudas es el Scrum Master, para gestionar las inquietudes debe organizar los Daily Scrum. Los Daily Scrum se debe realizar con el Team Scrum, para aclarar las dudas de todos los desarrolladores, estas reuniones tienen un tiempo de 15 minutos.

Una vez finalizado el tiempo de realizar el Sprint designado a los desarrolladores, el Scrum Master realiza una reunión con el objetivo de revisar las actividades realizadas y verificar si logró cumplir con el objetivo del Scrum. Las personas involucradas en la revisión son el Scrum Master, Developers y el Product Owner.

Cuando se cumplen los objetivos del Sprint se cierra, en caso de no dar cumplimiento a los objetivos se reasigna el Sprint hasta la próxima revisión, esta revisión tiene una duración de 4 horas aproximadamente.

Como finalización de la fase del Juego se realiza una retrospectiva de los Sprint que fueron asignados, en este punto los desarrolladores comunican los inconvenientes que se presentaron al realizar el Sprint. Esta reunión de retrospectiva tiene una duración de 4 horas.

2.2.4 Post-Juego

En esta última fase es necesario realizar una reunión en el que estén presentes el Scrum Master y los Developers, con la finalidad de obtener la información de los incrementos que se ha obtenido al desarrollar el producto.

El Scrum Master debe registrar los incrementos del producto, deben ser revisados por el Product Owner en conjunto con el cliente con el objetivo de saber si se ha cumplido con los requisitos y objetivos establecidos.

Si los incrementos no cumplen con las expectativas, se retrocede a la primera fase y luego realizar las demás fases hasta alcanzar los objetivos del proyecto. En caso de que todos los incrementos estén correctos se procede a la entrega del producto final.

2.3. Desarrollo de la guía metodológica

2.3.1 Introducción

La presente guía metodológica tiene como objetivo ofrecer a los estudiantes de la carrera de ingeniería de software una propuesta para documentar el capítulo metodológico y el capítulo de resultados y análisis de los Trabajos de Integración Curricular (TRIC) enfocados en los proyectos que realicen desarrollo de software con el marco de trabajo Scrum. Este documento está fundamentado en los lineamientos de la (GUÍA OPERATIVA DE LA UNIDAD DE INTEGRACIÓN CURRICULAR, 2023) y la estructura de las fases fundamentales para desarrollo de proyectos propuestas por el marco de trabajo Scrum (Anwer et al., 2017). En cada fase se explica las actividades clave y los entregables esperados, con el objetivo de proporcionar una visión clara y concisa de cómo aplicar Scrum en el desarrollo del TRIC.

2.3.2 Objetivo

Tiene como objetivo principal proporcionar a los estudiantes de la carrera de ingeniería de software una adecuada estructuración y organización del contenido del marco metodológico. Con esto se pretende que la guía sea una herramienta práctica y útil para los estudiantes de la carrera de software, que les permita mejorar la calidad y eficacia de la documentación de sus Trabajos de Integración Curricular.

2.3.3 Alcance

Realizar una guía metodológica para los estudiantes de Trabajo de Integración Curricular que desarrollen software que utilicen el marco de trabajo Scrum.

2.3.4 Audiencia

Enfocada en el sector educativo, los estudiantes que están cursando la unidad de trabajo de titulación.

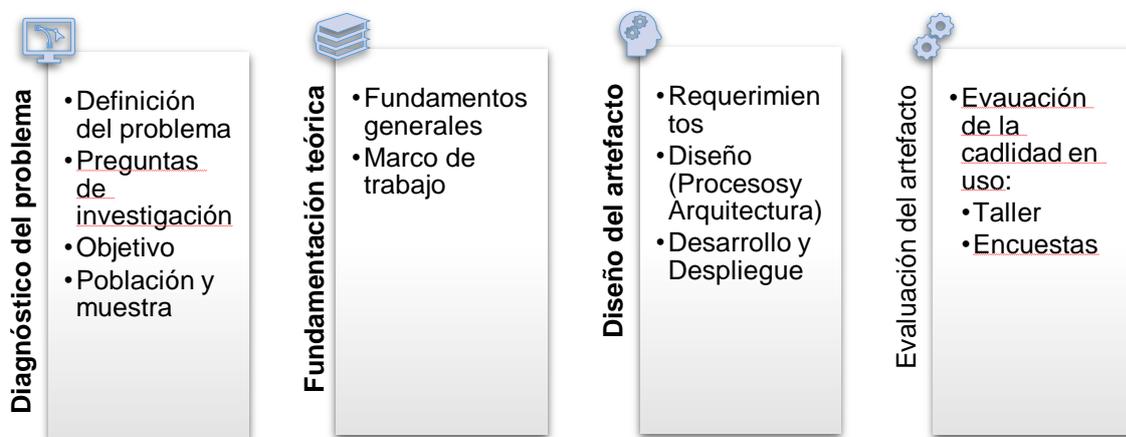
2.3.5 Desarrollo del capítulo 3 “Marco Metodológico”

Enfoques y tipos de investigación

[El enfoque y tipos de investigación lo definirán cada director y el tesista del trabajo de integración curricular]

Diseño de la Investigación

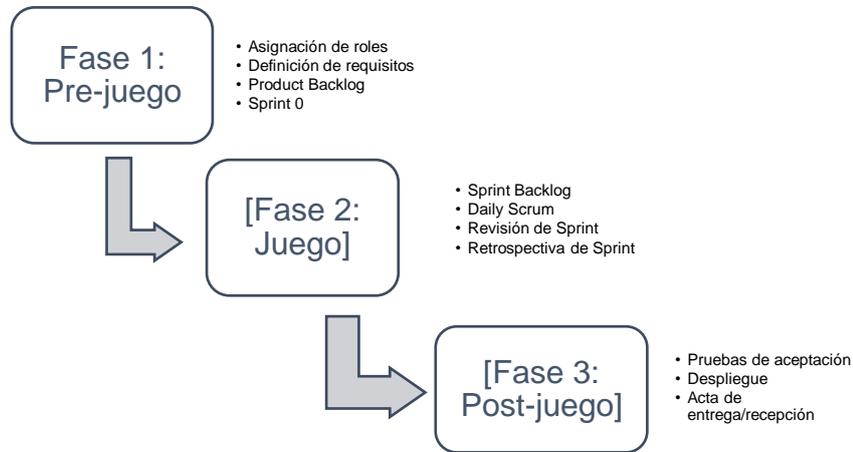
El diseño de la investigación se basa en las fases iniciales del enfoque Design Science Research (DSR)



2.3.7 Desarrollo del capítulo 4 “Resultados y análisis”

En este capítulo se describe el desarrollo del proyecto [[Nombre del proyecto de desarrollo de software](#)] basándose en el marco de trabajo Scrum. La Figura 10 muestra estructura del capítulo, y sus principales componentes.

Figura 10 Formato estructura del capítulo



NOTA: Esta sección es una introducción del capítulo que muestra la estructura, los temas y subtemas a tratar, para que el lector pueda tener una visión general del capítulo.

2.4 Ejecución de fases Scrum

2.4.1 Fase 1. Pre-juego

Esta fase de pre-juego también se conoce como la fase de preparación, donde se establece varias actividades clave para iniciar un nuevo Sprint. Las actividades entre las que se listan son: la visión, el presupuesto, el sprint backlog inicial, además de la arquitectura y diseños generales (Rivadeneira, 2012).

Asignación de roles

Es en esta sección se asigna los roles Scrum a los integrantes del proyecto [[Nombre del proyecto de desarrollo de software](#)]. A continuación, en la Tabla 2 se muestra los roles y una descripción corta del cargo e institución del equipo de trabajo.

Tabla 2 Formato de asignación de roles

Nombre ¹	Rol ²	Cargo ³
[Nombre del Tutor]	Product Owner	Director de Trabajo de Integración Curricular (Docente UTN)
[Nombre de Tesista]	Scrum Master	Estudiante de la carrera de software UTN
[Nombre de Tesista(s)]	Team Development	Estudiante(s) de la carrera de software UTN
[Nombre del Asesor] y/o [Nombre de interesado(s)]	Stakeholders	Descripción corta del cargo e institución de los interesados

La Tabla 3 muestra el formato de asignación de roles con todos los campos completados.

Tabla 3 Ejemplo de asignación de roles

Nombre	Rol	Cargo
PhD. Antonio Quiña MSc.	Product Owner	Director de Trabajo de Integración Curricular (Docente UTN)
Ruth Alejandra Huertas Burgos	Scrum Master	Estudiante de la carrera de software UTN
Ruth Alejandra Huertas Burgos	Team Development	Estudiante de la carrera de software UTN
PhD. Cathy Guevara MSc. Ing. Mauricio Rea MSc.	Stakeholders	Asesores del TRIC (Docente UTN)

¹ **Nombre:** El nombre completo del integrante del proyecto.

² **Rol:** Se asigna en base al marco de trabajo Scrum.

³ **Cargo:** Descripción corta del cargo o institución a la que pertenece.

NOTA: En el marco de trabajo de Scrum, la primera tarea es designar roles al equipo de trabajo involucrado en el proyecto, y tomando en cuenta el reglamento (Resolución Nro. 001-073-CEAACES-2013-13, 2013) de la UTN, los involucrados en el trabajo de integración curricular son: un tutor, un asesor, y uno o dos tesisistas (podrán ser dos estudiantes siempre y cuando sea justificado). Por lo tanto, se tomará en cuenta los siguientes roles de Scrum: Product Owner, Scrum Master, Team Development y Stakeholders (Beck, 2008).

Definición de requisitos

La definición de requisitos se realiza mediante Historias de Usuario (HU), las cuales, son una técnica ágil que captura las necesidades del software desde la perspectiva del usuario final. Cada historia describe brevemente una funcionalidad o característica deseada, centrada en el valor que aporta al usuario. A continuación en la Tabla 4, se detalla las historias de usuario que describen los requisitos funcionales específicos para el desarrollo del proyecto [Nombre del proyecto de desarrollo de software].

Tabla 4 Formato de Historia de Usuario

Historia de usuario		
ID⁴:	[Código HU-001]	Usuario⁵: [Nombre del usuario de la historia de usuario]
Nombre⁶:	[Asignación de un nombre a la HU]	
Prioridad⁷:	Dependencia⁸:	Estimación⁹:
[Baja, Media, Alta]	[Identificadores de otras HU]	[Número de estimación requerida para implementar la funcionalidad]
Descripción¹⁰:	Como un [usuario] quiero [descripción de la funcionalidad] con la finalidad de [descripción de razón o resultado].	
Criterios de Aceptación¹¹:		

⁴ **ID:** Identificador único de la historia de usuario.

⁵ **Usuario:** Nombre del usuario que se asigna esa HU.

⁶ **Nombre:** Debe ser corto y claro.

⁷ **Prioridad:** Determina el nivel de importante a implementarse.

⁸ **Dependencia:** En caso de que exista relación.

⁹ **Estimación:** Esfuerzo requerido para implementar la funcionalidad descrita en la historia de usuario.

¹⁰ **Descripción:** Síntesis de la historia de usuario.

¹¹ **Criterios de aceptación:** Una lista de condiciones que deben cumplirse para que la historia de usuario se considere completada.

[Valida que la funcionalidad implementada cumple con las expectativas del usuario y del equipo de desarrollo]

A continuación, en la Tabla 5 se indica una Historia de Usuario.

Tabla 5 Ejemplo de historia de usuario

Historia de usuario			
ID:	HU-001	Usuario:	Cliente
Nombre:	Creación de formatos		
Prioridad:	Dependencia:	Estimación:	
Media	NO	7	
Descripción:	Como un cliente quiero utilizar la estructura de los formatos con la finalidad de obtener una correcta documentación		
Criterios de Aceptación:			
<ul style="list-style-type: none">• Las plantillas deben poder adaptarse al proyecto• Cada campo debe ser comprensible y claro.			

NOTA: En las historias de usuario es fundamental escribir en un lenguaje claro y comprensible para todos los miembros del equipo, evitando tecnicismos innecesarios. Se debe considerar la descripción clara y concisa de la funcionalidad, la comprensión universal y la estimación de esfuerzo.

Además de los campos que se consideran necesarios se puede añadir otros campos adicionales según el tipo de proyecto, pueden ser aconsejable campos como:

Título: Descriptivo de la historia de usuario.

Observación: Para aclarar la información o cualquier uso que pueda ser de utilidad (Menzinsky et al., 2016).

Product Backlog

Con una definición precisa de los requisitos, se procede a crear el product backlog, que es esencialmente una lista priorizada de las funcionalidades que darán vida al producto. Los elementos más relevantes se sitúan en la parte superior de la lista (Le, 2017). La Tabla 6 indica el product backlog.

Tabla 6 Formato de Product Backlog

Nro. Orden¹²	ID Historia de Usuario¹³	Descripción¹⁴	Estimación¹⁵
[Ordenar según el nivel de importancia]	[Identificador de la historia de usuario]	[Descripción corta de la historia de usuario]	[Estimación de la historia de usuario]
	[Código- HU]	[Crear vistas]	
	[Código- HU]	[Crear CRUD]	
	[Código- HU]	[Obtener consulta]	
...

Se muestra a continuación la Tabla 7 utilizando el formato de Product Backlog a modo de ejemplo con los campos completados.

Tabla 7 Ejemplo de Product Backlog

Nro. Orden	ID Historia de Usuario	Descripción	Estimación
1	HU-001	Crear formatos	7
2	HU-002	Crear listas	7
3	HU-003	Crear vistas	5
4	HU-004	Crear vistas	5

¹² **Nro. Orden:** Indica el orden que están las historias de usuario.

¹³ **ID Historia de usuario:** Identificador de la historia de usuario.

¹⁴ **Descripción:** Una descripción corta de la historia de usuario.

¹⁵ **Estimación:** Tiempo estimado real para cumplir con esa tarea.

...

...

...

NOTA: El Product Backlog se debe mantener actualizado y revisar periódicamente para ajustar las prioridades según las necesidades del proyecto. es importante que las historias de usuario sean claras, concisas y estén bien definidas, de manera que el equipo de desarrollo pueda entenderlas fácilmente.

(Cohn, 2010) indica que se debe realizar el product backlog de acuerdo con el acrónimo DEEP para resumir los atributos clave de un buen product backlog, se describen a continuación:

- **Detallar apropiadamente:** Las historias de usuario que se encuentran en las primeras posiciones del product backlog y que se abordarán pronto deben estar lo suficientemente claras como para poder completarlas durante el próximo sprint. Por otro lado, las historias que no se abordarán en el corto plazo deben estar descritas con menos detalle.
- **Estimado:** El trabajo pendiente es una lista de todo el trabajo por realizar. Es una útil herramienta de planificación, debido a que los elementos que se encuentran más abajo en el trabajo pendiente no se entienden completamente, las estimaciones asociadas con ellos serán menos precisas que las estimaciones dadas a los elementos en la parte superior.
- **Emergente:** Tomar en cuenta que el product backlog no es estática, cambia con el tiempo, se agregará eliminarán o se cambiarán las prioridades de las historias de usuarios sobre el trabajo pendiente del producto.
- **Prioriza:** El product backlog debe ordenarse con los elementos más valiosos en la parte superior y lo menos valiosos en la parte inferior. Al trabajar siempre se debe hacer en orden de prioridad, el equipo puede maximizar el valor del producto o sistema que se está desarrollando.

Sprint 0

Durante una fase inicial llamada Sprint 0, se llevó a cabo la planificación del proyecto de desarrollo de software. En este período se definió la arquitectura tecnológica, se estableció un esquema inicial para la base de datos y se elaboró un diagrama de procesos del sistema. La Tabla 8 indica las tareas llevadas en este Sprint.

Tabla 8 Planificación Sprint 0

Asistentes ¹⁶ : [Indicar los roles de los asistentes]		
Objetivo ¹⁷ : [Hay que describir que se realizará en este Sprint]		
Duración ¹⁸	Fase de desarrollo ¹⁹	Tarea ²⁰
2	[Diseño]	[Diseñar el diagrama tecnológico...]
2	[Planificación]	[Esquema inicial de la base de datos...]
3	[Revisión]	...

La Tabla 9 se indica el ejemplo de la planificación del Sprint 0.

Tabla 9 Ejemplo de planificación Sprint 0

Asistentes : Product Owner, Scrum Master, Team Development.		
Objetivo : Definir una arquitectura tecnológica		
Duración	Fase de desarrollo	Tarea
3	Diseño	Diseñar el diagrama tecnológico
2	Diseño	Esquema inicial de la base de datos
...

¹⁶ **Asistentes**: Las personas que estuvieron presentes en la planificación

¹⁷ **Objetivo**: Que se desea cumplir en esa planificación

¹⁸ **Duración**: Tiempo que tomará en realizar esa tarea

¹⁹ **Fase de desarrollo**: Conjunto de etapas que contienen el proceso de creación del producto

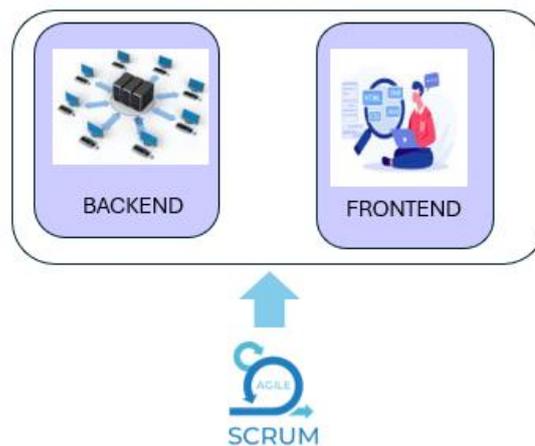
²⁰ **Tarea**: Nombre de la tarea a realizar

NOTA: A diferencia de los sprints regulares en los que se entregan incrementos de producto funcionales, en el Sprint 0 no se espera entregar un incremento de producto completo y potencialmente entregable.

Arquitectura del proyecto

La Figura 11 muestra la arquitectura de la aplicación las tecnologías que se utilizarán para el desarrollo de la aplicación. Para el manejo de datos se utilizará la base de datos [Nombre del gestor de base de datos], las herramientas tecnológicas que se utilizarán son: [Nombre de las herramientas tecnológicas a usar].

Figura 11 Ejemplo de Arquitectura del proyecto



NOTA: La arquitectura tecnológica es fundamental para el diseño y desarrollo de sistemas de software eficientes, seguros y escalables, y debe ser cuidadosamente planificada y gestionada para garantizar el éxito del proyecto.

(Arizbé, 2023) Indica que la arquitectura de un aplicativo puede incluir:

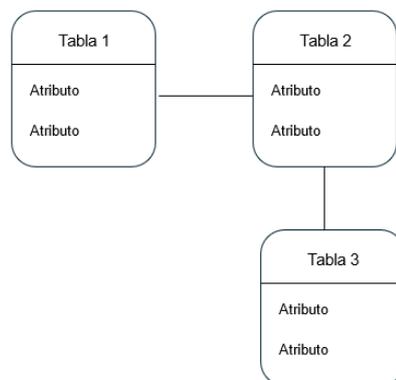
- **Componentes:** Son módulos, servicios o partes del software que realizan tareas específicas dentro del sistema.

- **Conexiones:** Describen como estos componentes se comunican entre sí. Implica interfaces, protocolos de comunicación y flujos de datos que permitan una interacción efectiva entre los diferentes elementos del sistema.
- **Patrones de diseño:** Son soluciones establecidas y recurrentes para problemas comunes en el diseño de software. Estos patrones brindan una guía probada para estructurar el software de manera eficiente, ayudando a mejorar la calidad y mantenibilidad de código.
- **Estilo arquitectónico:** es el enfoque de alto nivel que define la estructura general del sistema. Algunos ejemplos de estilos arquitectónicos incluyen la arquitectura de tres capas, la arquitectura orientada a servicios, la arquitectura de microservicios, entre otros.

Diagrama de la base de datos

En esta sección se utilizó como gestor de base de datos [[Nombre del gestor de base de datos](#)], la Figura 12 se detalla las tablas con los respectivos atributos y entidades.

Figura 12 Diagrama de la base de datos



NOTA: Los diagramas de base de datos pueden ser de varios tipos, como diagramas entidad-relación (ER), diagramas de modelo relacional, y diagramas de modelo de objetos.

2.4.2 Fase 2. Juego

En esta fase del juego existen una serie de iteraciones conocidas como sprint, aprender a entregar software en cada sprint es uno de los mayores desafíos, la intención es

que en cada sprint se entregue algo de valor inmediato a los usuarios o clientes. Se debe asegurar de terminar un sprint y estar preparado para el siguiente, la importancia de establecer y cumplir con un objetivo para el sprint y la necesidad de valorar cuánto tiempo tiene el equipo para alcanzar un objetivo (Cohn, 2010).

Entregar software funcional en cada sprint, es tan importante que uno de los cuatro valores dados en el manifiesto ágil establece que se debe valorar “el software funcional ante la documentación ágil”. Se enfatiza el software funcional por tres razones clave:

- **Software funcional fomenta la retroalimentación:** El equipo de trabajo puede recopilar más y mejores comentarios si proporciona un producto funcional pero parcial a los usuarios, al contrario de proporcionar un documento sobre lo que servirá el producto. Existirá una interacción, brindarán mejores comentarios al poder ver y tocar el producto.
- **Software funcional ayuda a medir el progreso:** Software que funciona y que requiere la entrega de una parte del valor del usuario en cada sprint, evita que no se tenga una idea de cuanto queda por hacer.
- **Software funcional permite entregar antes si así lo desea:** Entrega el producto antes puede ser muy valioso, en un mundo competitivo y cambiante (Cohn, 2010).

Dedica un poco de esfuerzo en cada sprint preparándote para el siguiente sprint. Evitar los sprints de actividades específicas, el equipo decide utilizar un sprint para análisis y diseño, un segundo sprint para codificación y un tercero para pruebas, pero con ello trae tres desventajas:

- **Existe un riesgo de cronograma:** planificar cuanto trabajo se puede realizar en un sprint será más propenso a errores porque el esfuerzo depende en gran medida de la calidad del trabajo realizado en el sprint anterior.
- **Mas tiempo para pasar de la idea a la función probada y en ejecución:** Prolonga el tiempo necesario para recibir comentarios de los clientes, usuarios u otras personas.
- **No resuelve el problema de la superposición de trabajos:** Cuando todo el trabajo se realiza en un sprint, todo el equipo se mueve al mismo ritmo. Cuando se introduce sprints de actividades específicas permitimos que los diferentes subequipos progresen a diferentes ritmos, hace que se acumule trabajo frente a algunos subequipos (Cohn, 2010).

Durante el desarrollo iterativo e incremental del proyecto es necesario mantener horarios regulares y estrictos. Los beneficios de los sprints manteniendo una longitud de tiempo fija:

- **Los equipos de trabajo se benefician de un ritmo regular:** Cuando se tiene una cadencia regular de entre una y cuatro semanas ayuda a los equipos a adaptarse al ritmo de trabajo que más les convenga.
- **La planificación de sprint se vuelve más fácil:** Tanto la planificación de sprints como de lanzamientos se simplifica cuando los equipos mantienen una duración de sprint constante.
- **La planificación de lanzamiento se vuelve más fácil:** Los equipos Scrum derivan sus planes de lanzamientos empíricamente, estiman el tamaño del trabajo a realizar en un proyecto y luego miden la cantidad completado por sprint.
- **Aprovechar energía:** Elegir la longitud al inicio de cada sprint es un desperdicio de energía, experimente con un par de veces, tome una decisión y manténgala firme hasta que exista una razón importante para cambiar.

Al enfatizar que todos los sprint deben tener la misma duración, no quiere decir que deba ser una regla inquebrantable, se debe elegir un día de la semana que funcione bien en tu entorno y comenzar todos los sprints ese día.

Un error común es extender un sprint, mantener los sprints estrictamente limitados en el tiempo refuerza la idea de avanzar continuamente el proyecto, cada cierto tiempo el equipo debe entregar algún incremento de producto nuevo potencialmente comercializable. Si se permite que los plazos varíen o se extienden ocasionalmente, esta valiosa disciplina se pierde (Cohn, 2010).

Sprint backlog

Es un elemento para visualizar el trabajo del sprint y está gestionado por el equipo de desarrolladores. El objetivo es mantener la transparencia dentro del desarrollo. Todo el trabajo que el equipo de desarrolladores haya seleccionado para hacer durante el siguiente sprint pasa al sprint backlog (Calzadilla et al., 2018). La Tabla 10 se detalla los elementos de la plantilla del Sprint Backlog.

Tabla 10 Formato de Sprint Backlog

Fecha ²¹ :		[día, mes, año]	
Asistentes ²² :		[Indicar los roles de los asistentes]	
Historia de Usuario ²³	Fase de desarrollo ²⁴	Tarea ²⁵	Estimación ²⁶
[Código HU-001]	[Análisis]	[Revisar documentación]	1
	[Pruebas]	[Realizar pruebas de petición]	2
	[Desarrollo]	[Crear función de obtener token]	3
[Código HU-002]	[Análisis]
	[Pruebas]
	[Desarrollo]
...
TOTAL HORAS			[X]

A modo de ejemplo se presenta la Tabla 10 del formato del sprint backlog.

Tabla 11 Ejemplo Sprint Backlog

Fecha:		05 de marzo del 2024	
Asistentes:		Product Owner, Scrum Master, Team Development	
Historia de Usuario	Fase de desarrollo	Tarea	Estimación
HU-001	Análisis	Revisar documentación del marco de trabajo Scrum	4
	Pruebas	Realizar pruebas de petición en el sistema	2
	Desarrollo	Crear función de obtener token	3
HU-002	Análisis
	Pruebas

²¹ **Fecha:** Especificar el día, mes y año que se realizó el Sprint Backlog.

²² **Asistentes:** Las personas que estuvieron presentes en la planificación.

²³ **Historia de Usuario:** Identificador de la historia de usuario.

²⁴ **Fase de desarrollo:** Conjunto de etapas que contienen el proceso de creación del producto.

²⁵ **Tarea:** Nombre de la tarea a realizar.

²⁶ **Estimación:** Tiempo estimado (horas) asignada para cada tarea.

	Desarrollo
...
TOTAL HORAS			9

NOTA: El Sprint Backlog debe actualizarse continuamente a lo largo del sprint, a medida que se completan tareas, surgen nuevos requisitos o se identifican cambios en las prioridades.

Daily Scrum

El presente trabajo está orientado a los trabajos de titulación y las personas involucradas son mínimas por lo que se omite esta sección, debido a que no se puede realizar las reuniones diarias cuando el equipo está conformado por una o dos personas y no se elaboró una plantilla.

Revisión de Sprint

En esta sección se debe detalla los incrementos hechos para la entrega del Sprint, se evidencia con imágenes que demuestran el avance del producto, el trabajo completado durante el sprint, mostrando las funcionalidades implementadas y cómo cumplen con los criterios de aceptación La Tabla 12 se plasma lo realizado.

Tabla 12 Revisión de Sprint

Fecha ²⁷ :	[día, mes, año]
Asistentes ²⁸ :	[Indicar los roles de los asistentes]
Descripción ²⁹	[Detallar lo que se realizó]
	[Colocar una imagen del incremento realizado]

²⁷ **Fecha:** Especificar el día, mes y año que se realizó la revisión del Sprint.

²⁸ **Asistentes:** Las personas que estuvieron presentes en la revisión.

²⁹ **Descripción:** Una descripción corta y clara de lo que se realizó.



En la Tabla 13 se presenta un ejemplo utilizando el formato mencionado de revisión.

Tabla 13 Ejemplo de Revisión de Sprint

Fecha³⁰:	05 de marzo del 2024
Asistentes³¹:	Product Owner, Scrum Master, Team Development
Descripción³²	Diseño de una guía metodológica en base al marco de trabajo Scrum

FASE	ACTIVIDAD SCRUM	TAREA SCRUM	Producto
PRE-JUEGO	Planteamiento	Definición de roles	Involucrados
		Definición de requisitos	Historias de Usuario
		Product Backlog	Lista priorizada
JUEGO	Sprint	Sprint Planning	Arquitectura
		Daily Scrum	No Aplica
		Revisión de Sprint	Lista de Observaciones
		Retrospectiva de Sprint	Lista de mejoras
POST-JUEGO	Entrega	Pruebas de integración/aceptación	Artefactos de validación
		Despliegue	No Aplica
		Acta entrega/recepción	Documento firmado

³⁰ **Fecha:** Especificar el día, mes y año que se realizó la revisión del Sprint.

³¹ **Asistentes:** Las personas que estuvieron presentes en la revisión.

³² **Descripción:** Una descripción corta y clara de lo que se realizó.

NOTA: La revisión de sprint va más allá de la demostración del incremento del producto y revisar el estado actual del proyecto, si no que sirve como un nuevo punto de referencia para tomar decisiones sobre el futuro y establecer una estrategia de negocio (Roche, 2019).

Se realiza al finalizar cada sprint para indicar el incremento del software desarrollado. El equipo de desarrollo hace una demostración sobre dicho incremento del software y resume las funcionalidades que han sido desarrolladas y entregadas en ese sprint, la reunión de revisión da mucho valor para el proyecto porque se obtiene una retroalimentación clara y directa acerca del software (Quesada, 2020).

Retrospectiva de Sprint

Es una reunión que tiene como objetivo reflexionar sobre el trabajo y el resultado del último sprint. Se realiza luego de la revisión del sprint y antes de la planificación del siguiente sprint. El resultado de la reunión es la creación de un plan de mejora a implantar tan pronto como sea posible, esta actividad es una de las más importantes en el proceso ágil (Quesada, 2020). La Tabla 14 se indica lo tratado en esta reunión.

Tabla 14 Formato de Retrospectiva del Sprint

Fecha:	[día, mes, año]
Asistentes:	[Indicar los roles de los asistentes]
Título	Descripción
¿Qué salió bien del sprint? ³³	[Se terminó la creación de las vistas]
¿Qué no salió como se esperaba? ³⁴	[Existen varias formas de presentar, lo que llevó a investigar más acerca...]
¿Qué mejoras se implementará? ³⁵	[Se prepara crear los archivos...]

³³ ¿Qué salió bien del sprint?: Aspectos positivos del sprint, logros alcanzados, tareas que salieron bien.

³⁴ ¿Qué no salió como se esperaba?: Obstáculos que dificultaron el trabajo.

³⁵ ¿Qué mejoras se implementará?: Posibles acciones para mejorar el siguiente sprint.

NOTA: La retrospectiva del sprint es una parte clave del marco de trabajo Scrum, ya que ayuda al equipo a adaptarse y mejorar continuamente, lo que contribuye a la eficacia y la satisfacción en el trabajo.

2.4.3 Fase 3. Post-juego

En esta fase todas las actividades se dan por terminado y finalmente se cierra el proyecto, realizando la liberación del producto, la generación de documentación final (León, 2022). Entre las tareas de cierre se encuentran: integración, pruebas del sistema, documentación de usuario (Scrum Manager Bok, 2024).

Pruebas de aceptación

Se acepta la funcionalidad de la historia de usuario especificada en la fase de la definición de los requisitos. Todas estas deben cumplir para dar paso a la totalidad del proyecto. Las pruebas de aceptación suelen realizarse al final del proceso de desarrollo, antes de que el sistema sea puesto en producción, y son una parte importante del proceso de garantía de calidad del software. A continuación, se muestra la Tabla 15.

Tabla 15 Formato de Pruebas de aceptación

ID HU	Nombre	Funcionalidad	Aceptación	
			SI	NO
[Identificador de la historia de usuario]	[Obtener consulta]	[Obtener la consulta de forma sencilla mediante...]	X	
[Código HU–0001]	[Crear vistas]	[Visualizar las vistas de...]	X	
[H-XX-03]	[Crear CRUD]	[Realizar las acciones...]	X	
[H-XX-04]	[Obtener consulta]	[Obtener consular de forma sencilla mediante...]	X	
....

NOTA: Las pruebas de aceptación se deben realizar a medida que se va produciendo el desarrollo de los sprint y cualquier error o incidencia se podrá detectar de manera temprana, para planificar su resolución en los siguientes sprints (González et al., 2014)

Son aquellas en las que los usuarios o expertos en el negocio validan los requisitos que se definieron al inicio del proyecto. Las reuniones que se realicen son de seguimiento de presentación y aceptación del sprint, se analizan los casos de uso cubiertos en el sprint, además de los aspectos relativos a la funcionalidad cubierta (González et al., 2014).

Despliegue

La fase de lanzamiento se da una vez que el producto esté totalmente terminado y es momento de entregar al cliente. Esto debe realizarse de manera clara y documentada, asegurándose que se cumplan todas las expectativas acordadas (Torres, 2023). Incluye la configuración de servidores, la transferencia de archivos, la configuración de base de datos, la implementación de medidas de seguridad y la realización de pruebas finales para asegurar que la aplicación funcione correctamente. Ver Tabla 16.

Tabla 16 Formato de despliegue

Recursos de Despliegue		
Sistema Operativo	Especificar el sistema operativo con el que se trabajó [Windows, Fedora]	
Servidor de despliegue	Especificar en que servidor se encuentra [Heroku, Vercel]	
Enlace de la aplicación desplegada	[Dirección para acceder a la aplicación] [Detallar los recursos utilizados]	
Máquina	Almacenamiento	
	CPU	...
	Memoria	...
Observaciones	En caso de que exista alguna novedad	

A modo de ejemplo se presenta la Tabla 17 del formato de despliegue con sus respectivos campos.

Tabla 17 Ejemplo de formato de despliegue

Recursos de Despliegue	
Sistema Operativo	Windows 10
Servidor de despliegue	Vercel
Enlace de la aplicación desplegada	https://aplicacionweb.vercel.app/

	Almacenamiento	512 SSD
Máquina	CPU	@ 2.00GHz 2.60 GHz
	Memoria	16 RAM
	Observaciones	Ninguna

NOTA: Al desplegar una aplicación, es importante planificar cuidadosamente el proceso, realizar pruebas exhaustivas y asegurarse de tener medidas adecuadas de seguridad y gestión de errores para garantizar un despliegue exitoso.

Acta entrega/recepción

Una vez concluido el trabajo se realiza a entregar por parte el Product Owner del proyecto mediante un acta de entrega recepción. En el proceso de entrega del producto los principales involucrados son los siguientes:

- **Responsable de entrega:** Se refiere al responsable del área de TIC de una entidad o dependencia universitaria y quién deberá realizar la entrega de los elementos técnicos y de información.
- **Responsable de recepción:** Se denomina como a la nueva persona que tomará el cargo de la persona responsable del área de TIC de la entidad o dependencia.
- **Autoridad administrativa:** Hace referencia a la figura administrativa encargada de gestionar las bajas de personal y el control de los bienes y servicios generales de la entidad o dependencia

Tabla 18 Acta de entrega/recepción



ACTA DE ENTREGA RECEPCIÓN DE SOFTWARE

Fecha de entrega:
[dd/mm/aaaa]

<Fecha en la cual se hace la entrega del requerimiento>

Autor:

[Nombre del ingeniero desarrollador]

Versión: [Número de a versión del software]

Lugar: [Ciudad en la que se realizó la entrega]

Proyecto/Sistema de información:

[Nombre del proyecto o sistema de información]

Se realiza la entrega por parte del Sr./Srta. [Nombres completos del estudiante] con C.I. [Cédula de Identidad] del trabajo “[Nombre del desarrollo del sistema]” para el área de [Nombre de entidad u organización], desarrollado de acuerdo con lo que se estableció y que puedan dar uso de este.

Productos entregados:

- [Proyecto de desarrollo (código fuente)]
- [Manual técnico y de usuario]
- Pruebas de funcionalidad
- Capacitación al funcionario encargado

RECIBÍ C ONFORME

[Nombre del responsable
que recibe]

Cargo

ENTREGUÉ CONFORME

[Nombre del responsable
que recibe]

Cargo

A continuación se presenta en la Figura 13, un ejemplo del modelo de acta de entrega con los campos completos.

Figura 13 Ejemplo de acta de entrega

 <p>UNIVERSIDAD TÉCNICA DEL NORTE</p> <p>FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS</p>		<h2>ACTA DE ENTREGA RECEPCIÓN DE SOFTWARE</h2>	
Fecha de entrega: 07/03/2024		Autor: Ruth Alejandra Huertas Burgos	
		Versión: aplicacionweb v2.0	
		Lugar: Ibarra, UTN	
Proyecto/Sistema de información: Guía metodológica para documentar el capítulo técnico que utilicen el marco trabajo Scrum			
<p>Se realiza la entrega por parte del Sr./Srta. Ruth Alejandra Huertas Burgos con C.I. 1004347710 del trabajo "Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de la carrera de software que utilicen la metodología Scrum" para el PhD. Antonio Quiña MSc. desarrollado de acuerdo con lo que se estableció y que puedan dar uso de este.</p> <p>Productos entregados:</p> <ul style="list-style-type: none"> • Manual técnico y de usuario • Pruebas de funcionalidad • Capacitación al funcionario encargado 			
<hr/> <p>RECIBÍ CONFORME</p> <p>PhD. Antonio Quiña Mera MSc.</p> <p>Docente Investigador</p>		<hr/> <p>ENTREGUÉ CONFORME</p> <p>Ruth Alejandra Huertas Burgos</p> <p>Desarrollador</p>	

CAPÍTULO 3

3. Validación de resultados

3.1 Validación de la prueba de concepto

Para la validación de la guía metodológica se seleccionó un trabajo de titulación que se encontraba en proceso de documentación del capítulo técnico y utilizó el marco de trabajo Scrum. El tema de trabajo de titulación seleccionado es “Desarrollo de una aplicación web para mejorar la redacción de artículos de divulgación científica en los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte”.

Para asegurar su efectividad y calidad se realizó la validación de la guía metodológica por un experto en el marco de trabajo Scrum el PhD Antonio Quiña Mera Msc, realizando recomendaciones, sugerencias y mejoras basadas en su experiencia y conocimiento del tema, lo que permitió fortalecer la guía metodológica.

3.1.1 Fase 1. Pre-Juego

El tesista de ese trabajo aplicó la guía metodológica para documentar el capítulo técnico y siguió cada una de las fases descritas en la guía. En la primera fase de Pre-juego se indicó que se debería detallar los roles del equipo, por lo cual en la Figura 13 se encuentra los roles del equipo.

Roles de equipo

Figura 14 Roles del equipo de la prueba de concepto

Rol	Nombre		Descripción
Product Owner	MSc. Arciniega	Silvia	Directora de Tesis
Scrum Máster	Sr. Arellano	Jonathan	Tesista
Scrum Team	Sr. Arellano	Jonathan	Tesista

Nota. Fuente (Arellano, 2024).

Historias de usuario

Luego de describir los roles del equipo se detalla las historias usadas que describen los requisitos funcionales de la aplicación web. Se muestran algunas historias de usuario para evidenciar el uso de las plantillas de la guía metodológica.

Figura 15 Historia de usuario 1 de la prueba de concepto

Historia de Usuario	
Código: HU-01	Usuario: Estudiante
Nombre: Registro de usuarios	
Prioridad: Alta	Estimación (días): S
Descripción:	
Como estudiante nuevo, deseo registrarme en la aplicación para acceder a información y actividades de redacción, gramática y ortografía.	
Criterios de aceptación:	
<ul style="list-style-type: none">• La aplicación debe contar con una vista de registro.• Solo se pueden registrar usuarios con correo institucional.• Contraseña con al menos 8 caracteres• Los campos de contraseña deben ser visibles opcionalmente.• Confirmación de registro por correo electrónico.	

Nota. Fuente (Arellano, 2024).

Figura 16 Historia de usuario 2 de la prueba de concepto

Historia de Usuario	
Código: HU-02	Usuario: Estudiante
Nombre: Inicio de sesión	
Prioridad: Alta	Estimación (días): S
Descripción:	
Como estudiante, quiero iniciar sesión para completar actividades y acceder a recursos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Vista de inicio de sesión. • Validación de campos (correo y contraseña). • Restricción de acceso a usuarios no registrados. • Sesión con tiempo de caducidad. 	

Nota. Fuente (Arellano, 2024).

Figura 17 Historia de usuario 3 de la prueba de concepto

Historia de Usuario	
Código: HU-03	Usuario: Administrador
Nombre: Gestión de roles de usuario	
Prioridad: Alta	Estimación (días): S
Descripción:	
Como administrador, necesito gestionar roles de usuarios para controlar permisos y accesos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Se debe contar con los roles administrador y estudiante. • El administrador puede asignar y eliminar el rol "administrador". • Restricción de acceso según roles. 	

Nota. Fuente (Arellano, 2024).

Product Backlog

El product backlog está detallado de acuerdo con el sistema de tallas de camisetas, registrando cuanto tiempo o esfuerzo tomará realizar cada actividad. En la guía metodológica esta parte de estimación de acuerdo con el sistema de tallas no está detallado, pero el tesista

contaba con este método. Las Figuras 17 y 18 como se realizó el product Backlog de acuerdo el sistema mencionado.

Figura 18 Estimación de la prueba de concepto

Estimación (días)	Talla de camiseta
1-3	S
3-7	M
7-10	L

Nota. Fuente (Arellano, 2024).

Figura 19 Product backlog de la prueba de concepto

Código	Historia de Usuario	Prioridad	Descripción	Estimación
HU-01	Registro de usuarios	Alta	Como estudiante nuevo, deseo registrarme en la aplicación para acceder a información y actividades de redacción, gramática y ortografía.	S
HU-02	Inicio de sesión	Alta	Como estudiante, quiero iniciar sesión para completar actividades y acceder a las evaluaciones.	S

Nota. Fuente (Arellano, 2024).

Sprint 0

El tesista hizo uso de la plantilla descrita en la guía metodológica modificando adaptando a su trabajo. La Figura 19 se detalla la matriz de planificación del Sprint 0.

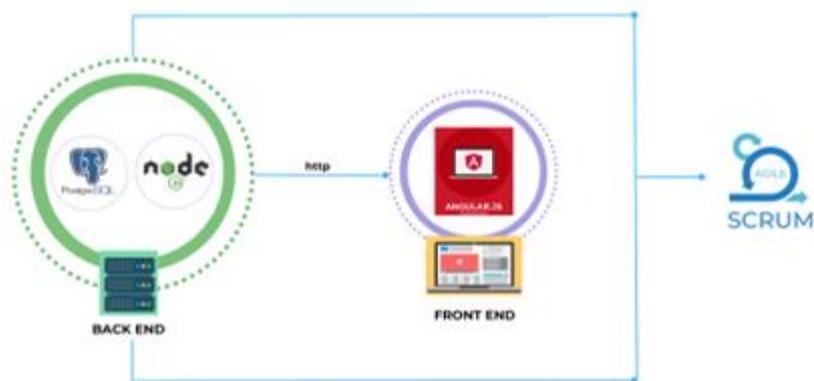
Figura 20 Sprint 0 de la prueba de concepto

Sprint 0	
Tarea	Horas
Identificar y definir los roles del equipo con sus respectivas responsabilidades.	1
Reunión con el equipo scrum.	1
Identificar de manera detallada los requisitos del usuario.	1
Crear un Product Backlog detallando las tareas necesarias.	2
Definir y documentar las historias de usuario basadas en los requisitos del usuario.	3
Elaborar el Sprint Backlog detallando las actividades de cada historia de usuario en su respectivo Sprint.	4
Crear y configurar el entorno de desarrollo local utilizando las tecnologías seleccionadas.	3
Diseñar la estructura de la base de datos utilizando PostgreSQL.	2
Diagramar el esquema de la base de datos.	2
Definir los juegos a poner en la aplicación.	2
Total:	21

Nota. Fuente (Arellano, 2024).

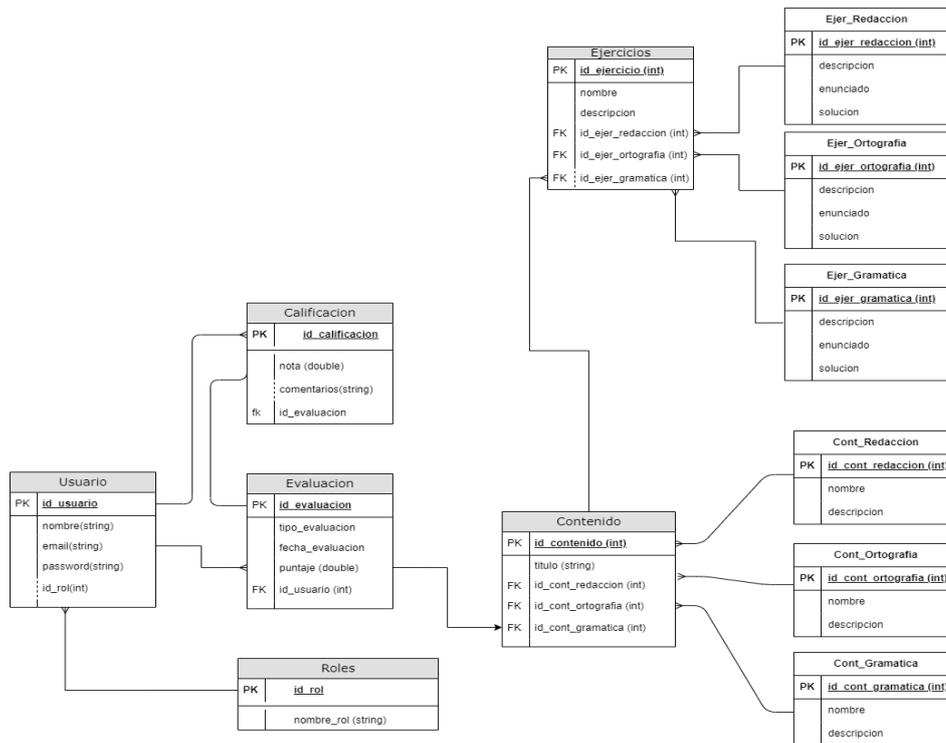
La arquitectura del software se detalla en este sprint, al igual que el diagrama de la base de datos. Ver Figura 20 y 21.

Figura 21 Arquitectura del software de la prueba de concepto



Nota. Fuente (Arellano, 2024).

Figura 22 Diagrama de la base de datos de la prueba de concepto



Nota. Fuente (Arellano, 2024).

3.1.2 Fase 2. Juego

En esta fase se procede con la ejecución de los sprint, el tesista realizó la planificación de cada sprint, en total son cuatro. Esta matriz de planificación de sprint no cuenta la guía metodológica, el tesista se basó de trabajos relacionados para elaborar estas planificaciones de sprint. A continuación, se muestra la Figura 22 del Sprint.

Figura 23 Planificación Sprint de la prueba de concepto

Sprint 1		
ID	Tarea	Horas
TA-01	Desarrollo del API para registro de usuarios	1
TA-02	Implementación de la API en el frontend	1
TA-03	Validación de datos y diseño de mensajes de información	1
TA-04	Diseño de la interfaz para el registro de usuarios	2
TA-05	Desarrollo del API para el inicio se sesión	3
TA-06	Implementación de la lógica de autenticación del usuario a través de una API y generar un token de sesión.	4
TA-07	Implementar una función para recuperar contraseña.	3
TA-08	Diseño de una interfaz para recuperar contraseña	2
TA-09	Diseño de la interfaz para el inicio de sesión	2
TA-10	Definición de roles	2
TA-11	Configuración para restringir el acceso según los roles	
TA-12	Verificación del control de roles	
Total:		21

Nota. Fuente (Arellano, 2024).

Sprint Backlog

Luego de realizar la matriz de planificación de cada sprint, se procede con el sprint backlog de cada uno, donde se especifican las actividades y tareas basadas en las historias de usuario. Se hizo uso de la plantilla que se detalla en la guía metodológica. Ver Figura 23.

Figura 24 Sprint 1 Backlog de la prueba de concepto

ID HU	ID Tarea	Descripción	Estimación (horas)	Total, Sprint horas
HU-01	TA-01	Desarrollo del API para registro de usuarios	3	1
	TA-02	Implementación de la API en el frontend	4	1
	TA-03	Validación de datos y diseño de mensajes de información	5	1
	TA-04	Diseño de la interfaz para el registro de usuarios	4	1
HU-02	TA-05	Desarrollo del API para el inicio se sesión	4	1

Nota. Fuente (Arellano, 2024).

Revisión del sprint 1

Esta revisión permite evidenciar los incrementos del producto. La Figura 24 se evidencia el incremento del sprint 1.

Fecha: 05 de diciembre 2023

Asistentes: Product Owner, Scrum Master, Scrum Team

Descripción: Se evidencia el incremento del producto (entregable)

Figura 25 Revisión Sprint 1 de la prueba de concepto



Nota. Fuente (Arellano, 2024).

Retrospectiva del Sprint 1

Se utilizó la plantilla de acuerdo con la retrospectiva del sprint

Figura 26 Retrospectiva del sprint 1 de la prueba de concepto

Fecha:	20 de diciembre 2023
Asistentes:	Product Owner, Scrum Master, Scrum Team
Título	Descripción
¿Qué salió bien del sprint?	Se cumplió con la creación del registro de usuarios
¿Qué no salió como se esperaba?	Existen varias formas de obtener un token lo que llevo a investigar la mejor forma para el presente caso.
¿Qué mejoras se implementará?	Mejorar el diseño del inicio de sesión.

Nota. Fuente (Arellano, 2024).

3.1.3. Fase 3. Post-Juego

Finalmente, en esta fase se utilizó la plantilla de pruebas de aceptación de la guía metodológica.

Figura 27 Prueba de aceptación de la prueba de concepto

Historia de usuario	Nombre	Funcionalidad	Aceptación	
			SI	NO
HU-01	Registro de usuarios	Registrar los usuarios de forma sencilla y eficaz	X	
HU-02	Inicio de sesión	Iniciar sesión sin complicaciones	X	
HU-03	Gestión de roles de usuario	Tener un rol asignado	X	
HU-04	Gestión de Módulos	De acuerdo con el rol acceder a los módulos	X	
HU-05	Gestión de Temas sobre redacción, ortografía y gramática	Gestionar los temas que se encuentren.	X	
HU-06	Gestión de actividades de redacción, ortografía y gramática	Gestionar todas las actividades.	X	
HU-07	Acceso al contenido relacionado con la redacción, ortografía y gramática.	Visualización del contenido de redacción, ortografía y gramática.	X	
HU-08	Acceso a las actividades relacionadas con la	Visualización de las actividades del contenido de redacción	X	

Nota. Fuente (Arellano, 2024).

Se finaliza con la guía mediante un acta de entrega/recepción por parte del tesista una vez finalizado el proyecto.

Figura 28 Acta de entrega/recepción de la prueba de concepto

<p>UNIVERSIDAD TÉCNICA DEL NORTE</p>  <p>FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS</p>	<h2>ACTA DE ENTREGA RECEPCIÓN DE SOFTWARE</h2>
<p>Fecha de entrega: 17/02/2024</p>	<p>Autor: Jonathan Mauricio Arellano Proaño</p> <hr/> <p>Versión: aplicativo web redacción 0.1</p> <hr/> <p>Lugar: Ibarra, UTN</p>
<p>Proyecto/Sistema de información:</p> <p>Desarrollo de una aplicación web para mejorar la redacción de artículos de divulgación científica en los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte</p>	
<p>Se realiza la entrega por parte del Sr./Srta. Jonathan Mauricio Arellano Proaño con C.I. 100424547-6 del trabajo “Desarrollo de una aplicación web para mejorar la redacción de artículos de divulgación científica en los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte” a la Msc. Silvia Arciniega, desarrollado de acuerdo con lo que se estableció.</p> <p>Productos entregados:</p> <ul style="list-style-type: none"> • Proyecto de desarrollo (código fuente) • Pruebas de funcionalidad • Implementación del proyecto en un servidor web 	
<hr/> <p>RECIBÍ CONFORME</p> <p>Msc. Arciniega Silvia</p> <p>Docente UTN</p>	<hr/> <p>ENTREGUÉ CONFORME</p> <p>Jonathan Arellano</p> <p>Desarrollador</p>

Nota. Fuente (Arellano, 2024).

3.2 Resultados de la prueba de concepto

Para conocer el porcentaje de eficiencia de la guía metodológica se hizo uso de la medida de completitud funcional el cual mide la satisfacción de las necesidades del software. La evaluación de esta guía metodológica es a través de la completitud funcional, que

involucra evaluar la capacidad para ejecutar un sistema y el nivel de aceptación por parte del usuario.

La evaluación es de acuerdo las fases de Scrum: Pre-Juego, Juego y Post-Juego.

3.2.1 Análisis de resultados

Medida de completitud funcional

La evaluación de la completitud de la tarea se emplea para evaluar en qué medida en el cual un conjunto de funciones cubre diversas tareas y objetivos establecidos por el usuario. A continuación, se presenta en la Tabla 11 donde se detalla el procedimiento de evaluación basado en la medida de completitud funcional.

Tabla 19 Completitud Funcional

Nombre	Propósito de la métrica de calidad	Fórmula
Completitud funcional	¿Cuán completa es la implementación de acuerdo con la especificación de tareas?	X= A/B
		A= Número de funciones que no están implementadas
		B= Número de las funciones establecidas en la especificación de tareas. Donde B>0

Nota: El resultado de X se procedió a multiplicar por el valor de la escala de medición, esta escala puede estar entre cero y diez, en este caso se tomó el valor de diez, para saber en qué rango de puntuación se ubica el resultado.

Aplicación de la fórmula para la completitud funcional

La evaluación de la guía se llevó a cabo en base al desarrollo de la prueba de concepto, donde se aplicaron actividades que fueron propuestas en la sección [2.2](#). A continuación, en la Tabla 12 se detallan los resultados obtenidos:

Tabla 20 Evaluación de la guía mediante la medida de completitud funcional

FASE	ACTIVIDAD SCRUM	TAREA SCRUM	Nivel de cumplimiento	Observaciones
PRE-JUEGO	Planteamiento	Definición de roles	SI	
		Definición de requisitos	SI	
		Product Backlog	SI	
JUEGO	Sprint	Sprint Planning	SI	
		Daily Scrum	NO	Al ser un trabajo de titulación no se realiza reuniones diarias
		Revisión de Sprint	SI	
		Retrospectiva de Sprint	SI	
POST-JUEGO	Entrega	Pruebas de integración/aceptación	SI	
		Despliegue	NO	
		Acta entrega/recepción	SI	
Funciones especificadas				10

En la evaluación de la completitud funcional se encuentran especificadas 10 funciones que representan el 100%. Luego de la evaluación se obtuvo como 8 funciones que cumplen con los objetivos de la guía y 2 que no están completas. Porcentaje de éxito aplicando la fórmula:

$$X = A/B$$

A: Número de funciones no satisfactorias

B: Número total de funciones

$$X = 8/10$$

$$X = 0,80$$

3.2.2 Evaluación de los resultados

La interpretación de los resultados se realizó de acuerdo con la escala de los niveles mínimos aceptables de calidad. El modelo de evaluación de la ISO/IEC 25040 se encuentran

en un rango del 0 al 10, organizados en cuatro niveles de puntuación. La manera de medir se define en función de aspectos positivos o negativos, tal como se indica en la Figura 21:

Figura 29 Niveles de calidad

Valor de medición	Nivel de puntuación	Grado de satisfacción
7.91-10	Cumple con los requisitos	Muy satisfactorio
4.91-7.9	Aceptable	Satisfactorio
1.91-4.9	Mínimamente aceptable	No satisfactorio
0-1.9	Inaceptable	No satisfactorio

Nota. Fuente (Paola et al., 2019)

El valor que resultó luego de aplicar la fórmula de satisfacción de funciones multiplicado por la escala de medición esperado que es diez.

$$X = 0.80 * 10 = 8$$

Este valor demuestra que la guía metodológica para la documentación de los capítulos técnicos de los trabajos de titulación que utilicen Scrum tiene un nivel de puntuación que cumple con los requisitos y el grado de muy satisfactorio.

CONCLUSIONES

La elaboración del marco conceptual permitió establecer una base sólida y estructurada para el desarrollo de la guía metodológica. Este marco proporciona los fundamentos teóricos y prácticos necesarios para orientar la elaboración, asegurando coherencia y consistencia con los principios y prácticas establecidos en el marco de trabajo Scrum.

La guía para la documentación del capítulo técnico con el marco de trabajo Scrum, permitió el desarrollo de trabajos de software ordenados y estandarizados, de acuerdo con las tres fases de Scrum (Pre-Juego, Juego y Post-Juego) y tareas, con lo que se pretende estructurar de manera adecuada la documentación del marco metodológico de los Trabajos de Integración Curricular (TIC) de los estudiantes de la carrera de Ingeniería de software de la Universidad Técnica del Norte.

La evaluación de la guía metodológica para la documentación del marco metodológico con el marco de trabajo Scrum, se realizó mediante una prueba de concepto y las métricas de completitud funcional de la ISO/IEC 25023. En esta evaluación, se validó las diez características de la guía metodológica, en base a su ejecución de un trabajo de titulación de un estudiante de octavo nivel de la carrera de software en el período académico 2023-2024. En el resultado de la evaluación se obtuvo el 80% de satisfacción en la utilización de las funciones propuestas, el cual indica un grado de muy satisfactorio y eficiente para la documentación de los TIC de los capítulos técnicos de desarrollo de software basados en el marco de trabajo Scrum.

RECOMENDACIONES

Conocer y mantenerse actualizado con las últimas tendencias y avances en el marco de trabajo Scrum, para facilitar la comprensión e implementación de la guía metodológica. Este trabajo está realizado en base al marco de trabajo Scrum por lo cual si se desea trabajar con otra metodología distinta va a existir confusiones, tomar en cuenta este factor.

Se recomienda incentivar la adopción de la guía metodológica, dado que la guía para la documentación del marco metodológico con el marco de trabajo Scrum ha demostrado ser efectiva, es importante promover su uso entre los estudiantes. Con la realización de talleres o capacitaciones para familiarizar a los usuarios con la guía y fomentar su aplicación en los Trabajos de Integración Curricular (TIC).

En caso de continuar con la investigación se recomienda seguir con evaluaciones regulares para verificar su efectividad y realizar mejoras si es necesario. Las evaluaciones pueden incluir encuestas a los usuarios y análisis de métricas de calidad, para garantizar que la guía siga siendo útil y eficaz en la documentación de los TIC.

Estas recomendaciones pueden ayudar a asegurar que la guía metodológica para documentar el marco metodológico siga siendo una herramienta efectiva y relevante para los estudiantes de la carrera de Ingeniería de software de la Universidad Técnica del Norte y que continúe contribuyendo al éxito de los TIC basados en el marco de trabajo Scrum.

BIBLIOGRAFÍA

- Alonso, M. (n.d.). *Guía para la elaboración de trabajo de grado*. Retrieved November 7, 2023, from https://bib.us.es/derechoytrabajo/sites/bib3.us.es.derechoytrabajo/files/guia_para_elaborar_un_trabajo_final_de_grado_1.pdf
- altexsoft. (2023, March 28). *Technical Documentation in Software Development: Types, Best Practices, and Tools*. <https://www.altexsoft.com/blog/technical-documentation-in-software-development-types-best-practices-and-tools/>
- Alvarez, S. (2022, October 27). *Importancia de la documentación*. <https://desarrolloweb.com/articulos/importancia-documentacion.html>
- Angel, M., Autores, S., Menzinsky, A., López, G., Palacio, J., Sobrino, M. Á., Álvarez, R., & Rivas, V. (2022). *Historias de Usuario Ingeniería de Requisitos Ágil*.
- Anwer, F., Shah Muhammad, S., waheed Waheed, U., Aftab, S., Shah Muhammad Shah, S., & Waheed, U. (2017). Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. In *International Journal of Computer Science and Telecommunications* (Vol. 8, Issue 2). www.ijcst.org
- Arellano, J. (2024). *Desarrollo de una aplicación web para mejorar la redacción de artículos de divulgación científica en los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte*.
- Arizbé, K. (2023, November 26). *Arquitectura de software: ¿Qué es y qué tipos hay?* .
- Beck, Kent. (2008). *User Stories Applied for agile software development*. Addison-Wesley.
- Bektas, A. (2023, April 19). *UserGuiding*. <https://userguiding.com/es/blog/herramientas-documentacion-software/>
- Beneyto Carlos. (2022, March 27). *Medium*. <https://medium.com/startups-es/documentar-un-producto-y-como-hacerlo-97529b29161>
- Calzadilla, A., González, E., Hernández, C., Pleítez, N., & Romero, M. (2018). *Fundamentos del desarrollo ágiles y SCRUM*.
- Casariago, E., & Costa, C. (2022, November 8). *Fisterra*. <https://www.fisterra.com/guias-clinicas/>
- Cohn, M. (2010). *Succeeding with agile. Software development using scrum*.
- Distillery. (2023, June 15). *Distillery*. <https://distillery.com/es/blog/documentacion-de-software-el-valor-de-una-documentacion-adeuada-a-su-finalidad/>

- GEOScéntricos. (2019, May 14). *Medium*.
<https://medium.com/@geos.centricos/manual-de-usuario-para-qué-f7f5e75ce29b>
- Gonçalves, L. (2018). Scrum. *Controlling & Management Review*, 62(4), 40–42.
<https://doi.org/10.1007/s12176-018-0020-3>
- González, J., Dominguez, F. J., Gutiérrez, J. J., & Escalona, M. J. (2014). *Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental*.
- GUÍA OPERATIVA DE LA UNIDAD DE INTEGRACIÓN CURRICULAR. (2023).
- Hadida, S., & Troilo, F. (2020). *La agilidad en las organizaciones: Trabajo comparativo entre metodologías ágiles y de cascada en un contexto de ambigüedad y transformación digital*. <http://hdl.handle.net/10419/238381>
- Humbachano, J. F. (2017, September 25). *Scrum.org*.
<https://www.scrum.org/resources/blog/que-es-scrum>
- Knott, R. (2020, September). *How to Build the Best User Manual*.
<https://www.techsmith.com/blog/user-documentation/>
- Lafuente, A., Gómez, D., & Freire, J. (2018). El arte de documentar. *Ciudadanía Digital y Democracia Participativa. Salamanca: Comunicación Social Ediciones y Publicaciones*, 47–59.
- Le, H. (2017, December 25). *Agile Product Management with Scrum – Roman Pichler*.
- Lee, W. T., & Chen, C. H. (2023). Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering. *Electronics (Switzerland)*, 12(15). <https://doi.org/10.3390/electronics12153291>
- León, C. I. (2022). *Diseño y aplicación de un modelo de reconocimiento de 11 gestos de la mano usando señales EMG y deep learning*.
- Martins, J. (2023, September 19). *Asana*. <https://asana.com/es/resources/what-is-scrum>
- Menzinsky, A., López, G., & Palacio, J. (2016). *Scrum Manager*.
<http://www.streetsofdublin.com/>
- Meza, E. (2014). *Propuesta de una guía metodológica para la elaboración de trabajos especiales de grado, con el fin de facilitar el trabajo de investigación de los aspirantes a maestría en el Ecuador*. International Lifelong Learning University.
- ODS. (2023). *Naciones Unidas*.
<https://www.un.org/sustainabledevelopment/es/infrastructure/>
- Paola, E., Guaña, R., Gabriela, S., Rosado, P., & Quijosaca, F. (2019). *Evaluación de la calidad en uso de un sistema web/ móvil de control de asistencia a clases de docentes y estudiantes aplicando la norma ISO/IEC 25000 SQuaRe*.

- Procida, D. (2020). *The Documentation System*. <https://documentation.divio.com/>
- Quesada, N. (2020). *Estudio sobre Metodologías Ágiles en los Proyectos Software. Propuesta de Plan de Implantación para PYMES*.
- Rea-Peñafiel, M., Maldonado-Arias, T., & Quiña-Mera, A. (2023). Integración de componentes Enterprise Java Beans y Front End Angular para analizar el desempeño de acceso a datos. *INNOVATION & DEVELOPMENT IN ENGINEERING AND APPLIED SCIENCES*, 5(2), 14. <https://doi.org/10.53358/ideas.v5i2.943>
- Resolución Nro. 001-073-CEAACES-2013-13. (2013). *Universidad Técnica del Norte*.
- Rivadeneira, S. (2012). *Metodologías ágiles enfocadas al modelado de requerimientos*.
- Robles Belmonte, M. T. (2017). *Guía Metodológica*. <https://docplayer.es/user/38697302/>
- Roche, J. (2019, January 14). *Las 5 ceremonias Scrum: claves para la gestión de procesos*. <https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>
- Rodríguez, J. (2018). *Guía para el desarrollo y documentación de una aplicación web .NET II*.
- Satish, C. J., & Anand, M. (2016). Software documentation management issues and practices: A survey. *Indian Journal of Science and Technology*, 9(20). <https://doi.org/10.17485/ijst/2016/v9i20/86869>
- Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum*.
- Scrum Manager Bok. (2023, December 12). *Historia de usuario*. https://www.scrummanager.com/bok/index.php/Historia_de_usuario
- Scrum Manager Bok. (2024, January 12). *Modelo original de Scrum para desarrollo de software*. https://www.scrummanager.com/bok/index.php/Modelo_original_de_Scrum_para_desarrollo_de_software
- Torres, A. (2023, December 19). *Optimiza la fase de lanzamiento Scrum con esta metodología paso a paso*. <https://metodologiascrum.com/metodologia-scrum-eventos-y-artefactos-fase-de-lanzamiento-scrum/>

ANEXOS

**GUÍA METODOLÓGICA
PARA DOCUMENTAR
EL MARCO
METODOLÓGICO DE
LOS TRABAJOS DE
INTEGRACIÓN
CURRICULAR CON
SCRUM**

Tabla de contenido

INDICE DE TABLAS	76
ÍNDICE DE FIGURAS	77
1. Introducción	78
2. Desarrollo del capítulo 3 “Marco metodológico”	79
2.1. Enfoques y tipos de investigación	34
2.2. Diseño de la investigación	79
3. Desarrollo del capítulo 4 “Resultados y análisis”	79
3.1. Fase 1. Pre-juego	80
3.1.1. Asignación de roles.....	80
3.1.2. Definición de requisitos.....	81
3.1.3. Product Backlog.....	83
3.1.4. Sprint 0.....	85
3.2. Fase 2. Juego	89
3.2.1. Sprint backlog	90
3.2.2. Daily Scrum.....	92
3.2.3. Revisión de Sprint.....	92
3.2.4. Retrospectiva de Sprint	94
3.3. Fase 3. Post-juego	95
3.3.1. Pruebas de aceptación	95
3.3.2. Despliegue	96
3.3.3. Acta entrega/recepción.....	97
4. Conclusiones	100

INDICE DE TABLAS

Tabla 1 Formato de asignación de roles	80
Tabla 2 Ejemplo de asignación de roles	36
Tabla 3 Formato de Historia de Usuario	82
Tabla 4 Ejemplo de historia de usuario	38
Tabla 5 Formato Product Backlog.....	84
Tabla 6 Ejemplo de Product Backlog	84
Tabla 7 Planificación Sprint 0	86
Tabla 8 Ejemplo de planificación Sprint 0	41
Tabla 9 Formato de Sprint Backlog.....	91
Tabla 10 Ejemplo Sprint Backlog	91
Tabla 11 Formato de revisión de Sprint	93
Tabla 12 Ejemplo de revisión de sprint	93
Tabla 13 Retrospectiva del Sprint	95
Tabla 14 Pruebas de aceptación	96
Tabla 15 Despliegue.....	97
Tabla 16 Acta de entrega/recepción	98

ÍNDICE DE FIGURAS

Figura 1 Estructura del capítulo.....	35
Figura 2 Arquitectura del proyecto.....	42
Figura 3 Diagrama de la base de datos.....	43

1. Introducción

La presente guía metodológica tiene como objetivo ofrecer a los estudiantes de la carrera de ingeniería de software una propuesta para documentar el capítulo metodológico y el capítulo de resultados y análisis de los Trabajos de Integración Curricular (TRIC) enfocados en los proyectos que realicen desarrollo de software con el marco de trabajo Scrum. Este documento está fundamentado en los lineamientos de la (GUÍA OPERATIVA DE LA UNIDAD DE INTEGRACIÓN CURRICULAR, 2023) y la estructura de las fases fundamentales para desarrollo de proyectos propuestas por el marco de trabajo Scrum (Anwer et al., 2017). En cada fase se explica las actividades clave y los entregables esperados, con el objetivo de proporcionar una visión clara y concisa de cómo aplicar Scrum en el desarrollo del TRIC.

¿Cómo utilizar esta Guía Metodológica? Mediante una revisión inicial de los apartados que contiene la guía metodológica, para comprender su estructura, objetivos y el proceso que se propone. Es importante seguir las actividades que se indica de manera secuencial y adecuada.

2. Desarrollo del capítulo 3 “Marco metodológico”

2.1. Enfoques y tipos de investigación

[El enfoque y tipos de investigación lo definirán cada director y el tesista del trabajo de titulación]

2.2. Diseño de la investigación

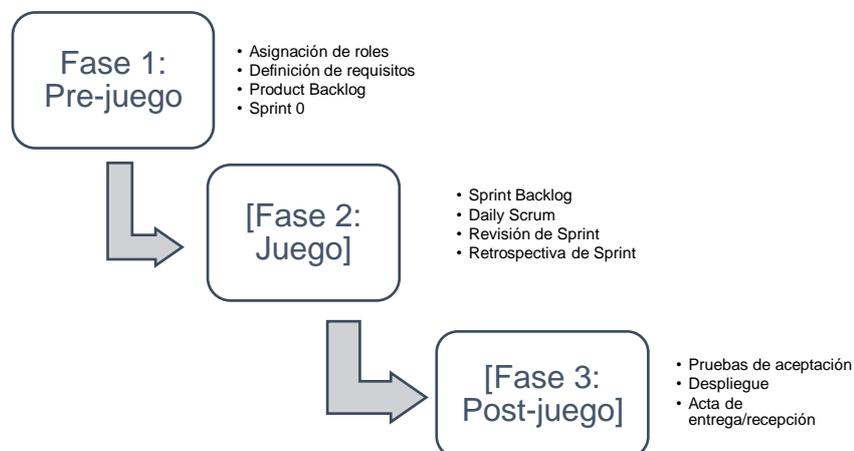
El diseño de la investigación se basa en las fases iniciales del enfoque Design Science Research (DSR).



3. Desarrollo del capítulo 4 “Resultados y análisis”

En este capítulo se describe el desarrollo del proyecto [Nombre del proyecto de desarrollo de software] basándose en el marco de trabajo Scrum. La Figura 1 muestra estructura del capítulo, y sus principales componentes.

Figura 30 Formato estructura del capítulo



NOTA: Esta sección es una introducción del capítulo que muestra la estructura, los temas y subtemas a tratar, para que el lector pueda tener una visión general del capítulo.

3.1. Fase 1. Pre-juego

Esta fase de pre-juego también se conoce como la fase de preparación, donde se establece varias actividades clave para iniciar un nuevo Sprint. Las actividades entre las que se listan son: la visión, el presupuesto, el sprint backlog inicial, además de la arquitectura y diseños generales (Rivadeneira, 2012).

3.1.1. Asignación de roles

Es en esta sección se asigna los roles Scrum a los integrantes del proyecto [[Nombre del proyecto de desarrollo de software](#)]. A continuación, en la **¡Error! No se encuentra el origen de la referencia.** se muestra los roles y una descripción corta del cargo e institución del equipo de trabajo.

Tabla 21 Formato de asignación de roles

Nombre ³⁶	Rol ³⁷	Cargo ³⁸
[Nombre del Tutor]	Product Owner	Director de Trabajo de Integración Curricular (Docente UTN)
[Nombre de Tesista]	Scrum Master	Estudiante de la carrera de software UTN
[Nombre de Tesista(s)]	Team Development	Estudiante(s) de la carrera de software UTN

³⁶ **Nombre:** El nombre completo del integrante del proyecto.

³⁷ **Rol:** Se asigna en base al marco de trabajo Scrum.

³⁸ **Cargo:** Descripción corta del cargo o institución a la que pertenece.

[Nombre del Asesor] y/o [Nombre de interesado(s)]	Stakeholders	Descripción corta del cargo e institución de los interesados
---	--------------	--

La Tabla 2 muestra el formato de asignación de roles con todos los campos completados.

Tabla 22 Ejemplo de asignación de roles

Nombre	Rol	Cargo
PhD. Antonio Quiña MSc.	Product Owner	Director de Trabajo de Integración Curricular (Docente UTN)
Ruth Alejandra Huertas Burgos	Scrum Master	Estudiante de la carrera de software UTN
Ruth Alejandra Huertas Burgos	Team Development	Estudiante de la carrera de software UTN
PhD. Cathy Guevara MSc. Ing. Mauricio Rea MSc.	Stakeholders	Asesores del TRIC (Docente UTN)

NOTA: En el marco de trabajo de Scrum, la primera tarea es designar roles al equipo de trabajo involucrado en el proyecto, y tomando en cuenta el reglamento (Resolución Nro. 001-073-CEAACES-2013-13, 2013) de la UTN, los involucrados en el trabajo de integración curricular son: un tutor, un asesor, y uno o dos tesistas (podrán ser dos estudiantes siempre y cuando sea justificado). Por lo tanto, se tomará en cuenta los siguientes roles de Scrum: Product Owner, Scrum Master, Team Development y Stakeholders (Beck, 2008).

3.1.2. Definición de requisitos

La definición de requisitos se realiza mediante Historias de Usuario (HU), las cuales, son una técnica ágil que captura las necesidades del software desde la perspectiva del usuario final. Cada historia describe brevemente una funcionalidad o característica deseada, centrada en el valor que aporta al usuario. A continuación, se detalla las historias de usuario que describen los requisitos funcionales específicos para el desarrollo del proyecto [[Nombre del proyecto de desarrollo de software](#)].

Tabla 23 Formato de Historia de Usuario

Historia de usuario		
ID³⁹:	[Código HU-001]	Usuario⁴⁰: [Nombre del usuario de la historia de usuario]
Nombre⁴¹:	[Asignación de un nombre a la HU]	
Prioridad⁴²:	Dependencia⁴³:	Estimación⁴⁴:
[Baja, Media, Alta]	[Identificadores de otras HU]	[Número de estimación requerida para implementar la funcionalidad]
Descripción⁴⁵:	Como un [usuario] quiero [descripción de la funcionalidad] con la finalidad de [descripción de razón o resultado].	
Criterios de Aceptación⁴⁶:	[Valida que la funcionalidad implementada cumple con las expectativas del usuario y del equipo de desarrollo]	

A continuación, en la Tabla 4 se indica una Historia de Usuario.

³⁹ **ID:** Identificador único de la historia de usuario.

⁴⁰ **Usuario:** Nombre del usuario que se asigna esa HU.

⁴¹ **Nombre:** Debe ser corto y claro.

⁴² **Prioridad:** Determina el nivel de importante a implementarse.

⁴³ **Dependencia:** En caso de que exista relación.

⁴⁴ **Estimación:** Esfuerzo requerido para implementar la funcionalidad descrita en la historia de usuario.

⁴⁵ **Descripción:** Síntesis de la historia de usuario.

⁴⁶ **Criterios de aceptación:** Una lista de condiciones que deben cumplirse para que la historia de usuario se considere completada.

Tabla 24 Ejemplo de historia de usuario

Historia de usuario			
ID:	HU-001	Usuario:	Cliente
Nombre:	Creación de formatos		
Prioridad:	Dependencia:	Estimación:	
Media	NO	7	
Descripción:	Como un cliente quiero utilizar la estructura de los formatos con la finalidad de obtener una correcta documentación		
Criterios de Aceptación:	<ul style="list-style-type: none">• Las plantillas deben poder adaptarse al proyecto• Cada campo debe ser comprensible y claro.		

NOTA: En las historias de usuario es fundamental escribir en un lenguaje claro y comprensible para todos los miembros del equipo, evitando tecnicismos innecesarios. Se debe considerar la descripción clara y concisa de la funcionalidad, la comprensión universal y la estimación de esfuerzo.

Además de los campos que se consideran necesarios se puede añadir otros campos adicionales según el tipo de proyecto, pueden ser aconsejable campos como:

Título: Descriptivo de la historia de usuario.

Observación: Para aclarar la información o cualquier uso que pueda ser de utilidad (Menzinsky et al., 2016).

3.1.3. Product Backlog

Con una definición precisa de los requisitos, se procede a crear el product backlog, que es esencialmente una lista priorizada de las funcionalidades que darán vida al producto. Los elementos más relevantes se sitúan en la parte superior de la lista (Le, 2017). La Tabla 3 indica el product backlog.

Tabla 25 Formato Product Backlog

Nro. Orden⁴⁷	ID Historia de Usuario⁴⁸	Descripción⁴⁹	Estimación⁵⁰
[Ordenar según el nivel de importancia]	[Identificador de la historia de usuario]	[Descripción corta de la historia de usuario]	[Estimación de la historia de usuario]
	[Código-HU]	[Crear vistas]	
	[Código-HU]	[Crear CRUD]	
	[Código-HU]	[Obtener consulta]	
...

Se muestra a continuación la Tabla 6 utilizando el formato de Product Backlog a modo de ejemplo con los campos completados.

Tabla 26 Ejemplo de Product Backlog

Nro. Orden	ID Historia de Usuario	Descripción	Estimación
1	HU-001	Crear formatos	7
2	HU-002	Crear listas	7
3	HU-003	Crear vistas	5

⁴⁷ **Nro. Orden:** Indica el orden que están las historias de usuario.

⁴⁸ **ID Historia de usuario:** Identificador de la historia de usuario.

⁴⁹ **Descripción:** Una descripción corta de la historia de usuario.

⁵⁰ **Estimación:** Tiempo estimado real para cumplir con esa tarea.

4	HU-004	Crear vistas	5
...

NOTA: El Product Backlog se debe mantener actualizado y revisar periódicamente para ajustar las prioridades según las necesidades del proyecto. es importante que las historias de usuario sean claras, concisas y estén bien definidas, de manera que el equipo de desarrollo pueda entenderlas fácilmente.

(Cohn, 2010) indica que se debe realizar el product backlog de acuerdo con el acrónimo DEEP para resumir los atributos clave de un buen product backlog, se describen a continuación:

- **Detallar apropiadamente:** Las historias de usuario que se encuentran en las primeras posiciones del product backlog y que se abordarán pronto deben estar lo suficientemente claras como para poder completarlas durante el próximo sprint. Por otro lado, las historias que no se abordarán en el corto plazo deben estar descritas con menos detalle.
- **Estimado:** El trabajo pendiente es una lista de todo el trabajo por realizar. Es una útil herramienta de planificación, debido a que los elementos que se encuentran más abajo en el trabajo pendiente no se entienden completamente, las estimaciones asociadas con ellos serán menos precisas que las estimaciones dadas a los elementos en la parte superior.
- **Emergente:** Tomar en cuenta que el product backlog no es estática, cambia con el tiempo, se agregará eliminarán o se cambiarán las prioridades de las historias de usuarios sobre el trabajo pendiente del producto.
- **Prioriza:** El product backlog debe ordenarse con los elementos más valiosos en la parte superior y lo menos valiosos en la parte inferior. Al trabajar siempre se debe hacer en orden de prioridad, el equipo puede maximizar el valor del producto o sistema que se está desarrollando.

3.1.4. Sprint 0

Durante una fase inicial llamada Sprint 0, se llevó a cabo la planificación del proyecto de desarrollo de software. En este período se definió la arquitectura tecnológica, se estableció

un esquema inicial para la base de datos y se elaboró un diagrama de procesos del sistema. La Tabla 4 indica las tareas llevadas en este Sprint.

Tabla 27 Planificación Sprint 0

Asistentes ⁵¹ :	[Indicar los roles de los asistentes]	
Objetivo ⁵² :	[Hay que describir que se realizará en este Sprint]	
Duración ⁵³	Fase de desarrollo ⁵⁴	Tarea ⁵⁵
2	[Diseño]	[Diseñar el diagrama tecnológico...]
2	[Planificación]	[Esquema inicial de la base de datos...]
3	[Revisión]	...

La Tabla 8 se indica el ejemplo de la planificación del Sprint 0.

Tabla 28 Ejemplo de planificación Sprint 0

Asistentes :	Product Owner, Scrum Master, Team Development.	
Objetivo:	Definir una arquitectura tecnológica	
Duración	Fase de desarrollo	Tarea
3	Diseño	Diseñar el diagrama tecnológico
2	Diseño	Esquema inicial de la base de datos
...

⁵¹ **Asistentes:** Las personas que estuvieron presentes en la planificación

⁵² **Objetivo:** Que se desea cumplir en esa planificación

⁵³ **Duración:** Tiempo que tomará en realizar esa tarea

⁵⁴ **Fase de desarrollo:** Conjunto de etapas que contienen el proceso de creación del producto

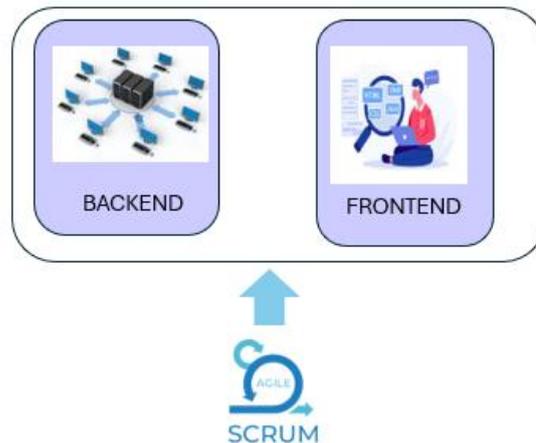
⁵⁵ **Tarea:** Nombre de la tarea a realizar

NOTA: A diferencia de los sprints regulares en los que se entregan incrementos de producto funcionales, en el Sprint 0 no se espera entregar un incremento de producto completo y potencialmente entregable.

Arquitectura del proyecto

La Figura 2 muestra la arquitectura de la aplicación las tecnologías que se utilizarán para el desarrollo de la aplicación. Para el manejo de datos se utilizará la base de datos [Nombre del gestor de base de datos], las herramientas tecnológicas que se utilizarán son: [Nombre de las herramientas tecnológicas a usar].

Figura 31 Ejemplo de Arquitectura del proyecto



NOTA: La arquitectura tecnológica es fundamental para el diseño y desarrollo de sistemas de software eficientes, seguros y escalables, y debe ser cuidadosamente planificada y gestionada para garantizar el éxito del proyecto.

(Arizbé, 2023) Indica que la arquitectura de un aplicativo puede incluir:

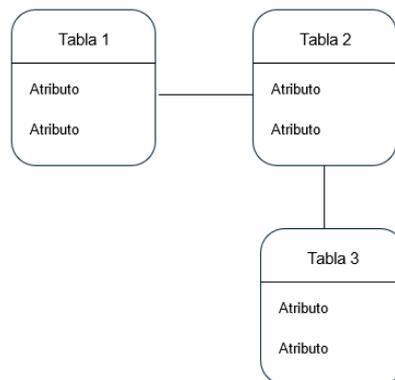
- **Componentes:** Son módulos, servicios o partes del software que realizan tareas específicas dentro del sistema.

- **Conexiones:** Describen como estos componentes se comunican entre sí. Implica interfaces, protocolos de comunicación y flujos de datos que permitan una interacción efectiva entre los diferentes elementos del sistema.
- **Patrones de diseño:** Son soluciones establecidas y recurrentes para problemas comunes en el diseño de software. Estos patrones brindan una guía probada para estructurar el software de manera eficiente, ayudando a mejorar la calidad y mantenibilidad de código.
- **Estilo arquitectónico:** es el enfoque de alto nivel que define la estructura general del sistema. Algunos ejemplos de estilos arquitectónicos incluyen la arquitectura de tres capas, la arquitectura orientada a servicios, la arquitectura de microservicios, entre otros.

Diagrama de la base de datos

En esta sección se utilizó como gestor de base de datos [[Nombre del gestor de base de datos](#)], la Figura 3 se detalla las tablas con los respectivos atributos y entidades.

Figura 32 Ejemplo del diagrama de la base de datos



NOTA: Los diagramas de base de datos pueden ser de varios tipos, como diagramas entidad-relación (ER), diagramas de modelo relacional, y diagramas de modelo de objetos.

3.2. Fase 2. Juego

En esta fase del juego existen una serie de iteraciones conocidas como sprint, aprender a entregar software en cada sprint es uno de los mayores desafíos, la intención es que en cada sprint se entregue algo de valor inmediato a los usuarios o clientes. Se debe asegurar de terminar un sprint y estar preparado para el siguiente, la importancia de establecer y cumplir con un objetivo para el sprint y la necesidad de valorar cuánto tiempo tiene el equipo para alcanzar un objetivo (Cohn, 2010).

Entregar software funcional en cada sprint, es tan importante que uno de los cuatro valores dados en el manifiesto ágil establece que se debe valorar “el software funcional ante la documentación ágil”. Se enfatiza el software funcional por tres razones clave:

- **Software funcional fomenta la retroalimentación:** El equipo de trabajo puede recopilar más y mejores comentarios si proporciona un producto funcional pero parcial a los usuarios, al contrario de proporcionar un documento sobre lo que servirá el producto. Existirá una interacción, brindarán mejores comentarios al poder ver y tocar el producto.
- **Software funcional ayuda a medir el progreso:** Software que funciona y que requiere la entrega de una parte del valor del usuario en cada sprint, evita que no se tenga una idea de cuanto queda por hacer.
- **Software funcional permite entregar antes si así lo desea:** Entrega el producto antes puede ser muy valioso, en un mundo competitivo y cambiante (Cohn, 2010).

Dedica un poco de esfuerzo en cada sprint preparándote para el siguiente sprint. Evitar los sprints de actividades específicas, el equipo decide utilizar un sprint para análisis y diseño, un segundo sprint para codificación y un tercero para pruebas, pero con ello trae tres desventajas:

- **Existe un riesgo de cronograma:** planificar cuanto trabajo se puede realizar en un sprint será más propenso a errores porque el esfuerzo depende en gran medida de la calidad del trabajo realizado en el sprint anterior.
- **Mas tiempo para pasar de la idea a la función probada y en ejecución:** Prolonga el tiempo necesario para recibir comentarios de los clientes, usuarios u otras personas.
- **No resuelve el problema de la superposición de trabajos:** Cuando todo el trabajo se realiza en un sprint, todo el equipo se mueve al mismo ritmo. Cuando se introduce sprints de actividades específicas permitimos que los diferentes subequipos

progresen a diferentes ritmos, hace que se acumule trabajo frente a algunos subequipos (Cohn, 2010).

Durante el desarrollo iterativo e incremental del proyecto es necesario mantener horarios regulares y estrictos. Los beneficios de los sprints manteniendo una longitud de tiempo fija:

- **Los equipos de trabajo se benefician de un ritmo regular:** Cuando se tiene una cadencia regular de entre una y cuatro semanas ayuda a los equipos a adaptarse al ritmo de trabajo que más les convenga.
- **La planificación de sprint se vuelve más fácil:** Tanto la planificación de sprints como de lanzamientos se simplifica cuando los equipos mantienen una duración de sprint constante.
- **La planificación de lanzamiento se vuelve más fácil:** Los equipos Scrum derivan sus planes de lanzamientos empíricamente, estiman el tamaño del trabajo a realizar en un proyecto y luego miden la cantidad completado por sprint.
- **Aprovechar energía:** Elegir la longitud al inicio de cada sprint es un desperdicio de energía, experimente con un par de veces, tome una decisión y manténgala firme hasta que exista una razón importante para cambiar.

Al enfatizar que todos los sprint deben tener la misma duración, no quiere decir que deba ser una regla inquebrantable, se debe elegir un día de la semana que funcione bien en tu entorno y comenzar todos los sprints ese día.

Un error común es extender un sprint, mantener los sprints estrictamente limitados en el tiempo refuerza la idea de avanzar continuamente el proyecto, cada cierto tiempo el equipo debe entregar algún incremento de producto nuevo potencialmente comercializable. Si se permite que los plazos varíen o se extienden ocasionalmente, esta valiosa disciplina se pierde (Cohn, 2010).

3.2.1. Sprint backlog

Es un elemento para visualizar el trabajo del sprint y está gestionado por el equipo de desarrolladores. El objetivo es mantener la transparencia dentro del desarrollo. Todo el trabajo que el equipo de desarrolladores haya seleccionado para hacer durante el siguiente sprint pasa al sprint backlog (Calzadilla et al., 2018). La Tabla 9 se detalla los elementos de la plantilla del Sprint Backlog.

Tabla 29 Formato de Sprint Backlog

Fecha ⁵⁶ :	[día, mes, año]		
Asistentes ⁵⁷ :	[Indicar los roles de los asistentes]		
Historia de Usuario ⁵⁸	Fase de desarrollo ⁵⁹	Tarea ⁶⁰	Estimación ⁶¹
[Código HU-001]	[Análisis]	[Revisar documentación]	1
	[Pruebas]	[Realizar pruebas de petición]	2
	[Desarrollo]	[Crear función de obtener token]	3
[Código HU-002]	[Análisis]
	[Pruebas]
	[Desarrollo]
...
TOTAL HORAS			[X]

A modo de ejemplo se presenta la Tabla 10 del formato del sprint backlog.

Tabla 30 Ejemplo Sprint Backlog

Fecha:	05 de marzo del 2024		
Asistentes:	Product Owner, Scrum Master, Team Development		
Historia de Usuario	Fase de desarrollo	Tarea	Estimación

⁵⁶ **Fecha:** Especificar el día, mes y año que se realizó el Sprint Backlog.

⁵⁷ **Asistentes:** Las personas que estuvieron presentes en la planificación.

⁵⁸ **Historia de Usuario:** Identificador de la historia de usuario.

⁵⁹ **Fase de desarrollo:** Conjunto de etapas que contienen el proceso de creación del producto.

⁶⁰ **Tarea:** Nombre de la tarea a realizar.

⁶¹ **Estimación:** Tiempo estimado (horas) asignada para cada tarea.

HU-001	Análisis	Revisar documentación del marco de trabajo Scrum	4
	Pruebas	Realizar pruebas de petición en el sistema	2
	Desarrollo	Crear función de obtener token	3
HU-002	Análisis
	Pruebas
	Desarrollo
...
TOTAL HORAS			9

NOTA: El Sprint Backlog debe actualizarse continuamente a lo largo del sprint, a medida que se completan tareas, surgen nuevos requisitos o se identifican cambios en las prioridades.

3.2.2. Daily Scrum

El presente trabajo está orientado a los trabajos de titulación y las personas involucradas son mínimas por lo que se omite esta sección, debido a que no se puede realizar las reuniones diarias cuando el equipo está conformado por una o dos personas y no se elaboró una plantilla.

3.2.3. Revisión de Sprint

En esta sección se debe detalla los incrementos hechos para la entrega del Sprint, se evidencia con imágenes que demuestran el avance del producto, el trabajo completado durante el sprint, mostrando las funcionalidades implementadas y cómo cumplen con los criterios de aceptación La Tabla 11 se plasma lo realizado.

Tabla 31 Formato de revisión de Sprint

Fecha ⁶² :	[día, mes, año]
Asistentes ⁶³ :	[Indicar los roles de los asistentes]
Descripción ⁶⁴	[Detallar lo que se realizó]
[Colocar una imagen del incremento realizado]	
	

En la Tabla 12 se presenta un ejemplo utilizando el formato mencionado de revisión.

Tabla 32 Ejemplo de revisión de sprint

Fecha ⁶⁵ :	05 de marzo del 2024
Asistentes ⁶⁶ :	Product Owner, Scrum Master, Team Development
Descripción ⁶⁷	Diseño de una guía metodológica en base al marco de trabajo Scrum

⁶² **Fecha:** Especificar el día, mes y año que se realizó la revisión del Sprint.

⁶³ **Asistentes:** Las personas que estuvieron presentes en la revisión.

⁶⁴ **Descripción:** Una descripción corta y clara de lo que se realizó.

⁶⁵ **Fecha:** Especificar el día, mes y año que se realizó la revisión del Sprint.

⁶⁶ **Asistentes:** Las personas que estuvieron presentes en la revisión.

⁶⁷ **Descripción:** Una descripción corta y clara de lo que se realizó.

FASE	ACTIVIDAD SCRUM	TAREA SCRUM	Producto
PRE-JUEGO	Planteamiento	Definición de roles	Involucrados
		Definición de requisitos	Historias de Usuario
		Product Backlog	Lista priorizada
JUEGO	Sprint	Sprint Planning	Arquitectura
		Daily Scrum	No Aplica
		Revisión de Sprint	Lista de Observaciones
		Retrospectiva de Sprint	Lista de mejoras
POST-JUEGO	Entrega	Pruebas de integración/aceptación	Artefactos de validación
		Despliegue	No Aplica
		Acta entrega/recepción	Documento firmado

NOTA: La revisión de sprint va más allá de la demostración del incremento del producto y revisar el estado actual del proyecto, si no que sirve como un nuevo punto de referencia para tomar decisiones sobre el futuro y establecer una estrategia de negocio (Roche, 2019).

Se realiza al finalizar cada sprint para indicar el incremento del software desarrollado. El equipo de desarrollo hace una demostración sobre dicho incremento del software y resume las funcionalidades que han sido desarrolladas y entregadas en ese sprint, la reunión de revisión da mucho valor para el proyecto porque se obtiene una retroalimentación clara y directa acerca del software (Quesada, 2020).

3.2.4. Retrospectiva de Sprint

Es una reunión que tiene como objetivo reflexionar sobre el trabajo y el resultado del último sprint. Se realiza luego de la revisión del sprint y antes de la planificación del siguiente sprint. El resultado de la reunión es la creación de un plan de mejora a implantar tan pronto como sea posible, esta actividad es una de las más importantes en el proceso ágil (Quesada, 2020). La Tabla 13 se indica lo tratado en esta reunión.

Tabla 33 Retrospectiva del Sprint

Fecha:	[día, mes, año]
Asistentes:	[Indicar los roles de los asistentes]
Título	Descripción
¿Qué salió bien del sprint? ⁶⁸	[Se terminó la creación de las vistas]
¿Qué no salió como se esperaba? ⁶⁹	[Existen varias formas de presentar, lo que llevó a investigar más acerca...]
¿Qué mejoras se implementará? ⁷⁰	[Se prepara crear los archivos...]

NOTA: La retrospectiva del sprint es una parte clave del marco de trabajo Scrum, ya que ayuda al equipo a adaptarse y mejorar continuamente, lo que contribuye a la eficacia y la satisfacción en el trabajo.

3.3. Fase 3. Post-juego

En esta fase todas las actividades se dan por terminado y finalmente se cierra el proyecto, realizando la liberación del producto, la generación de documentación final (León, 2022). Entre las tareas de cierre se encuentran: integración, pruebas del sistema, documentación de usuario (Scrum Manager Bok, 2024).

3.3.1. Pruebas de aceptación

Se acepta la funcionalidad de la historia de usuario especificada en la fase de la definición de los requisitos. Todas estas deben cumplir para dar paso a la totalidad del proyecto. Las pruebas de aceptación suelen realizarse al final del proceso de desarrollo,

⁶⁸ ¿Qué salió bien del sprint?: Aspectos positivos del sprint, logros alcanzados, tareas que salieron bien.

⁶⁹ ¿Qué no salió como se esperaba?: Obstáculos que dificultaron el trabajo.

⁷⁰ ¿Qué mejoras se implementará?: Posibles acciones para mejorar el siguiente sprint.

antes de que el sistema sea puesto en producción, y son una parte importante del proceso de garantía de calidad del software. A continuación, se muestra la Tabla 14.

Tabla 34 Formato de pruebas de aceptación

ID HU	Nombre	Funcionalidad	Aceptación	
			SI	NO
[Identificador de la historia de usuario]	[Obtener consulta]	[Obtener la consulta de forma sencilla mediante...]	X	
[Código HU–0001]	[Crear vistas]	[Visualizar las vistas de...]	X	
[H-XX-03]	[Crear CRUD]	[Realizar las acciones...]	X	
[H-XX-04]	[Obtener consulta]	[Obtener consular de forma sencilla mediante...]	X	
....

NOTA: Las pruebas de aceptación se deben realizar a medida que se va produciendo el desarrollo de los sprint y cualquier error o incidencia se podrá detectar de manera temprana, para planificar su resolución en los siguientes sprints (González et al., 2014)

Son aquellas en las que los usuarios o expertos en el negocio validan los requisitos que se definieron al inicio del proyecto. Las reuniones que se realicen son de seguimiento de presentación y aceptación del sprint, se analizan los casos de uso cubiertos en el sprint, además de los aspectos relativos a la funcionalidad cubierta (González et al., 2014).

3.3.2. Despliegue

La fase de lanzamiento se da una vez que el producto esté totalmente terminado y es momento de entregar al cliente. Esto debe realizarse de manera clara y documentada, asegurándose que se cumplan todas las expectativas acordadas (Torres, 2023). Incluye la configuración de servidores, la transferencia de archivos, la configuración de base de datos,

la implementación de medidas de seguridad y la realización de pruebas finales para asegurar que la aplicación funcione correctamente. Ver Tabla 15

Tabla 35 Formato de documentar el despliegue

Recursos de Despliegue	
Sistema Operativo	Especificar el sistema operativo con el que se trabajó [Windows, Fedora]
Servidor de despliegue	Especificar en que servidor se encuentra [Heroku, Vercel]
Enlace de la aplicación desplegada	[Dirección para acceder a la aplicación]
Máquina	Almacenamiento [Detallar los recursos utilizados]
	CPU ...
	Memoria ...
Observaciones	En caso de que exista alguna novedad

NOTA: Al desplegar una aplicación, es importante planificar cuidadosamente el proceso, realizar pruebas exhaustivas y asegurarse de tener medidas adecuadas de seguridad y gestión de errores para garantizar un despliegue exitoso.

3.3.3. Acta entrega/recepción

Una vez concluido el trabajo se realiza a entregar por parte el Product Owner del proyecto mediante un acta de entrega recepción. En el proceso de entrega del producto los principales involucrados son los siguientes:

- **Responsable de entrega:** Se refiere al responsable del área de TIC de una entidad o dependencia universitaria y quién deberá realizar la entrega de los elementos técnicos y de información.
- **Responsable de recepción:** Se denomina como a la nueva persona que tomará el cargo de la persona responsable del área de TIC de la entidad o dependencia.

- **Autoridad administrativa:** Hace referencia a la figura administrativa encargada de gestionar las bajas de personal y el control de los bienes y servicios generales de la entidad o dependencia

Tabla 36 Acta de entrega/recepción



ACTA DE ENTREGA RECEPCIÓN DE SOFTWARE

Fecha de entrega:

[dd/mm/aaaa]

<Fecha en la cual se
hace la entrega del
requerimiento>

Autor:

[Nombre del ingeniero desarrollador]

Versión: [Número de a versión del software]

Lugar: [Ciudad en la que se realizó la entrega]

Proyecto/Sistema de información:

[Nombre del proyecto o sistema de información]

Se realiza la entrega por parte del Sr./Srta. [Nombres completos del estudiante] con C.I. [Cédula de Identidad] del trabajo “[Nombre del desarrollo del sistema]” para el área de [Nombre de entidad u organización], desarrollado de acuerdo con lo que se estableció y que puedan dar uso de este.

Productos entregados:

- [Proyecto de desarrollo (código fuente)]
- [Manual técnico y de usuario]
- Pruebas de funcionalidad
- Capacitación al funcionario encargado

RECIBÍ CONFORME

[Nombre del
responsable que
recibe]

Cargo

ENTREGUÉ CONFORME

[Nombre del
responsable que
entrega]

Cargo

4. Conclusiones

- La guía para la documentación del capítulo técnico con el marco de trabajo Scrum, permitió el desarrollo de trabajos de software ordenados y estandarizados, de acuerdo con las tres fases de Scrum (Pre-Juego, Juego y Post-Juego) y tareas, con lo que se pretende estructurar de manera adecuada la documentación de los capítulos técnicos de los trabajos de grado de los estudiantes de la carrera de software de la Universidad Técnica del Norte.
- Este marco proporciona los fundamentos teóricos y prácticos necesarios para orientar la elaboración, asegurando coherencia y consistencia con los principios y prácticas establecidos en el marco de trabajo Scrum