UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

"SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022"

Tesis presentado por : Bach. Judith Maribel Vilca Alviar

Tipo de Investigación : Aplicada

Área de Investigación : Gestión de Proyectos de Software

Asesor : Mg. Ing. Hubner Janampa Patilla

Ayacucho - Perú

2022

Dedicatoria

La presente tesis dedico principalmente a Dios, por permitirme haber llegado hasta este momento tan importante de mi formación profesional. Asimismo, a mi familia por el apoyo incondicional que me ha permitido lograr una más de mis metas. Sin olvidar a esta casa de estudios que me formó con principios y valores y más aún, me enseñó valorar la inigualable cultura de la ciudad de Huamanga.

Agradecimientos

A nuestros maestros de la Escuela Profesional de Ingeniería de Sistemas de esta comunidad emblemática, por inculcarme su eficaz saber durante desarrollo de mi carrera, también doy las gracias al asesor de tesis quien me ha guiado con sus conocimientos, experiencias y su rectitud como docente, y a mis amigos por el apoyo y motivación que me brindaron dentro y fuera de las aulas.

Contenido

Lista de Figuras

Lista de Tablas

Resumen

En el mundo de los negocios, se están aplicando nuevas tecnologías fundamentales para mejorar la calidad en los proyectos de software teniendo en cuenta la necesidad del negocio y las expectativas del cliente. Las empresas, buscan nuevas metodologías ágiles efectivas, reconociendo que esto se da solo si se adopta la adecuada y según las necesidades de la organización.

El enfoque de gestión de proyectos ágil ha crecido y se está desarrollando a través del proceso empírico. La programación extrema consta de buenas prácticas y principios además de enfocarse en la participación del cliente detectando sus necesidades mediante pruebas repetidas y mejoras del prototipo. La gestión ágil a través del método Scrumban casi obliga al cliente a incrementar su participación en el proyecto, así mismo, mediante la gestión del tiempo y reuniones específicas es beneficiado para realizar un seguimiento del proceso y el estado del proyecto.

El objetivo de este trabajo de investigación es utilizar el Scrumban sobre la programación extrema para ayudar a la gestión de proyectos de software en entornos ágiles, para ello se analizará la implementación de Scrumban utilizando una herramienta para gestionar el ciclo de vida del desarrollo del software, ScrumDo como soporte de la metodología.

Palabras clave: Scrumban, XP, ScrumDo, gestión de proyectos de software, entornos ágiles.

Introducción

Desde hace una década, el desarrollo de software ágil ha estado en la industria del software. En el año 2001, diecisiete profesionales especializados en metodologías consideradas ligeras establecieron principios y características comunes en cada una, dando como resultado la "Alianza Ágil", que constituyó el documento denominado "Manifiesto Ágil" (Mancl y Fraser, 2019). Este momento de gran trascendencia, alentaron a las empresas a mejorar sus formas de trabajar en proyectos de software (Stol & Fitzgerald, 2018).

Actualmente existen varias metodologías ágiles y todos tienen diferentes características que se pueden aplicar en varios tipos de requerimientos de desarrollo de proyectos. Scrum y Kanban son los marcos ágiles más adoptados, en los dominios del desarrollo de software (RV. Anand, K. Bhavsar et al., 2020).

La presente investigación titulada: "Scrumban sobre la Programación Extrema para la Gestión de Proyectos de Software en entornos ágiles" se inserta en las líneas "metodologías ágiles" y en la línea de investigación "gestión de proyectos de software", donde se pretende implementar Scrumban basado en la integración de Scrum y Kanban, bajo el enfoque de los principios y "buenas prácticas y llevarlas al extremo" (Rosado et al., 2012), con la Programación Extrema (XP).

La investigación se realizó por el interés de conocer y mejorar las limitaciones que presentan estas metodologías, por el cual me motiva a proponer un modelo de desarrollo de software de Scrumban sobre la Programación extrema, de esta manera, poder entregar un producto de software de calidad, de acuerdo a las necesidades del negocio y las expectativas del cliente.

Scrumban sobre XP, adopta lo mejor de estos marcos de trabajo, por un lado, la agilidad o empirismo de Scrum y por otro lado el sistema de gestión de flujo de trabajo transparente de Kanban, todo ello encaminado por la colección de prácticas de ingeniería de software de XP.

El estudio comprende los siguientes objetivos: a) Aplicar el Scrumban a través de iteraciones sobre la programación extrema para ayudar a la gestión de proyectos de software en entornos ágiles para adoptar una estrategia de desarrollo incremental. b) Aplicar el Scrumban a través de una planificación bajo demanda sobre la programación extrema para ayudar a la gestión de proyectos de software en entornos ágiles para generar calidad en el resultado. c) Aplicar el Scrumban a través de la priorización sobre la programación extrema para ayudar a la gestión de proyectos de software en entornos ágiles para generar conocimiento en equipos auto organizados.

Capítulo I

Planteamiento de la Investigación

1.1. Diagnóstico y Enunciado del Problema

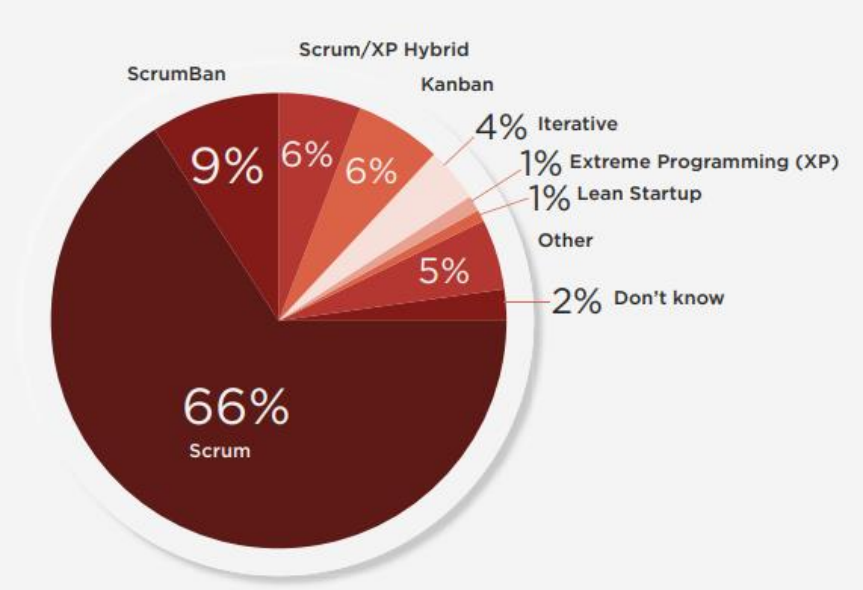
Al pasar de los tiempos, las Metodologías de Desarrollo de Software, se encuentran con proyectos de mayor demanda en los requerimientos con modificaciones cambiantes y tiempos de entrega de menor plazo, ante este suceso, no existe una adecuada metodología de desarrollo que se adapte idóneamente al proyecto en el marco de gestión de software, de tal forma , surgen las metodologías de desarrollo de software ágiles, reforzando la interacción entre el usuario y el equipo de trabajo, estableciendo entregas continuos del usuario (Zumba-Gamboa & León-Arreaga, 2018).

Según la plataforma (Digital.ai Software, 2021), publicó su 15° Informe sobre el estado de Agile. La encuesta destacó, en el tema de Técnicas Ágiles y Madurez.

> Una vez más a Scrum como el enfoque Ágil más popular con 66% identificándolo como la metodología que siguen más de cerca, con un 15% adicional que sigue derivaciones de Scrum (ScrumBan 9% y Scrum/XP 6%) tempranas y soportando la adaptación de los cambios.

Figura 1

¿Qué metodología Agile sigue más de cerca en el nivel del equipo?



Fuente: El 15.º Informe sobre el estado de Agile tiene derechos de autor de Digital.ai. Todos los derechos reservados.

Así también, cabe mencionar que Ken Schwaber, cocreador de Scrum anuncio lo siguiente:

> Estimo que el 75 % de las organizaciones que utilizan Scrum no lograrán obtener los beneficios que espera de él … Scrum es un marco muy simple dentro del cual se juega el "juego" del desarrollo de productos complejos. Scrum expone todas las deficiencias o disfunciones dentro de las prácticas de desarrollo de productos y sistemas de una organización. La intención de Scrum es hacerlos transparentes para que la organización pueda solucionarlos. Desafortunadamente, muchas organizaciones cambian Scrum para adaptarse a las deficiencias o disfunciones en lugar de resolverlas.

Si bien, Scrum es la metodología de desarrollo de software ágil más empleada por las organizaciones debido a su fácil comprensión y aplicación de sus principios y prácticas fundamentales, muchos desarrolladores fracasan en el intento porque es complicado de implementar ya que no cuentan con prácticas de desarrollo de productos.

Bajo los contextos previos, se plantea como una satisfactoria solución al problema identificado, en la parte de implementar lo mejor de cada metodología ágil, el marco de trabajo Scrumban y la metodología de Programación Extrema (XP) dan pase a la creación de un nuevo modelo para ayudar a la gestión de proyectos de software en entornos ágiles.

1.2. Formulación del Problema de Investigación

1.2.1. Problema principal

¿Cómo **Scrumban** sobre la programación extrema ayuda a la gestión de proyectos de **software en entornos ágiles**, 2022?

1.2.2. Problemas secundarios

a. ¿Cómo **Scrumban a través de iteraciones** sobre la programación extrema ayuda a la **gestión de proyectos de software** en entornos ágiles para adoptar una **estrategia de desarrollo incremental**, 2022?
b. ¿Cómo **Scrumban a través de una planificación bajo demanda** sobre la programación extrema ayuda a la **gestión de proyectos de software** en entornos ágiles para generar **calidad en el resultado**, 2022?
c. ¿Cómo **Scrumban a través de la priorización** sobre la programación extrema ayuda a la **gestión de proyectos de software** en entornos ágiles para generar **conocimiento en equipos auto organizados**, 2022?

1.3. Objetivos de la Investigación

1.3.1. Objetivo general

Utilizar Scrumban sobre la programación extrema para ayudar a la gestión de proyectos de software en entornos ágiles 2022.

1.3.2. Objetivos específicos

a. Aplicar **Scrumban a través de iteraciones** sobre la programación extrema para ayudar a la **gestión de proyectos de software** en entornos ágiles para adoptar una **estrategia de desarrollo incremental**, 2022.
b. Aplicar **Scrumban a través de una planificación bajo demanda sobre** la programación extrema para ayudar a la **gestión de proyectos de software** en entornos ágiles para generar **calidad en el resultado**, 2022.
c. Aplicar **Scrumban a través de la priorización** sobre programación extrema para ayudar a la **gestión de proyectos de software** en entornos ágiles para generar **conocimiento en equipos auto organizados**, 2022.

1.4. Hipótesis de la Investigación

En la "investigación cuantitativa" no siempre planteamos hipótesis. El alcance inicial del estudio, siendo un factor esencial, no depende de proponer una hipótesis. Si las investigaciones cuantitativas cuentan con una hipótesis, entonces el alcance inicial vendría a ser correlacional o explicativo, además las que también muestran un alcance descriptivo, pero deben pronosticar una cifra o un hecho (Hernández, Baptista y Collado, 2014, p. 104).

Las investigaciones de tipo descriptivo no necesitan plantear hipótesis, de tal manera que es suficiente con solo cuestionar estas partes de la investigación tales como, el planteamiento del

problema, de los objetivos y, por supuesto, del marco teórico que soporta el estudio (Bernal, 2010, p. 136).

No se realizará el planteamiento de hipótesis en esta investigación, debido a que no se pretende pronosticar una cifra o un hecho; al ser una investigación de tipo descriptivo, se logrará alcanzar los respectivos objetivos.

1.5. Justificación y Delimitación de la Investigación

1.5.1. Justificación

Teniendo las metodologías de desarrollo de software ágil, las organizaciones de proyectos de desarrollo de software se enfrentan a problemas en sus procesos de gestión y desarrollos de proyectos de software, como Scrum, que presenta limitaciones que no permite alteraciones en sus reglas, lo que conlleva a buscar soluciones. Kanban, puede proporcionar soluciones a algunas de los límites. Scrumban siendo híbrido de Scrum y Kanban, carece de artefactos, principios y buenas prácticas.

En la actualidad, las organizaciones de proyectos de desarrollo de software deben contar con una metodología que sea fácil de entender e implementar, así mismo, elegir la que mejor se adapte a la política del negocio; el modelo propuesto es la integración del marco de trabajo Scrumban y la metodología ágil (XP), para valorar la entrega a tiempo y la calidad en el software.

La presente investigación es viable, ya que se dispone de los recursos tecnológicos, económicos, fuentes de información y talento humano para llevar a cabo el modelo propuesto.

El trabajo de investigación será beneficioso para las organizaciones de proyectos de desarrollo de software, cliente, dueños del negocio, el gerente general, los inversionistas, etc.

Los beneficios que se obtendrían de este modelo propuesto serian: el cliente obtendría su proyecto en el plazo acordado, también contaría con un software de calidad; así mismo el trabajo resultaría disciplinado y comprometido.

1.5.2. Delimitación

La investigación se limitará en el análisis de la metodología de desarrollo de software ágil del marco de trabajo Scrumban sobre la metodología de la Programación Extrema (XP), aplicados a la gestión de proyectos de software, contemplando las diferentes etapas del ciclo de vida del proyecto. Scrumban con agilidad en la gestión de flujo de trabajo transparente y los artefactos, principios y buenas prácticas de XP.

Capítulo II

Marco de Referencia de la Investigación

2.1. Antecedentes de la Investigación

(Dias, 2019) en su investigación realizada para optar el grado de bachiller en Ingeniería de Sistemas Computacionales, identificó las diferentes metodologías existentes en el mercado, los cambios de las metodologías de gestión de proyectos de sistemas, la demanda en el mercado, teniendo como resultado una aceptación significativa sobre la metodología Scrum, liderando entre las demás en el mercado peruano, ya que se aprecia la entrega a tiempo y se adecua a los cambios adaptativos manteniendo el valor del producto. Por otro lado, se comprueba que Scrumban accede a realizar cambios en diferentes fases del proyecto y refuerza la retroalimentación del equipo de trabajo.

(Salvay, 2015) nos da a conocer en su estudio analizado, las diferentes metodologías de desarrollo ágiles, enfocados principalmente en Kanban y Scrumban, en el marco de proyectos TI (Information Technology), mediante la implementación de estas metodologías, se logra enfrentar los problemas que siempre surgen en la gestión y desarrollo de un proyecto de TI.

(Pérez Pérez, 2012) en su proyecto, concluyó que en un enfoque global sobre la agilidad e identificar las metodologías ágiles más empleadas; se muestra, que, compartiendo las mismas características y principios de agilidad, cada una de las metodologías ágiles, se enfoca en estudiar particularidades diferentes que tiene como resultado la adaptación de una mejor investigación de trabajo. Para tal sentido, creó una herramienta de solución en donde se identifica la metodología ágil óptima para la organización.

(Sepulveda Castaño, 2016) en su trabajo de investigación, plantea un modelo de aplicación de Scrumban para gestionar el proceso de generaciones de proyectos del I+D+I con el modelo Canvas; donde se concluye el análisis teórico que sustenta y justifica la investigación, el modelo se desarrolla de manera detallada sobre la implementación de la metodología ágil Scrumban para administrar la creación de proyectos según el modelo Canvas.

Gonzales R. & Perez L. (2015) en su proyecto, plantearon la implementación de las metodologías ágiles para proyectos software en el ámbito de las TI. Las metodologías seleccionadas para el estudio fueron: Scrum, eXtrema Programing, Crystal, Kanban y Scrumban.

Primero, se realizó una descripción y posterior análisis de estas metodologías ágiles para luego plantear una herramienta que sirva como apoyo para poder elegir la metodología ágil que más se ajusta y adapte a un proyecto o a una organización.

2.2. Marco Teórico

2.2.1. Metodologías ágiles

Las metodologías tradicionales son caracterizadas por ser rígido y con una cuantiosa documentación, el termino ágil surge como iniciativa por un grupo de expertos en el marco de desarrollo de software, teniendo como finalidad optimizar el proceso de creación de proyectos de software (Leiter & Sanchez, 2003).

Las metodologías ágiles brindan una solución puntual, lo más cercano a la medida para el proyecto dado. Las cualidades que más destaca es la sencillez que se aplica en el aprendizaje y la reducción de costos en la implementación del equipo de desarrollo. Logrando tener mayor aceptación en las metodologías ágiles.

2.2.2. Relación de scrumban con el pensamiento ágil

Según (Reddy, 2014) el manifiesto ágil describe cuatro principios básicos:

1. Individuos e iteraciones sobre procesos y herramientas
2. Software de trabajo sobre documentación completa
3. Colaboración con el cliente sobre la negociación del contrato
4. Responde al cambio sobre el siguiente plan

Claramente, Scrum incorpora estos principios de manera bastante efectiva.

Prescribe periodos cortos de entrega de software incremental. Sus circuitos de retroalimentación incorporados están diseñados para facilitar un enfoque de "inspección y adaptación" en lugar de recopilar requisitos. Prescribe la participación del cliente a nivel de equipo (Reddy, 2014, pág. 50).

En muchos contextos de desarrollo de software, el marco de Scrum no proporciona una guía específica sobre cosas como la interoperabilidad entre equipos de desarrollo, y excluye intencionalmente cualquier marco formal para administrar dependencias de recursos externos (Reddy, 2014, pág. 50).

Scrumban ayuda con tales desafíos debido a su énfasis en optimizar el flujo de trabajo a través de un proceso Scrum existente. Cataliza la maduración de las características clave de Scrum/ Agile en lugar de introducir algo nuevo o diferente (Reddy, 2014, pág. 50).

2.2.3. Scrumban sobre programación extrema

El marco de trabajo Scrumban y la metodología de desarrollo ágil XP se complementan entre sí. Por un lado Scrumban, siendo la integración de Scrum y Kanban, el primero es un marco de trabajo basado en lo interactivo e incremental aplicado en el desarrollo de proyectos

además de la estructura de los sprint que duran entre 2 y 4 semanas y el segundo es un camino esvelto para el desarrollo ágil de software, siendo un método donde se usa tarjetas como símbolos visuales para desencadenar y controlar el flujo a través de un proceso de producción; son en conjunto la integración completa para desarrollar proyectos en entornos ágiles. Del otro lado XP, aporta una colección de prácticas de ingeniería de software que ayudan a resolver las dificultades a lo largo del desarrollo del proyecto.(Bougroun et al., 2016).

2.2.3.1. Iteraciones. Periodos de tiempos que contienen diferentes actividades para ser completadas en cada una de esas repeticiones (Flores Leyva & Molina Espinoza, 2014).

Ajay Reddy (2014) nos presenta en su libro sobre la longitud de iteraciones; muchas organizaciones buscan establecer longitudes de iteración uniformes porque creen erróneamente que este es un buen enfoque para alinear diferentes sistemas con la misma cadencia. Sin embargo, durante muchos años, el consenso general entre los practicantes de Scrum ha sido recomendar iteraciones de dos semanas.

Las iteraciones representan los cambios adicionales surgidos mediante los procesos periódicos de prueba y retroalimentación que en diferentes circunstancias dirigen a un sistema firme pero en cambios repentinos (Kendall & Kendall, 2011).

2.2.3.2. Planificación bajo demanda. La planificación se realiza según las necesidades del cliente donde se determina las Historias de usuarios y tareas que se completarán en la próxima iteración, aportando flexibilidad y un cierto ahorro de tiempo (Flores Leyva & Molina Espinoza, 2014).

2.2.3.3. Priorización. Cuando el cliente elije la prioridad de las funciones, el equipo de trabajo entiende el objetivo del negocio y ofrece al cliente el mejor valor comercial.(Paul & Rahman, 2018).

2.2.4. Gestión de proyectos de software en entornos ágiles

Son métodos de ingeniería del software basados en el desarrollo iterativo e incremental, además, están basados en valores, principios y prácticas básicas consiguiendo terminar a tiempo (Kendall & Kendall, 2011).

La gestión de proyectos es un conjunto de esfuerzos asociados, en un determinado tiempo, donde sigue un objetivo definido y presentan conocimientos específicos y recursos. Por otro lado, es una organización transitoria con el objetivo de alcanzar una finalidad significativa. Se puede decir que cuando se alcanzan los objetivos es el monto en que el proyecto se ha concluido. La incorporación de diversos elementos que actúan en un proyecto marca la diferencia de éste (Bedini Gonzáles & Guerre Genshowsky, 2005).

La gestión de software ágil, es un conjunto de estrategias para implementar productos software encaminados a la entrega a tiempo de los productos tangibles, así como también optimizar la respuesta ágil y flexible a los requerimientos impuestos en el mercado que se encuentra en evolución. Un proyecto de software da inicio con una visión superficial del cliente, donde en circunstancias desconoce del proyecto concluido porque el ritmo del trabajo es constante en la innovación y la velocidad que produce el negocio, dando como resultado una constante evolución permitiendo mejorar la visión inicial del software, por lo tanto, la formulación de la gestión de proyectos ágil, no se plantea a la necesidad de anticipación, sino en base a la adaptación continua de los requerimientos del cliente (Salazar et al., 2018).

2.2.4.1. Estrategia de desarrollo incremental. Esto se refiere a insertar nuevas funciones al código que se está desarrollando (Canty, 2015).

Para Salyay (2015), el desarrollo incremental consiste en el crecimiento del producto, este se entrega al cliente incrementalmente sobre un periodo de tiempo planificado.(Salvay, 2015)

Un proceso de desarrollo incremental consiste en la creación del software y su entrega próxima mediante piezas o aumentos, que es parte del proyecto totalitario (Cohn, 2009).

2.2.4.2. Calidad del resultado. "Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan"(Pressman, 2002).

En su libro (Pressman, 2002) menciona la calidad de software como "el resultado de la buena administración del proyecto y de una correcta práctica de la ingeniería de software". Estos factores se enmarcan en cuatro actividades primordiales que guían al equipo de desarrollo a alcanzar la calidad del proyecto: Métodos de la ingeniería de software, técnicas de administración de proyectos y acciones de control de calidad.

2.2.4.3. Conocimiento de equipo autoorganizado. Son equipos disciplinados con la experiencia requerida que trabajan con reglas explícitas que buscan desempeñarse satisfactoriamente para lograr el objetivo del proyecto (Z. A. Khan, 2014).

El trabajo que se emplea está sujeta a reglas explícitas, es decir que todos los integrantes del equipo puedan atribuirse actividades libremente para alcanzar un flujo de labores más dispuestos y eficiente. Estás reglas son beneficiosos ante los casos frecuentes donde se necesitan tomar decisiones de una situación particular a lo largo del desarrollo del proyecto (Z. A. Khan, 2014).

2.2.5. Programación extrema (XP)

Esta metodología ágil de trabajo, es indudablemente la más reconocida e implementada. El término fue incorporado por Beck (2000) puesto que el enfoque está enmarcado en la implementación de reconocidas buenas prácticas, así mismo, como el desarrollo iterativo, y la involucración del cliente en niveles "extremos". En XP, todos los requerimientos del cliente se enuncian como situaciones (denominados historia de usuarios), este artefacto se implementa claramente de tareas. Los principios más destacados de esta metodología es la programación en parejas y el enfoque de que antes de desarrollar código se ejecute las pruebas de cada tarea. Así mismo, las pruebas en su totalidad deben tener un resultado satisfactorio para integrar el código nuevo al software (Borja López et al., 2005, p.364).

La programación extrema en el marco del enfoque “extremo”, admite esta característica para el desarrollo iterativo. Al realizarse constantemente cambios imprevistos en los proyectos de software, surge un problema de degradación en la estructura del software. Ante este problema, la programación extrema, mediante la refactorización constante del software, es decir, el equipo de programación investiga posibles mejoras del software y las implementa inmediatamente, para que de esta manera el software sea fácil de entender y cambiar cuando se efectúen nuevas historias.

Valores de la programación extrema.

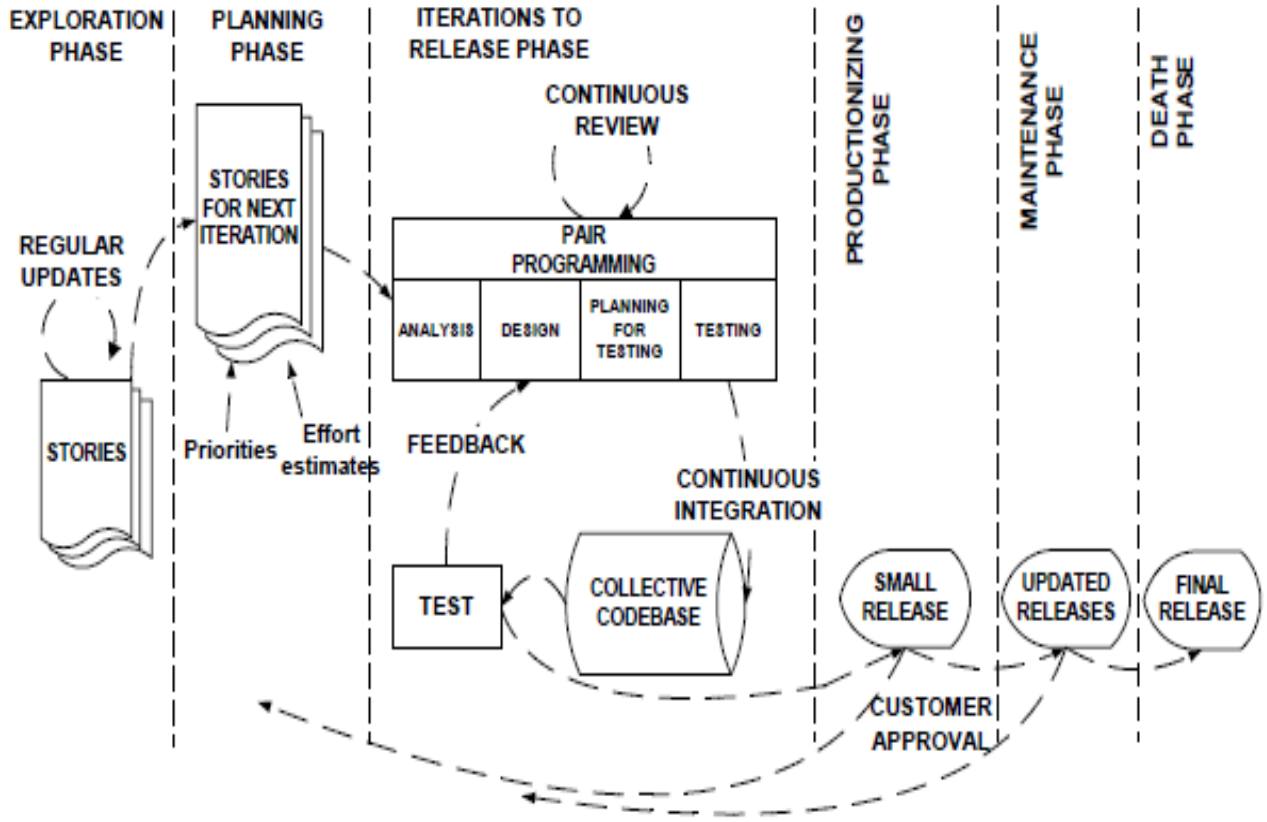
Según (Beck et al., 2000), determina cinco valores donde se fundamentan todo el trabajo que realiza el equipo de desarrollo de XP. Estos valores son sumamente esenciales para esta metodología que encontramos durante todo el desarrollo de XP.

a. **Comunicación.** Esta interacción se da entre el equipo de desarrollo y otros participantes interesados, XP pone en práctica el uso de las metáforas para la comunicación y conceptos principales, así mismo, ayuda en la retroalimentación continua y evita el exceso de documentación como medio de documentación, de esta manera, enfatiza la colaboración cercana pero informal (verbal) entre los clientes y los desarrolladores.
b. **Simplicidad.** Es enfocarse en las necesidades inmediatas, para que los desarrolladores consideren las tareas principales en lugar de las del futuro. Tiene como objetivo crear un diseño sencillo de manera que sea fácil de implementar en el desarrollo del código. Si se presenta la situación de mejorar el diseño implementado, se rediseñará posteriormente.
c. **Realimentación.** Nace de tres fuentes: de la implementación del software, el cliente y otros miembros del equipo de software. En la etapa del diseño e implementación se realizá una estrategia de pruebas eficaz, donde el software (por medio de los resultados de las pruebas) da retroalimentación al equipo ágil. XP usa la prueba unitaria como su táctica principal de pruebas.
d. **Coraje.** Consiste en que los desarrolladores estén encaminados al cambio, un cambio inevitable que es posible al estar preparado con una metodología donde ayuda a ese cambio. Está basado en realizar las actividades para hoy y no para mañana (Borja López, 2013).
e. **Respeto**. El equipo debe seguir un trabajo unificado, sin realizar decisiones repentinas. La metodología Programación Extrema concientiza el trabajo en conjunto. Todos los roles presentes en la metodología (desarrolladores, cliente, etc.) conforman una parte fundamental del equipo, teniendo como resultado un esfuerzo conjunto enfocado en desarrollar software de calidad (Borja López, 2013).

Ciclo de Vida de la Programación Extrema

Figura 2

Ciclo de Vida de la Programación Extrema



Fuente: Adaptado de Beck (2005).

a. **Fase de exploración.** En esta fase, inicia con la descripción de forma general de las historias de usuario planteados por el cliente, donde son de utilidad para la primera entrega del producto. De la misma manera, los programadores reconocen y determinan las herramientas, tecnologías y prácticas para ejecutar el proyecto. Se realizan pruebas

respectivas de conjunto de técnicas y se investigan los medios de la arquitectura del sistema, con el fin de construir un prototipo. El tiempo de duración de esta fase, se da entre pocas semanas y pocos meses, dependiendo del manejo de las tecnologías que tenga los programadores (Borja López, 2013).

b. **Fase de planificación.** En este evento, el cliente elige las historias de usuario más relevantes de acuerdo a su necesidad, y paralelamente, los desarrolladores asignan un valor de esfuerzo a cada tarea. Además, se establecen acuerdos sobre el contenido de la primera entrega y se determina un cronograma de todo el proyecto con el cliente. El tiempo estimado para cada entrega debería durar unos pocos días, no más de tres meses (Borja López, 2013).

c. **Fase de iteraciones y lanzamientos**.En esta fase se realizan diversas iteraciones conforme al software para que posteriormente sea entregado al cliente. Así mismo, se da el Plan de Entrega, donde está comprendido por un sprint no mayor de tres semanas. Generalmente, dada la primera iteración, se puede desear alcanzar formar una arquitectura del proyecto para que sirva como modelo en toda la ejecución del proyecto. Para lograr esto, se debe seleccionar los requerimientos que obliguen a determinar esta arquitectura, sin embargo, esto solo se ve en teoría, puesto que el usuario tiene la última palabra respecto a la implementación estimada para cada iteración, de esta manera se maximiza el valor del negocio. Finalmente, en la última iteración se obtiene un sistema preparado para ejecutarse en producción.

d. **Fase de Producción.** El evento está integrado por pruebas añadidas y de rendimiento para que posteriormente el sistema sea presentado al cliente. De la misma manera, se

deben determinar nuevas características para ser incluidas al último debido a los cambios repentinos que se pueda presentar durante esta fase.

e. **Fase de mantenimiento.** En esta fase, mientras la primera versión se halla en la fase de producción, XP tiene que conservar el sistema en marcha, de la misma manera debe desarrollar iteraciones nuevas. Para efectuar este trabajo se necesita de tareas de soporte para el cliente. Donde esto conlleva a un pausado desarrollo ante una implementación del sistema en la fase de producción. Este evento consigue pretender la inserción de personal adicional al equipo, además de presentar variaciones en su estructura (Borja López, 2013).

f. **Fase de muerte.** En esta fase, al cliente ya no se le permite añadir más historias al sistema debido a que se ha realizado de manera satisfactoria todas las peticiones del usuario; en este evento el equipo de desarrollo se enfoca exclusivamente en el funcionamiento y credibilidad del sistema. La documentación del sistema culmina en esta fase donde la arquitectura es invariable. Por otro lado, el proyecto llega a su fin cuando no hay beneficios suficientes para satisfacer al cliente o no hay financiamiento para su mantenimiento (Borja López, 2013).

Roles de la programación

Según Beck & Andreas (2004) la metodología XP está encaminada en 7 roles que se presentan a continuación:

a. **Cliente.** El cliente, para validar la implementación del proyecto, describe sus requerimientos y las pruebas funcionales. Además, debe seleccionar los requerimientos que le parezcan más importantes y determinar cuáles de estas se implementarán en cada iteración, en ese sentido, el cliente se enfoca en contribuir un importante valor al negocio.

b. **Encargado de pruebas (Tester).** Este rol apoya al cliente en escribir las pruebas funcionales del sistema. Además, efectúa las pruebas constantemente, es el encargado de informar los resultados al equipo de trabajo y está comprometido con las herramientas de soporte que se desarrollarán en las pruebas.

c. **Encargado de seguimiento (Tracker).** El responsable del seguimiento de desarrollo del proceso XP, brinda retroalimentación al equipo. Su función es constatar el grado de veracidad entre las evaluaciones ejecutadas y el periodo verdadero trabajado, para posteriormente comunicar los resultados y tomar decisiones en optimar las estimaciones posteriores. Además, es el encargado de supervisar el avance de cada iteración y evaluar el alcance de los objetivos mediante las variaciones de periodo y recursos.

d. **Entrenador (Coach).** Es el encargado del proceso general. El entrenador debe conocer a detalle todo el proceso de XP, para que de esta manera pueda proporcionar direccionamiento a los integrantes del equipo, efectuándose la aplicación de las prácticas XP y asegurar que el transcurso del desarrollo se ejecute sin problemas.

e. **Consultor.** Es un miembro externo del equipo, donde es responsable de guiar al equipo de trabajo para solucionar problemas en específico, es especializado en un tema en específico necesario para el proyecto.

f. **Gestor (Big boss).** Es el intermediario entre quienes usarán el sistema y el equipo desarrollador, por el cual, ayuda a entender al equipo lo que el cliente tiene en mente, para que de esta manera se efectúe las situaciones adecuadas. Siendo de mayor importancia la relación de coordinación.

Artefactos de XP

a. **Historias de usuario.** Es una técnica utilizada en la metodología XP en donde se especifica los requisitos del software mediante la descripción breve de las características que realiza el cliente. Este artefacto consiste de tarjetas de papel e indica lo que el sistema debe poseer, además consta de los requisitos funcionales y no funcionales (Borja López, 2013).

Figura 3

Plantilla para las Historias de Usuario

HISTORIA DEL USUARIO	
Número:	***Usuario:***
Nombre de historia:	
Prioridad en negocio:	***Riesgo de desarrollo:***
Puntos estimados:	***Iteración asignada:***
Programador responsable:	
Descripción: .	
Observación:	

Fuente: Fuente: Universidad Politécnica de Valencia (2006).

Tareas de ingeniería. Para Borja López (2013), la historia de usuario está compuesta por diferentes tareas de ingeniería, el cual especifica la descripción de actividades a realizar, y que estas son dirigidas al desarrollador, puesto que ayuda a facilitar la interpretación para posteriormente plasmarlo en código.

Figura 4

Plantilla para las Tareas de Ingenierías

TAREA DE INGENIERÍA	
Número de tarea de ingeniería:	*Número de historia de usuario:*
Nombre de tarea:	
Tipo de tarea:	*Puntos estimados:*
Fecha de inicio:	*Fecha fin:*
Programador responsable:	
Descripción:	

Fuente: Universidad Politécnica de Valencia (2006).

b. **Pruebas de aceptación.** Se desprenden de las historias de usuario, donde se enfocan en realizar distintas pruebas en cada ciclo de la iteración en desarrollo. El cliente tiene que simular diferentes situaciones para verificar que la historia fue aplicada satisfactoriamente (Borja López, 2013).

Figura 5

Plantilla para las Pruebas de Aceptación.

CASO PRUEBA DE ACEPTACIÓN	
Código:	*Número de historia de usuario:*
Nombre de la prueba:	
Descripción:	
Condiciones de ejecución: ninguna	
Entrada:	
Resultado esperado:	
Evaluación esperada:	

Fuente: Universidad Politécnica de Valencia (2006).

c. **Tarjetas CRC (Clase-Responsabilidad-Colaboración).** Estas se subdividen en tres campos importantes que sujetan información, en el campo clase se indica el nombre de forma puntual seguido de las responsabilidades efectuadas y, por último, la determinación de la colaboración. En el enfoque de la programación orientada a objetos, el campo clase está relacionada con el objeto, el campo de la responsabilidad define lo que ha de hacer el sistema y la colaboración significa cómo se comunica entre las demás clases (Beck, 1999).

Prácticas esenciales de XP

La Programación Extrema o XP concierne al grupo de las metodologías ágiles, enfocado al desarrollo de sistemas mediante buenas prácticas para llevarla al extremo (Kendall & Kendall, 2011).

a. **Entregas limitadas o pequeñas**: Consiste en " solucionar las características principales a tiempo, por lo tanto se reduce el tiempo de entrega, donde próximamente el proyecto será mejorado"(Kendall & Kendall, 2011, p. 171).
b. **Semana de trabajo de 40 horas:** Esta práctica permite "motivar a los miembros del equipo para que trabajen en forma intensa y después se tomen tiempo de descanso" (Kendall & Kendall, 2011, p. 171).
c. **Cliente en el sitio:** Consiste en contar con la presencia del usuario experto en el negocio constante para guiar el trabajo durante el proceso de desarrollo (Kendall & Kendall, 2011, p. 171).
d. **Programación en pareja:** Ésta práctica consiste en trabajar con el programador elegido para así desarrollar la codificación y realizar las pruebas (Kendall & Kendall, 2011, p. 171).

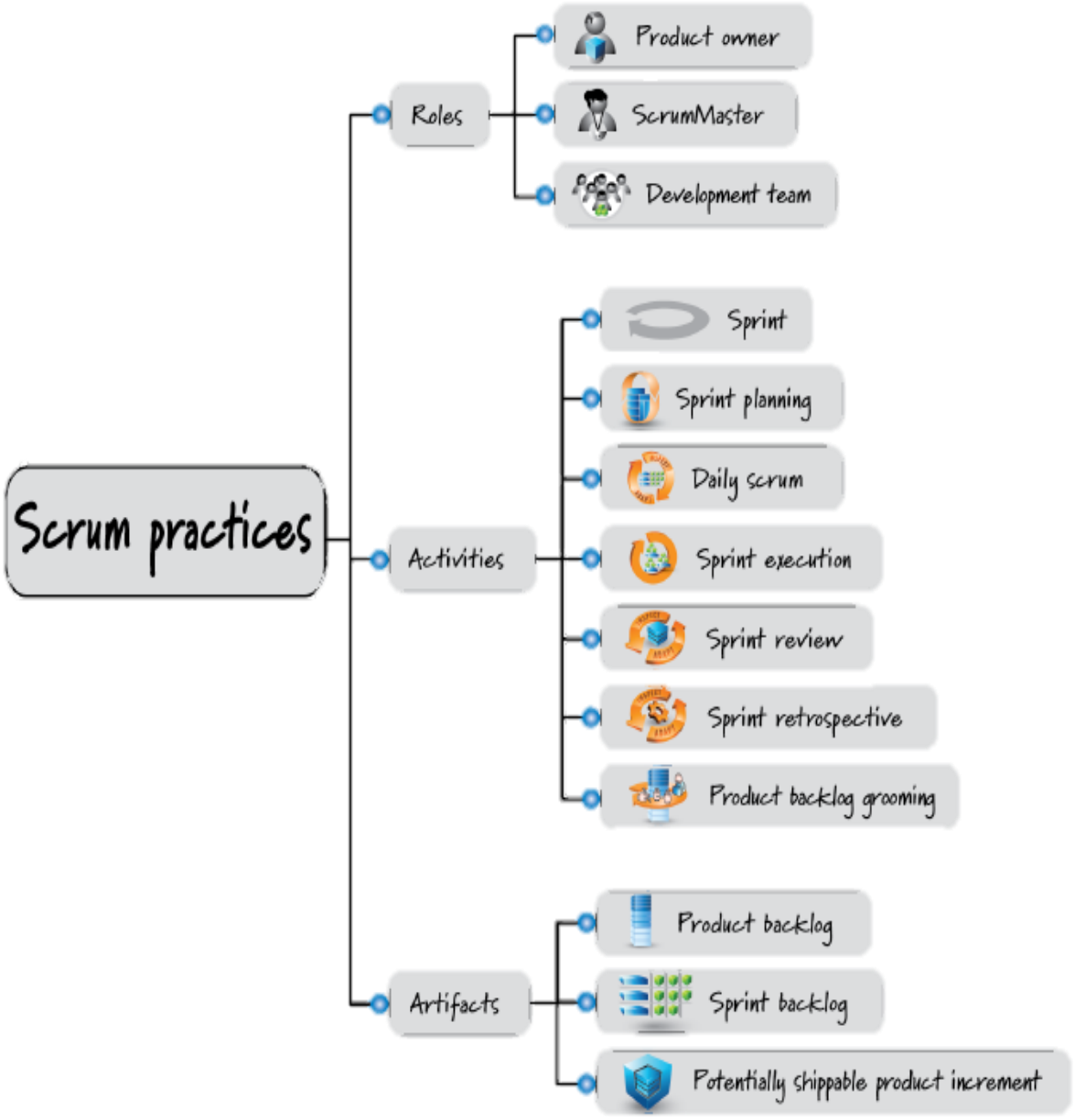
2.2.6. Scrum

Es el enfoque de desarrollo de software ágil más utilizado desde los últimos 10 años. Scrum está basado en la teoría del manejo de técnicas empíricos o empirismo. Cuando adquirimos los saberes mediante la rutina y además tomamos elecciones de los previsto, entonces nos estamos refiriendo al método del empirismo. En esta metodología se utiliza el enfoque iterativo e incremental para mejorar la previsibilidad y manejar el riesgo (Schwaber Ken & Sutherland Jeff, 2017).

El trabajo en equipo conlleva a mejorar obtener mejores resultados “Scrum está diseñado para maximizar la capacidad del equipo hacia la productividad y calidad mejorada. Takeuchi y Nonaka anunciaron la palabra Scrum por primera vez (1986).A partir de la estrategia del juego de Rugby, definieron la ideología central de Scrum” (Bhavsar et al., 2020).

Figura 6

Prácticas de Scrum.



Fuente: Adaptado de Rubín (2013).

Valores de Scrum

Ladas (2008), menciona que, debido a su énfasis en fomentar la agilidad, el trabajo en equipo y la mejora continua, el conjunto central de valores de Scrum está centrado en el equipo (pág. 64).

a. **Atención**. Al centrarse en solo unas pocas cosas a la vez (la acumulación de sprints), los equipos trabajarán bien juntos, producirán un trabajo de mayor calidad y entregarán elementos valiosos antes.
b. **Coraje.** Un sentido de propiedad del equipo (en lugar de propiedad individual) significa que las personas no están solas. Se sienten apoyados y tienen más recursos a su disposición, lo que les da valor para enfrentar mayores retos.
c. **Franqueza.** A medida que los equipos trabajan juntos, practican como expresar como lo están haciendo y que se interpone en su camino. Aprenden a expresar preocupaciones que permite abordar los problemas.
d. **Compromiso.** Debido a que los equipos tienen un mayor control sobre su propio destino, se comprometen más con el éxito.
e. **El respeto.** Con el transcurso en que se desarrolla el trabajo en equipo, se van compartiendo tanto el éxito como el fracaso, lo cual conlleva a apoyarse mutuamente respetándose unos a otros.

Marco de trabajo de scrum

Scrum surge de la investigación de Ken Schwaber, Jeff Sutherland y Mike Beedle, donde se determina como un marco de trabajo para la gestión de proyectos dirigido esencialmente a las personas y al equipo de desarrollo que elaboran el proyecto. Su finalidad es obtener productos

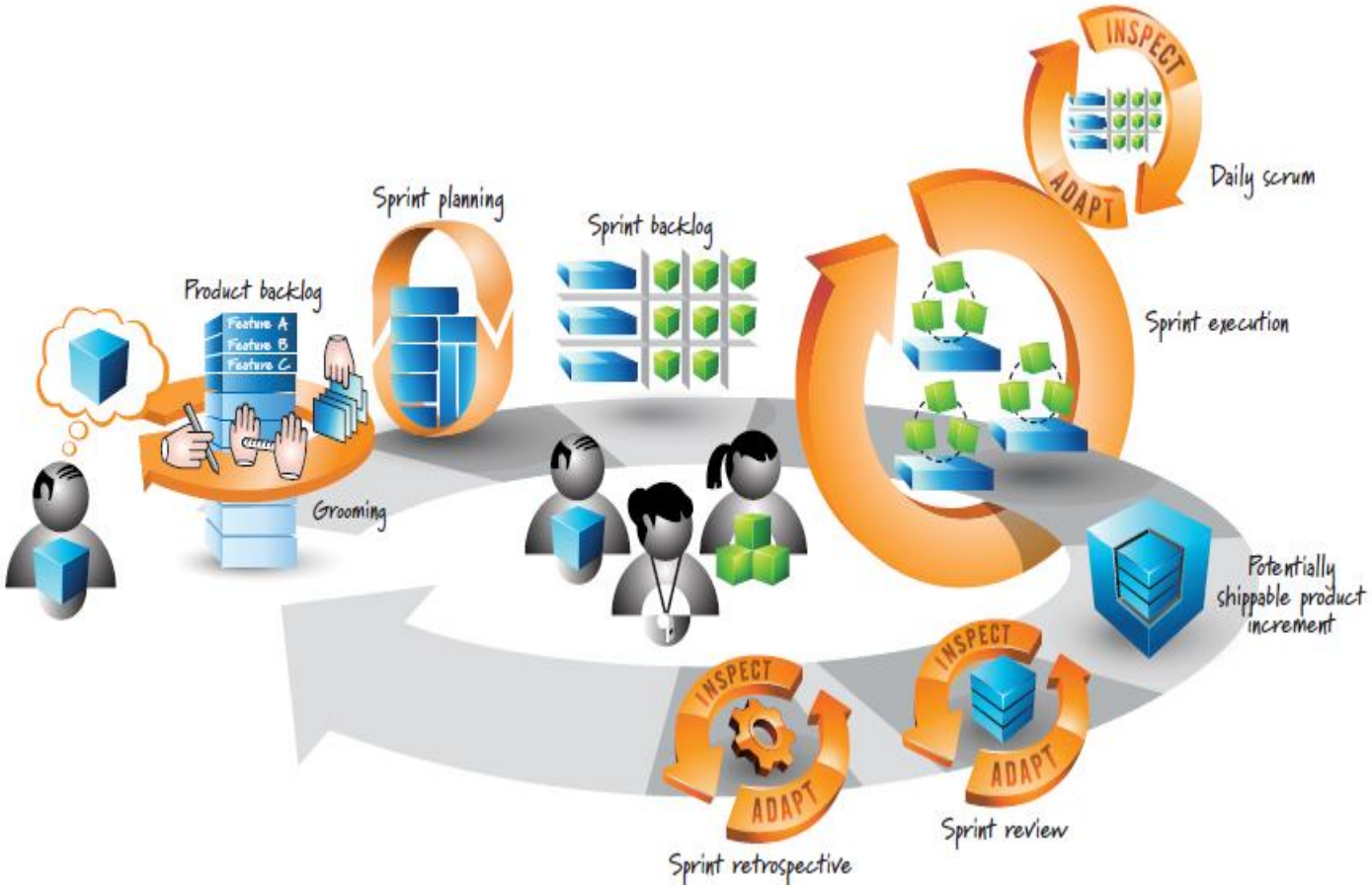
complejos y sofisticados a base del trabajo en equipo y de manera eficiente. Scrum como una ingeniería social, desea alcanzar la conformidad de todos los involucrados en el proceso, promoviendo la colaboración a través de la autoorganización.

Este marco de trabajo está enfocado especialmente para proyectos con un rápido cambio de requisitos. Sus principales características son, primero el desarrollo de software se efectúa mediante iteraciones, nombradas sprints, con 30 días de duración; cada sprint muestra como resultado un incremento ejecutable que posteriormente se muestra al cliente, como segunda característica importante es la realización de las reuniones sobre el transcurso del proyecto, entre ellas se destaca la reunión diaria con un tiempo de 15 minutos para la coordinación e integración (Letelier & Sanchez, 2003).

Scrum abarca problemas complejos adaptativos, a su vez se entregan productos óptimos de forma eficiente y creativa con el máximo valor. Este modelo permite a una organización la división de pequeños equipos autoorganizados con tamaños que van de 4 a 10 personas. De acuerdo con la práctica de scrum, un equipo de scrum debe ser autoorganizado y multifuncional, y debe tener todas las competencias necesarias para realizar el proyecto sin la necesidad de competencias externas (Schwaber Ken & Sutherland Jeff, 2017).

Figura 7

Scrum Framework



Fuente: Essential Scrum (p. 17), por K.S. Rubin, 2013, Addison-Wesley

El marco de Scrum constituye equipos de scrum y sus roles además de eventos, artefactos y reglas asociados. Así mismo, Scrum establece cuatro eventos formales: Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective. Además, el equipo Scrum consta de actores capacitados un propietario de producto, el equipo de desarrollo y un Scrum Master (Schwaber Ken & Sutherland Jeff, 2017).

El propietario del producto que es encargado de cada equipo, es responsable de la creación de las historias de usuario que están basadas en los requisitos de alto nivel del cliente / negocio, lo que genera una cola de acumulación de productos; siendo una lista de acumulación

dinámica de requisitos, además es la única fuente que almacena los requisitos manteniéndose durante todo el ciclo de vida del desarrollo del producto. El responsable de la administración de la acumulación del producto, es el propietario del producto, que se encarga de especificar cada elemento de la acumulación del producto, priorizar los elementos, garantizar que la acumulación sea clara para el equipo de desarrollo y dirigir al equipo de scrum sobre qué trabajar a continuación (Schwaber Ken & Sutherland Jeff, 2013).

En el modelo Scrum, el Scrum master es la persona que facilita el desarrollo de productos mediante pequeñas iteraciones llamadas sprints, además que asegura que el equipo de scrum aplique los valores y prácticas scrum. La duración de iteración de sprint suele durar entre 2 y 4 semanas; sin embargo, es determinada por la organización. Al iniciar la planificación del Sprint los miembros del equipo intentan estimar/emparejar el esfuerzo necesario para las historias en la acumulación.

Al final de cada sprint, se lleva a cabo una reunión de revisión del sprint para inspeccionar los objetivos del sprint y adaptar en consecuencia la acumulación de productos, en esta reunión participa el equipo scrum, el propietario del producto y los propietarios clave invitados por el propietario del producto. Los miembros del equipo demuestran las historias completadas al propietario del producto y a las partes interesadas clave y, en consecuencia, responden a las preguntas relacionadas con el incremento del sprint. Por lo general, el resultado de la Revisión del Sprint es un Product Backlog revisado que define los elementos probables del Product Backlog para el próximo Sprint (Schwaber Ken & Sutherland Jeff, 2013).

La reunión Sprint Review se mantiene intencionalmente informal para que no se convierta en una carga para los miembros del equipo, sino que será el resultado natural de un sprint. Hay circunstancias en las que los sprints pueden cancelarse todos juntos y, en tal caso, se

volverán a estimar los elementos de la pila de producto que se encuentren incompletos para que posteriormente sean nuevamente insertados y los que ya están completados se revisan y marcan como hechos (Schwaber Ken & Sutherland Jeff, 2017).

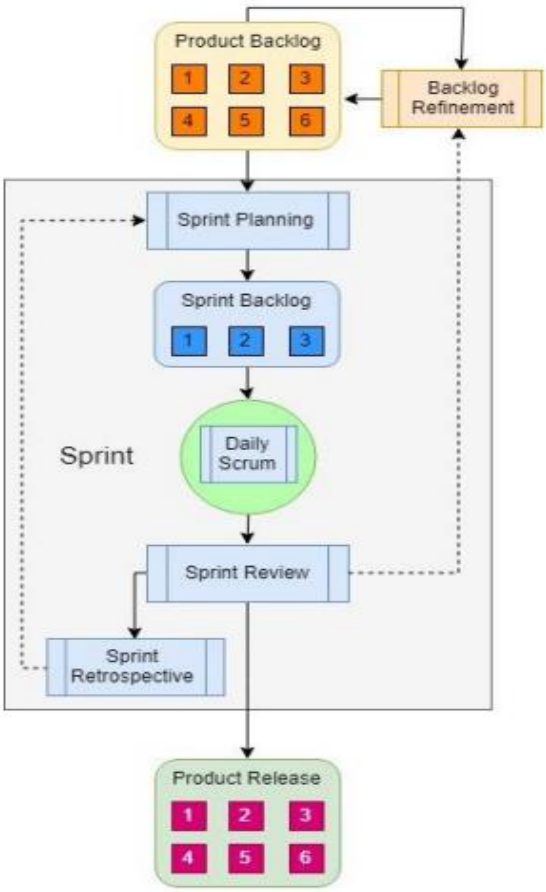
Después de finalizar un sprint, generalmente se hace una reunión llamada Sprint Retrospectiva, con el propósito buscar oportunidades de mejora y retrospectivamente de cómo trabajar mejor en el futuro, en esta reunión los miembros del equipo reflexionan sobre cómo fueron las cosas en el último sprint para que puedan mejorar las deficiencias y continuar haciendo cosas buenas en el próximo sprint. Según Schwaber & Sutherland (2013).

Para (Bhavsar et al., 2020), Scrum ha demostrado ser fascinante entre todos los enfoques ágiles, para la Gestión de la Ingeniería de Software además se construye bajo el techo de los principios ágiles y sobre todo la teoría del control del empirismo, define las características claves de Scrum como los artefactos, valores, pilares, roles y eventos. Scrum es un enfoque ágil iterativo e incremental basado en la teoría de control de procesos utilizando una longitud flexible (duración) del proyecto de una semana a un mes con el objetivo de lograr DOD (definición de terminado) como resultado del Sprint. La figura 8 representa el flujo de scrum interactuando con algunos de sus artefactos. El propietario del producto gestiona los elementos de la cartera de productos y el refinamiento. Un Sprint consta de cuatro eventos principales: Sprint Planning, Daily Scrum, Sprint review y Sprint Retrospective. Los elementos de la cartera de productos (PBI) se introducen en la cartera de pedidos de Sprint (SB) durante la planificación del Sprint para posteriormente los miembros del equipo de desarrollo lo ejecuten y discutan las actualizaciones durante las reuniones diarias de Scrum. El Scrum Máster se encarga de los impedimentos si alguno ocurre durante el sprint. Los elementos de la cartera de pedidos desarrollados son inspeccionados durante la Revisión de Sprint por el Equipo Scrum y las partes

interesadas externas. Los elementos de trabajo que cumplen con DOD (Definición de Terminado) se consideran para su lanzamiento como un incremento de producto. Los ciclos de Sprint continúan hasta el final del desarrollo del producto.

Figura 8

Flujo de Scrum



Nota. El flujo de Scrum junto con algunos artefactos.

Fuente: International Journal of Innovative Technology and Exploring Engineering (1627), por Bhavsar, K., Shah,V. y Gopalan, S., 2020, Blue Eyes Intelligence Engineering & Sciences Publication.

En resumen, el modelo Scrum proporciona un marco y herramientas de trabajo para el desarrollo de software y se ejecuta mediante pequeñas iteraciones llamadas Sprint. Todo el equipo se encuentra capacitado y comprometido demostrándolo al final de un sprint y, al mismo tiempo, el equipo trata de mejorar constantemente asumiendo los aprendizajes de un sprint a otro. Además, este modelo define ciertos roles para facilitar y gestionar el trabajo en equipo (Bhavsar et al., 2020).

2.2.7. Kanban

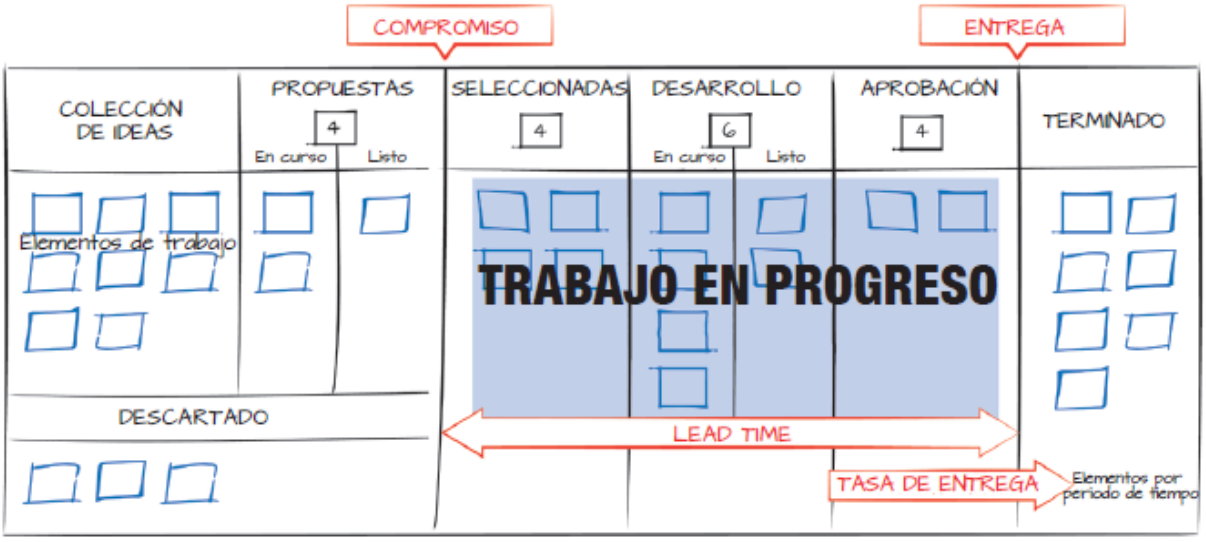
El modelo Kanban se origina del sistema de producción de Toyota. Kanban es una palabra japonesa que significa "letrero". Este modelo se utilizó en el sistema de control de producción con un enfoque de producción justo a tiempo y sacando la mejor capacidad de los trabajadores (Sugimori, 1977).

El Sistema de Kanban promueve a los equipos de los proyectos a visualizar el flujo de trabajo, limitar el trabajo en progreso (WIP) en la etapa del flujo de trabajo y medir el tiempo del ciclo (es el tiempo promedio para completar las tareas) (Henrik Kniberg, 2009). En el modelo Kanban, el flujo de trabajo en el proyecto de desarrollo de software se visualiza utilizando un tablero llamado Tablero Kanban. El tablero de Kanban, los elementos de trabajo (generalmente las historias de usuario) se representan mediante tarjetas de Kanban, siendo las más utilizadas como notas adhesivas. Este tablero consta de columnas que representan cada etapa del flujo de trabajo del proceso de desarrollo. Para una mejor gestión del flujo de trabajo, en cada columna se limita el número de elementos. Como resultado, los desarrolladores se enfocan en los elementos que están en curso e intentan completar las anteriores y comenzar a trabajar en nuevos elementos de trabajo. Siguiendo el proceso, cuando un elemento de trabajo se completa en una etapa, se

mueve a la siguiente columna y como resultado, se puede extraer otro elemento de trabajo de la columna anterior. A continuación, se muestra una simple representación del tablero de Kanban.

Figura 9

Tablero de Kanban



Fuente: Kanban Escencial Condensado (p.11), por Anderson D. & Carmichael A., 2016, LeanKanban

Valores de Kanban.

De acuerdo a Anderson & Carmichael (2016) el método Kanban está guiado por los siguientes valores:

a. **Transparencia.** El flujo de trabajo, los estados de trabajo, los parámetros, las políticas y las restricciones deben estar visibles. La visualización mejora la comprensión, nos permite ver patrones y mejora nuestra toma de decisiones (Ladas, 2008).
b. **Equilibrio.** Es la capacidad de comprender que los diferentes aspectos, puntos de vista y capacidades deben mantener un equilibrio para alcanzar la efectividad.

c. **Colaboración.** El Método Kanban está enfocado en trabajar juntos de manera que la colaboración está en su corazón.
d. **Foco en el cliente.** Una preocupación centrada en el cliente significa ir más allá de una perspectiva centrada en la actividad (“tarea completada”) o centrada en el producto (“producto potencialmente entregable”).
e. **Flujo.** Las personas son más felices y productivas cuando están en un estado de fluidez.
f. **Liderazgo.** Un buen liderazgo y una buena gestión crean las condiciones en las que prospera la autoorganización.
g. **Entendimiento.** El proceso que no puede defenderse es una señal segura de que las personas que la utilizan han olvidado lo que quieren y creen.
h. **Acuerdo.** Sin un acuerdo no se puede lograr un desempeño significativo y sostenible.
i. **Respeto.** Kanban aplica el valor del respeto de formas ligeramente diferentes, comenzando con el principio de respetar los roles y formas de trabajo actuales.

Características fundamentales

Según Pérez (2016), determina las características fundamentales que representan la fortaleza del método de trabajo Kanban.

a. **Requiere poco aprendizaje.** El aprendizaje se simplifica en aprender las reglas propias del equipo ya que en Kanban las políticas del proceso son explícitas para generar valor.
b. **Simple construcción.** Está conformado por tarjetas incluidas en columnas que representan el proceso del desarrollo de proyecto, para ello su construcción sólo es necesario de papel y lápiz (Kniberg & Skarin, 2010).
c. **Baja inversión inicial y operacional.** Construir un tablero Kanban resulta ser muy barato porque se puede construir con materiales que se encuentran en cualquier oficina.

Para operar se necesita un lápiz y papeles de colores (post-it), que son de bajo costo en el mercado.

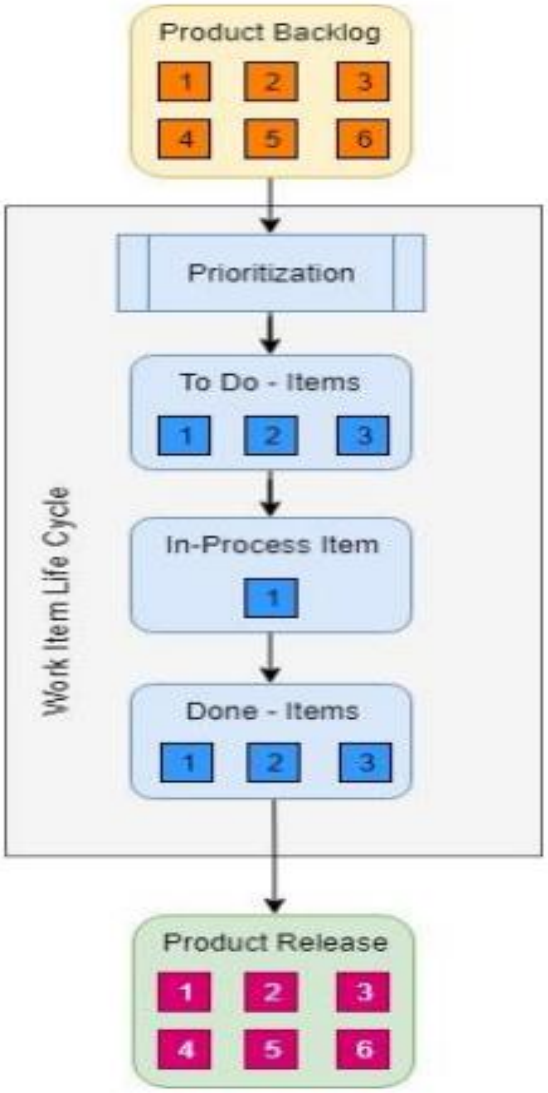
Orientación a la generación de valor

a. **Visualización de flujo de trabajo.** Consiste en distribuir las tareas que se detallan en las tarjetas Kanban para finalmente pegarlo en la pared. Además, se crean las columnas con nombres para percibir con facilidad los elementos del flujo de trabajo.
b. **Limitar el trabajo en curso (WIP).** Es determinar los límites explícitos de las tarjetas que están en la "columna progreso" en cada estado de flujo de trabajo.
c. **Medir el tiempo de entrega.** Es el tiempo promedio para culminar un artículo, mejora el flujo para así realizar la entrega en el menor tiempo.

La Figura 10 representa el flujo de Kanban, junto con la visualización de la gestión del flujo de trabajo para el estado de cada elemento de trabajo. Los requisitos de desarrollo de productos se convierten en Product Backlog Items (PBI) con la priorización de cada uno de ellos. Los elementos de trabajo se colocan en la lista de tareas pendientes según la capacidad del equipo de desarrollo. De acuerdo con el límite WIP, cada desarrollador toma solo un elemento de trabajo en proceso. Los elementos completados se mueven a la lista "Listo". Kanban admite CICD (Interacción Continua y Entrega Continua para la liberación de elementos de trabajo Completados) o lanzamiento de producto (Bhavsar, Shah & Gopalan, 2020).

Figura 10

Flujo de elementos de trabajo Kanban.



Nota. El flujo de Kanban en el flujo de trabajo.

Fuente: Adaptado de International Journal of Innovative Technology and Exploring Engineering (1628), por Bhavsar, K., Shah,V. y Gopalan, S., 2020, Blue Eyes Intelligence Engineering & Sciences Publication.

En resumen, Kanban se enfoca en visualizar el flujo de trabajo, mejorar el proceso del flujo de trabajo al limitar la cantidad de elementos de Work in Pro-gress (WIP) en diferentes etapas y, finalmente, mejorar el proceso en los plazos de entrega orientados en ser más pequeños y predecibles (Bhavsar et al., 2020).

2.2.8. Scrumban

Para (Pérez Pérez, 2012) "es una metodología originaria de las guías Scrum y Kanban" (p.44).

Esta metodología gestiona el desarrollo de software basado en diferentes situaciones de proyectos, además es híbrida ya que presenta elementos y definiciones que se emplean en conjunto, para alcanzar el flujo de desarrollo del trabajo óptimo.

Para Khan (2014) el framework Scrumban utiliza las buenas características de ambos modelos "Utiliza la naturaleza prescriptiva de Scrum para ser ágil (equipos autoorganizados, superación personal, flujo de información constante, etc.) y por otro lado fomenta la mejora de procesos utilizando Kanban"(p.46).

Scrumban no se trata de usar solo algunos elementos de Scrum y Kanban para crear un proceso de desarrollo de software. Más bien, enfatiza la aplicación de sistemas Kanban dentro de un contexto de Scrum y la superposición del Método Kanban junto con Scrum como un vehículo para el cambio evolutivo, en última instancia, se trata de ayudar y ampliar las capacidades que ya son inherentes a Scrum, además de proporcionar nuevas perspectivas y capacidades (Reddy, 2014).

Características

Según (Z. Khan, 2014), define las principales actividades que se realizan en Scrumban:

a. **Visualizar el flujo de trabajo**. Es el corazón de Kanban que consiste en visualizar el flujo de proceso de las historias de usuario durante cada etapa de desarrollo de software, estas etapas se representan en columnas, Scrumban opta esta característica en su marco de trabajo. El flujo de trabajo se desarrolla de izquierda a derecha, partiendo de la columna de inicio con el Sprint backlog y finalizando con la columna "completado". La visualización literalmente, ayuda a identificar los problemas que se presentan en el transcurso del desarrollo del software, de la misma manera, ayuda a ver el avance logrado por el equipo de trabajo de esta manera también aporta la sincronización del equipo (Z. A. Khan, 2014).

b. **Cola de trabajo.** El marco de trabajo que realiza Scrum genera un compromiso de entrega del proyecto esperado del cliente, esto se realiza mediante la selección prioritaria de las historias de usuario que se desarrollará en un sprint particular. Es decir, al iniciarse el sprint, ya no se aceptan cambios en sus historias de usuarios. Por otro lado, en Scrumban sucede lo contrario, debido a que Kanban presenta las "colas de trabajo" a su vez permite las modificaciones de los sprint cuando sea solicitado (Z. A. Khan, 2014).

c. **Limitar el trabajo en progreso (WIP).** Otra característica principal de Scrumban es respetar la capacidad de tareas empleadas en cada etapa de trabajo dependiendo del rendimiento del equipo. De esta manera se puede detectar la acumulación de trabajo en el flujo de proceso para facilitar la culminación de trabajo y aceptar los nuevos requerimientos (Z. A. Khan, 2014).

d. **Reglas explícitas**. En el marco de trabajo Scrum, el trabajo se desarrolla a través de reglas implícitas, es decir, cada quien organiza, coordina sus labores que van a desarrollar en el proyecto. Pero en la práctica se observa una realidad distinta de como un equipo se

autoorganiza y como lo están poniendo en marcha. Por esta razón, otra de las características de Scrumban es que el trabajo que se emplea está sujeta a reglas explícitas, es decir que todos los integrantes del equipo puedan atribuirse actividades libremente para alcanzar un flujo de labores más dispuesto y eficiente. De esta manera, el equipo que integra el proyecto gana tiempo en decidir sin presentar muchas dudas e incluso presenta pocas posibilidades a equivocarse, además de acceder requerimientos en una situación de trabajo recargado. Estas reglas son beneficioso ante los casos frecuentes donde se necesitan tomar decisiones de una situación particular a lo largo del desarrollo del proyecto (Z. A. Khan, 2014).

e. **Reuniones en Scrumban.** Al igual que en Scrum, existe la presencia de reuniones con cambios significativos como, el número de veces que se desarrolla las reuniones (Z. A. Khan, 2014).

f. **Reuniones de planificación.** Mientras Scrum presenta estas reuniones más prolongadas, en Scrumban se desarrollan en un menor tiempo con el objetivo de renovar el backlog cuando lo amerite**.** Así mismo, ante el constante cambio que presenta el backlog, no es conveniente realizarse reuniones de planificación con un mayor tiempo. En este sentido, Scrumban se enfoca en reducir el tiempo en este tipo de reuniones (Ladas, 2008).

   - ✓ Reunión Stand-up (diaria). Al igual que el enfoque Scrum, este tipo de reunión es operante y apoya en la reorganización de los movimientos que se desarrollarán de manera cotidiana además de resolver los obstáculos que se presentan durante el desarrollo del proyecto.
   - ✓ Reunión de revisión de Sprint. Al igual que Scrum, se mantienen las mismas características para este tipo de reunión.

- ✓ Reunión de retrospectiva. Scrumban enfatiza los problemas encontrados en el flujo del proceso que se presentaron en los trabajos anteriores, de manera que pueda ser más claro las dificultades encontradas para anticiparse en las próximas situaciones.

Scrumban es distinto de Scrum porque enfatiza ciertos principios y prácticas que son bastante diferentes de la base tradicional de Scrum (Reddy, 2016). Estos incluyen los siguientes:

a. Reconocer el papel de la dirección (la autoorganización sigue siendo un objetivo, pero dentro del contexto de límites específicos).
b. Habilitación de equipos y funciones especializados.
c. Aplicar políticas explícitas en torno a las formas de trabajar.
d. Aplicar leyes de flujo y teoría de colas.

Scrumban se distingue del método Kanban de las siguientes formas:

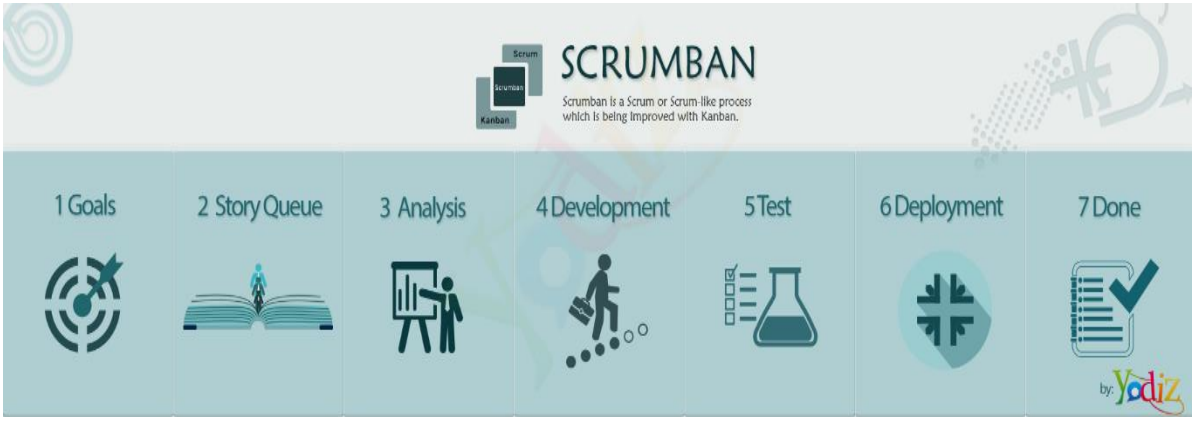
a. Prescribe un marco de proceso de desarrollo de software implícito.
b. Adquiere el proceso como núcleo de Scrum.
c. Está organizado y comprometidos como equipos.
d. Las iteraciones encuadradas son reconocidas en el tiempo cuando es apropiado.
e. Establece técnicas de mejora continua dentro de ceremonias definidas.

El proceso de Scrumban.

El trabajo se divide en procesos iterativos y se monitorea con la ayuda de un tablero visual. (Yodiz, 2015).

Figura 11

Proceso de Scrumban: Una amalgama de Scrum y Kanban



Fuente: Scrumban: una fusión de Scrum y Kanban., por el equipo de Yodiz, 2015, https://www.yodiz.com/blog/scrumban-an-amalgamation-of-scrum-and-kanban/

Scrumban brinda al equipo la facilidad de incluir constantemente las historias de usuario, sin sentirse sobrecargado por el proceso. Además, Scrumban opta por eliminar los eventos de reuniones como se realiza en el marco de Scrum para incrementar el desarrollo de labores del equipo Scrumban. La ventaja que se obtiene con este marco es que facilita la eficiencia del trabajo en equipo mediante la determinación del tiempo en el desarrollo del proyecto, logrando obtener la calidad del proyecto en un menor tiempo.

Valores de Scrumban

Según Ladas (2008), Como marco independiente centrado en los resultados finales, Scrumban aporta tres valores (pág. 64).

a. ***Empirismo.*** Los enfoques empíricos siempre optan por las teorías, al igual que los resultados verificables sobre las dogmas.

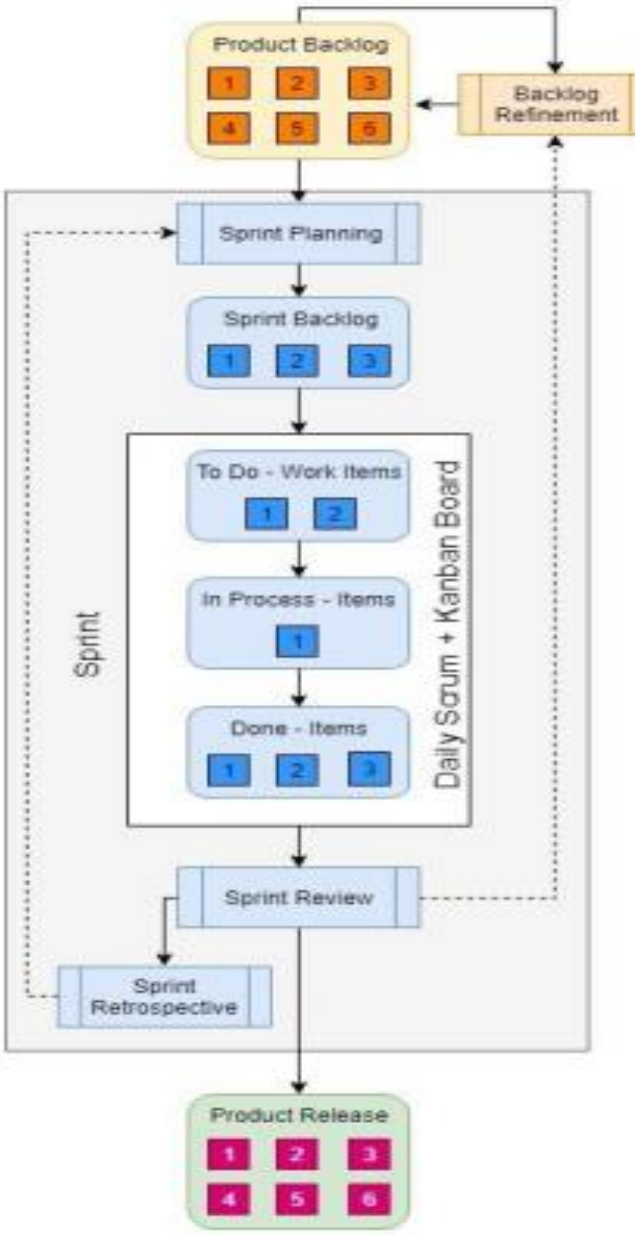
b. ***Humildad.*** Los sistemas cambian constantemente y estamos aprendiendo constantemente. Siempre debemos de estar dispuestos a desafiar nuestro entendimiento. La comprensión y los enfoques mejorados pueden provenir de cualquier fuente**.**

c. ***Interacción constructiva.*** Siempre habrá marcos y métodos de gestión en competencia. Scrumban enfatiza el debate constructivo que mejora la comprensión de las fortalezas y limitaciones de cada uno sobre la aceptación ciega de que cualquier marco representa la "única" o la "mejor" forma de lograr un resultado en particular.

La Figura 12 representa el flujo del sistema para el marco Scrumban propuesto mediante la adopción de elementos y procesos centrales de Scrum y Kanban. Los elementos de desarrollo de productos se recopilan en el grupo de Product Backlog, en forma de historias de usuario, donde se refinan con la frecuencia que el cliente o las partes interesadas solicitan los cambios. El proceso de refinamiento incluye la priorización de las historias de usuario y cada historia se incluye en el Scrum Backlog durante el evento Sprint Planning según la prioridad. El papel de Kanban comienza desde esta etapa en Scrumban. Las historias de usuario deben convertirse en tarjetas Kanban en el Sprint Backlog por parte de los miembros del equipo de desarrollo. Cada uno contiene una o más especificaciones de tareas técnicas con estimación precisa para cada tarea y pasa por cada estado de los elementos de trabajo. Los elementos completados que confirman DoW (Definición de Flujo de trabajo), debe trasladarse a los eventos Scrum Review donde las historias completadas son inspeccionadas por el equipo Scrum junto con las partes interesadas. Las historias de usuario confirman DoD (Definición de Listo), se considerará para el lanzamiento del producto. A diferencia de Scrum, el sistema Kanban no tiene eventos de tiempo en caja y eso es por qué Scrumban toma prestado Sprint Planning, Daily Scrum y Revisión de Sprint desde Scrum.

Figura 12

Flujo de Scrumban



Nota. Fuente: International Journal of Innovative Technology and Exploring Engineering (1630), por Bhavsar, K., Shah,V. y Gopalan, S., 2020, Blue Eyes Intelligence Engineering & Sciences Publication.

A continuación, (Rodríguez Franco, 2014), presenta la tabla 1 de las diferencias entre Scrum y Kanban.

Tabla 1

Diferencias entre las Metodologías Scrum y Kanban.

Scrum	Kanban
Las iteraciones se desarrollan en un tiempo definido.	Las iteraciones no están definidas en un tiempo establecido, ya que están basadas en el plan del proyecto y en mejorar el desarrollo del proceso.
A llevar a cabo en evento del sprint, el equipo asume la responsabilidad de trabajo.	La adquisición de compromisos no es obligatoria.
La variable velocidad es un factor que mide el proceso de la panificación y demás de la mejora del proceso	Se define el tiempo de entrega que demora la cuestación en culminar el ciclo.
Se determinan equipos especializados.	Se determinan equipos especializados.
Los requerimientos se distribuyen de manera que se ejecute correctamente en un sprint.	Las distribuciones no se encuentran ordenadas.
Se emplean los gráficos Burndown.	No está definido los diagramas de seguimiento.
Se realiza la determinación de tareas establecidas de manera indirecta con los límites WIP.	Se aplica la limitación WIP directa de acuerdo al trabajo realizado.
Se realizan estimaciones de las historias de usuario.	La estimación es opcional.
Dada la iteración, no es posible agregar tareas adicionales.	Siempre en cuando se goce capacidad especializada se podrá agregar tareas.
Las historias de usuario a la espera de ejecutarse en el sprint, le corresponde a un quipo dado.	La tabla de Kanban se puede colaborar y utilizar con los demás integrantes de los equipos.

Se instituyen roles definidos.	No existe roles determinados.
Para cada sprint se culmina el tablero de seguimiento.	El tablero Kanban es persistente hasta culminar el proyecto.
La pila del requerimiento debe estar priorizada según el cliente.	La priorización del requerimiento es opcional.

Fuente: Rodríguez J. (2014)

Según las definiciones determinadas por (Sáez, 2013), nos presenta la tabla 2 de resumen de las metodologías Kanban, Scrum y Scrumban.

Tabla 2

Tabla Resumen Metodologías Kanban, Scrum y Scrumban.

Parámetro	Kanban	Scrum	Scrumban
Tiempo para cada iteración recomendado	Flujo de trabajo es invariable. Los entregables se liberan de acuerdo a la necesidad.	Se establece de 2 a 4 semanas.	Flujo invariable. De igual manera que en Kanban, se liberan acuerdo a la necesidad.
Tamaño del equipo	No se encuentra definido.	Se pueden presentar de diferentes tamaños (Scrum de Scrum).	Equipos pequeños y necesarios.
Comunicación en el equipo	Se establece de manera Voluble, de frente, mediante el tablero Kanban, además cada equipo es independiente de elegir sus procesos.	Se establece de manera voluble, mediante práctica de reuniones diarias parados y reuniones de retrospectiva.	Se establece de manera voluble, además de la práctica de reuniones diarias parados, mediante el tablero Kanban y también de las reuniones de retrospectiva.

Tamaño del proyecto	Proyectos de corta, mediana duración.	Proyectos de corta y mediana duración	Proyectos de corta y mediana duración.
Involucración del cliente	Se da por conveniente una colaboración alta del cliente.	El cliente presenta colaboración activa mediante cruzando el rol de ProductOwner.	El cliente está comprometido y motivado mediante la comunicación activa.
Documentación del proyecto	No se elabora la documentación del proyecto.	Se elabora solo la documentación primordial.	Al igual que Scrum, solo se elabora la documentación principal.
Habilidades especiales	Implementación del Tablero Kanban	Establecen los eventos como: Sprint Product, Sprint Backlog, Scrum Board, Scrum Master y Planning Poker.	Al ser un marco de trabajo híbrido, se implementa lo principal de Scrum y Kanban.
Ventajas	Proporciona la facilidad de conocer las actividades que se están desarrollando Restringe la cantidad de tareas establecidas en el mismo tiempo. Implementación simple Accesible a cambios.	Equipo autoorganizado. Equipo de trabajo tienen claro su labor. El cliente conoce el producto de cada sprint. Accesible a cambios. Los desarrolladores cuentan con libertad. Reduce la labor de gestión.	Se complementan las ventajas de Kanban con los aportes de mejores prácticas de Scrum.
Desventajas	Es necesario la definición de las fases del flujo de trabajo.	Al presentar continuos sprint, puede ocasionar una tendencia a cansancio acumulado.	La percepción de cada marco de trabajo como independiente, y no como

	No determina roles, ni fases, tampoco ahonda en el tablero Kanban. Se inclina a ser una técnica que una metodología de trabajo.	Necesita de un equipo especializado, con experiencia para afrontar todas las actividades. Es necesario el compromiso del cliente. Dificultades en la repartición de espacios.	una nueva manera de trabajo. No se define de manera clara los roles y fases.

Nota. Identificación y valoración de técnicas ágiles de gestión de proyectos software.

Fuente: Sáez P. (2013), Oviedo: Universidad de Oviedo.

2.2.9. ScrumDo

Reddy (2016) fundó CodeGenesys, una boutique de consultoría Lean-Agile, en 2009. En 2010, codesarrolló ScrumDo, una herramienta de gestión de Scrum y Kanban, con el propósito expreso de facilitar las implementaciones Agile en toda la industria. Durante los últimos cinco años, ha ayudado a organizaciones tanto grandes como pequeñas a emplear Scrumban con gran éxito. Ajay cree que Scrumban es un marco particularmente simple pero poderoso con una gran cantidad de potencial sin explotar, además cocreó el juego GetScrumban como una ayuda práctica y eficaz para ayudar a las personas y organizaciones a orientarse hacia las verdaderas capacidades de este marco (p.79).

ScrumDo permite una configuración sencilla con un asistente y planificación de proyectos a nivel de iteración o nivel de lanzamiento. Es fácil rastrear y desplazar las historias de usuario en su iteración o en un nivel de flujo continuo.

Capítulo III

Metodología de la Investigación

3.1. Tipo de Investigación

De acuerdo a Hernández, Fernández y Baptista (2014), las características fundamentales de la investigación aplicada es presentar fines prácticos, puntuales y a tiempo, con el propósito de indagar para ejecutar, cambiar, modelar y crear la transformación de un enfoque de la realidad. Para efectos de esta investigación es necesario el aporte de los estudios principales y esenciales que se desprenden de las teorías científicas.

Considerando el estudio de los autores citados previamente, el tipo de **investigación es aplicada.**

3.2. Nivel de la Investigación

Conforme al investigador Hernández Sampieri et. al (2014), las investigaciones descriptivas son estudios que consisten en detallar los rasgos, las características y los contornos destacados de los individuos, asociaciones u otro fenómeno que sea puesto al análisis. La función principal de la investigación descriptiva es tener la suficiencia de elegir las cualidades esenciales del objeto de estudio y su definición minuciosa sobre los fragmentos, especies o clases de dicho objeto; así mismo, la investigación descriptiva tiene una gran aceptación de manera reconocida y usada por los nuevos usuarios que se están formando en la investigación.

La investigación descriptiva, es mayormente empleada en los estudios de los diferentes grados académicos. Teniendo en cuenta que los eventos se narran, describen o detectan hechos, escenas de un objeto de estudio, también se crean productos, modelos, diseños, etcétera.

Carrasco (2019), señala que,

> La investigación descriptiva se sujeta principalmente de estas técnicas: la encuesta, la entrevista, la observación y revisión documental". Esto tiene como finalidad "estudiar, analizar, describir y especificar realidades y cualidades de personas, grupos, comunidades o cualquier otro fenómeno u objeto que sea analizado.

Apreciando la idea del autor, el nivel de investigación es descriptivo.

3.3. Diseño de la Investigación

Conforme al anunciado de Hernández Sampieri et. al. (2014), donde define que la investigación no experimental es una indagación que consiste en no modificar a propósito las variables, este estudio consiste en que las variables independientes no los sometidos a variaciones de manera intencional. Al realizar una observación de los fenómenos en el contexto natural, para que a partir de esto sea analizado, estamos incurriendo a la investigación no experimental (p. 191).

Acorde a Carrasco (2019), se pronuncia sobre el diseño de investigación transversal descriptivo, que consiste analizar para que de esta manera se pueda identificar las particularidades, caracteres internas y externas y rasgos principales de los casos y actividades que se produce en la naturaleza de un determinado caso en el tiempo preciso.

Conforme a Hernández et. al. (2014), la investigación de diseño transversal consiste en la colección de datos en un tiempo único con fin de detallar las variables y analizar los sucesos que se presentan. Durante del proceso, para la recolección de datos se usan los instrumentos de forma única.

Apreciando las investigaciones expuestas, en esta investigación es importante conocer y analizar los procesos, características, estudiar rasgos y entender las funcionalidades, y así poder puntualizar los datos esenciales para la creación del modelo operativo; consiguientemente, el diseño de investigación se determina como **no experimental de tipo transversal descriptivo.**

3.4. Población y Muestra

A. Población

La población está conformada por todas las metodologías de gestión de proyectos de software en entornos ágiles.

B. Muestra

La muestra está conformada por la metodología ágil de desarrollo programación extrema sobre el marco de trabajo Scrumban para la gestión de Proyectos de Software.

3.5. Variables e Indicadores

3.5.1. Definición Conceptual de la Variable

A. Variable de estudio 1

X: Scrumban. Es una metodología híbrida, para la gestión de proyectos, que combina el enfoque de Scrum y Kanban de manera que; el primero, utiliza la naturaleza prescriptiva para ser ágil y el segundo, mejora el proceso. Es decir, enfatiza la aplicación del sistema Kanban dentro de un contexto de Scrum.

Indicadores

X1: Iteraciones. Son ciclos en el flujo de trabajo con cambios incrementales, creados por medio de los procesos repetidos de prueba y retroalimentación, donde se realizan modificaciones en la funcionalidad del proyecto.

X2: Planificación bajo demanda. Se realiza cuando se va deliberar las tareas que se encuentran pendientes en el tablero Scrumban. Cuando cae por debajo del valor limitado, se realiza el evento de la planificación.

X3: Priorización. El cliente indica las tareas que se deben desarrollar primero y las que se pueden postergar, el equipo de trabajo comprende qué es lo más importante para el negocio del cliente y puede ofrecer funciones que le brinden mayor valor.

B. Variable de estudio 2

Y: Gestión de proyectos de software en entornos ágiles. Son metodologías de desarrollo de software basados en el desarrollo iterativo e incremental, así mismo, adapta el trabajo en función de las necesidades del cliente y las expectativas del usuario, además, crea un entorno de trabajo con profesionales motivados y comprometidos en sus labores.

Indicadores

Y1. Estrategia de desarrollo incremental. Consiste en insertar nuevas funcionalidades al código existente; este incremento representa un subconjunto completo de funcionalidad y es entregado al cliente en un periodo de tiempo planificado.

Y2. Calidad del resultado. Es un proceso eficaz de software que cumple con los requisitos específicos y las necesidades o expectativas del cliente o usuario; así mismo,

proporciona valor medible a quienes lo producen y a quienes lo utilizan, además es el resultado de la buena administración del proyecto y la correcta práctica de ingeniería de software.

Y3. Conocimiento de equipos auto organizados. Los integrantes del equipo son disciplinados que cuentan con la experiencia requerida y conocen el proyecto de forma general y detallada; además trabajan con reglas explícitas, donde les permite tomar decisiones libremente para lograr el objetivo del proyecto.

3.5.2. Definición operacional de la variable

A. Variable de Estudio 1

X: Scrumban

Indicadores

X1: Iteraciones

X2: Planificación bajo demanda

X3: Priorización

B. Variable de Estudio 2

Y: Gestión de proyectos de software en entornos ágiles.

Indicadores

Y1: Estrategia de desarrollo incremental

Y2: Calidad del resultado

Y3: Conocimiento de equipos auto organizados.

3.6. Técnicas e Instrumentos para el Tratamiento de Datos e Información

3.6.1. Técnicas para recolectar información

Se empleó la técnica de "Análisis Documental" para recabar información de los estudios e investigaciones realizadas por los expertos correspondiente a la metodología de Scrumban y la programación extrema.

3.6.2. Instrumentos para recolectar información

Guía de análisis documental, es un instrumento que se ha implementado en esta investigación para el desarrollo del marco de trabajo Scrumban y la metodología de la programación extrema, la recolección de información realizada por el instrumento se obtuvo del análisis completo del investigador.

3.6.3. Herramientas para el tratamiento de datos

Las herramientas que servirán como base para la implementación del modelo propuesto en la aplicación de Scrumban sobre la Programación Extrema para alcanzar el desarrollo de gestión de proyectos ágiles, se describen a continuación:

Tabla 3

Herramientas para el Tratamiento de Datos.

Nombre	Creador	Uso
Programación Extrema (XP)	Kent Beck	Es una metodología ágil de desarrollo de software basado en la simplicidad y agilidad para entregar a los clientes en el momento en el que lo necesita.

Scrumban	Corey Ladas	Es una metodología derivada de los enfoques Scrum y Kanban, combina la agilidad o empirismo con un sistema de gestión de flujo de trabajo transparente.
ScrumDo	Ajay Reddy	Es una herramienta de gestión de Scrum y Kanban, se basa en los principios básicos de Scrumban.

Fuente: Elaboración propia en base (Borja López, 2013).

3.7. Desarrollo del Modelo Propuesto Scrumban sobre la Programación Extrema en la Gestión de Desarrollo de Software

3.7.1. Análisis comparativo

El modelo propuesto busca utilizar Scrumban sobre la Programación Extrema para la gestión de proyectos de software en entornos ágiles a través de las iteraciones, planificación bajo demanda y la priorización de las tareas, con el objetivo de adoptar una estrategia de desarrollo incremental, generar calidad en el resultado y generar conocimiento en equipos auto organizados respectivamente. Para lograr los objetivos, se necesita el enfoque de gestión y organización de Scrum, que conlleva a realizar un análisis comparativo mediante tablas de comparación de las diferencias y similitudes del marco de trabajo Scrumban y la metodología XP.

Tabla 4

Análisis Comparativo de las Metodologías Ágiles: XP, Scrum, Kanban y Scrumban.

Característica	XP	Scrum	Kanban	Scrumban
Tiempo de Iteración	De 2 a 6 semanas	De 2 a 4 semanas	Flujo continuo. La liberación de los entregables se da en base a las necesidades.	Flujo iterativo e incremental de forma continua en base a eventos o necesidades.
Tamaño del equipo de trabajo	Menos de 30 personas.	Equipos pequeños, multifuncionales y autoorganizados (4 a 10 personas).	Sin prescripción	Equipos pequeños especializados y multifuncional.
Manera que se muestra la comunicación	De frente (cara a cara). Práctica de reuniones diarias de pie. Se realiza mediante los artefactos.	Prácticas de reuniones diarias de pie. Además de Reuniones de retrospectiva.	De frente (cara a cara). Mediante el tablero Kanban. Los integrantes del equipo son libres de elegir sus procesos.	Prácticas de reuniones diarias de pie. Mediante el tablero Kanban. Reuniones de retrospectiva.
Tamaño del proyecto	Proyectos pequeños.	Proyectos pequeños y medianos.	Proyectos de corta y mediana duración.	Proyectos de corta y mediana duración.
Involucración del cliente	Cliente comprometido. Comunicación expresiva.	Cliente comprometido a través del rol de Product Owner.	Cliente involucrado.	Cliente comprometido. Comunicación expresiva. El cliente decide prioridades.
Documentación del proyecto	Documentación básica, código autodocumentado.	Documentación básica.	Evade la documentación. Los procesos son definidos por el equipo.	Se implementa una documentación necesaria.

Fuente: Elaboración propia con datos en base a Sáez P. (2013).

Tabla 5

Comparación de los Eventos de las Metodologías: XP, Scrum, Kanban y Scrumban.

XP	Scrum	Kanban	Scrumban	Nota
Fase de Exploración	Sprint Planning	No define	Sprint Planning	Actividades similares
Fase de Planificación				
Fase de Interacción y lanzamientos	Daily Scrum	No define	-	Scrumban adquiere el Daily Scrum de Scrum.
Fase de Producción				
Fase de Mantenimiento	-	-	-	Este evento es propio de XP.
Fase de Muerte	Sprint Review	No define	Sprint Review	Scrumban adquiere el Sprint Review de Scrum. Las fases de XP son similares.
-	Sprint Retrospective	No define	Sprint Retrospective	Scrumban adquiere el Sprint Retrospective de Scrum.

Nota. Comparación de los Eventos de las Metodologías: XP, Scrum, Kanban y Scrumban; para integrarlo en el modelo propuesto.

Tabla 6

Comparación de los Artefactos de las Metodologías: XP, Scrum, Kanban y Scrumban.

XP	Scrum	Kanban	Scrumban	Nota
Historias de Usuario	Product Backlog	No define	Product Backlog	Se realizan actividades similares
Tareas de ingeniería	Sprint Backlog	No define	Sprint Backlog	Se realizan actividades similares
-	-	Tablero de Kanban	Scrumban Board	Scrumban adopta el tablero Kanban con más funcionalidades
Prueba de aceptación	-	-	-	Estos artefactos son propiamente de XP
Tarjetas CRC				
	Increment	-	Increment	Scrumban adquiere Increment Scrum.

Nota. Comparación de los Artefactos de las Metodologías: XP, Scrum, Kanban y Scrumban; para integrarlo en el modelo propuesto.

Tabla 7

Comparación de Roles de las Metodologías: XP, Scrum, Kanban y Scrumban.

Parámetro	XP	Scrum	Kanban	Scrumban	Nota
Roles	Cliente	Product Owner	El Gestor de Peticiones de Servicio	Los roles necesarios	Se realizan funciones similares
	Entrenador (Coach)	ScrumMaster	No define	Los roles necesarios	Son similares
	Encargado de pruebas	Development Team	El Gestor de Prestación de Servicio.	Los roles necesarios	Los roles de XP están involucrados en development team de Scrum y así mismo en el Gestor de Prestación de Servicio del marco de Kanban.
	Encargado de seguimiento				
	Consultor				
	Gestor				

Nota. Comparación de los Roles de las Metodologías: XP, Scrum, Kanban y Scrumban; para integrarlo en el modelo propuesto.

Tabla 8

Comparación de Valores de las Metodologías: XP, Scrum, Kanban y Scrumban.

Parámetro	XP	Scrum	Kanban	Scrumban	Nota
Valores	Coraje	Coraje	Liderazgo	-	Similares
	Respeto	Respeto	Respeto	-	Similares
	Retroalimentación	- - -	Entendimiento Transparencia Equilibrio	Interacción constructiva	Todos los valores se integrarán en la Retroalimentación de XP.
	Simplicidad	-	-	-	Se incluirá
	-	Compromiso	Colaboración	Humildad	Estos valores se integrarán en el compromiso del marco Scrum.
	-	Atención	Foco en el cliente.	-	Similares
	Comunicación	Franqueza	Acuerdo	-	Estos valores tienen relación y se integrará en la comunicación de XP.
		-	-	Empirismo	Se incluirá
	-	-	Flujo	-	Se incluirá en el modelo.

Nota. Comparación de los Valores de las Metodologías: XP, Scrum, Kanban y Scrumban; para integrarlo en el modelo propuesto.

Tabla 9

Comparación de los Principios y Prácticas de las Metodologías: XP, Scrum, Kanban y Scrumban.

	Kanban	Scrumban	Scrum	XP	Nota
Prácticas y Principios	Visualización de flujo de trabajo.	Visualizar el flujo de trabajo.	-	-	Prácticas iguales
	Limitar el trabajo en curso (WIP).	Limitar el trabajo en progreso (WIP).	-	-	Prácticas iguales
	Medir el tiempo de entrega.	-	Bloque de tiempo asignado.	-	Prácticas similares
	-	Autoorganización	Autoorganización	-	Prácticas iguales
	-	Reuniones en Scrumban.	Reuniones.	-	Prácticas similares
		Reuniones de planificación.			

	-	-	Desarrollo iterativo	Se incluirá
			Entregas pequeñas	Se incluirá
	-	Reglas explícitas	-	-	Estas prácticas y principios se incorporarán en el modelo planteado.
	-	-	Colaboración	-	
				Metáfora.	
				Ritmo sostenido.	
				Estándares de codificación.	
				Cliente en sitio	Esos valores son propios de XP.
				Programación en pareja	
				Recodificación	

Nota. Comparación de los principios y prácticas de las Metodologías: XP, Scrum, Kanban y Scrumban.

3.7.2. Estudio de integración

"Scrum es un marco de desarrollo de software increíblemente simple, efectivo y popular su valor aumenta a medida que los equipos y las organizaciones desarrollan su comprensión y aplicación de sus principios y prácticas fundamentales" (Reddy, 2014).

Kanban, tiene como objetivo principal, optimizar el flujo de trabajo que se logra a través de sus prácticas (Reddy, 2014).

La programación extrema está basada en los principios y buenas prácticas que ayuda al equipo de desarrollo a adaptar las historias con repentinos cambios del cliente.

Por lo expuesto se presenta detalladamente la integración de las metodologías ágiles mencionadas para el planteamiento del modelo, adquiriendo lo mejor de cada una de estos marcos de trabajo para obtener un proyecto final de calidad, funcional y a tiempo.

Tabla 10

Integración de los Eventos para el Modelo Propuesto.

Parámetro	Modelo Propuesto	Observación
Eventos	Sprint Planning	Con la ayuda del análisis comparativo, se pudo sintetizar los eventos del marco híbrido Scrumban para su posterior diseño e implementación.
	Daily Scrum	
	Scrumban Board	
	Sprint Review	
	Sprint Retrospective	

Tabla 11

Integración de los Roles para el Modelo Propuesto.

Parámetro	Modelo Propuesto	Observación
Roles	Product Owner. ScrumMaster Development Team.	Con la ayuda del análisis comparativo, se identificó los roles esenciales de Scrumban.

Tabla 12

Integración de los Artefactos para el Modelo Propuesto.

Parámetro	Origen	Modelo Propuesto	Observación
Artefactos	XP	Historia de Usuario.	Teniendo en cuenta la importancia de requerimientos del cliente, se ve por conveniente utilizar este artefacto para organizar la información.
	Scrum	Product Backlog. Sprint Backlog. Increment.	Estos artefactos son heredados de Scrum a Scrumban.
	Kanban	Tablero Kanban.	Es el artefacto fundamental de Scrumban.

Tabla 13

Integración de los Valores para el Modelo Propuesto.

Parámetro	Origen	Modelo Propuesto	Observación
Valores	XP	Comunicación	Nos damos cuenta que los valores de XP predominan ante los demás, ya que ésta abarca casi todos los valores de los marcos de trabajo.
		Respeto	
		Retroalimentación	
		Coraje	
	Scrum	Compromiso	
	Scrumban	Empirismo	
	Kanban	Flujo	
		Foco en el cliente	

Tabla 14

Integración de las prácticas y principios para el Modelo Propuesto.

	Origen	Modelo Propuesto	Observación
Prácticas y Principios	Kanban y Scrumban	Visualización de flujo de trabajo.	Se implementarán estas prácticas y principios de los diferentes marcos de trabajo y metodología ágil XP.
		Limitar el trabajo en curso (WIP) Scrumban.	
		Medir el tiempo de entrega.	
	Scrumban y Scrum	Autoorganización	
		Reunión	
	Scrum	Colaboración	
	Scrum y XP	Desarrollo iterativo	
		Metáfora	
		Ritmo sostenido	
		Estándares de codificación.	

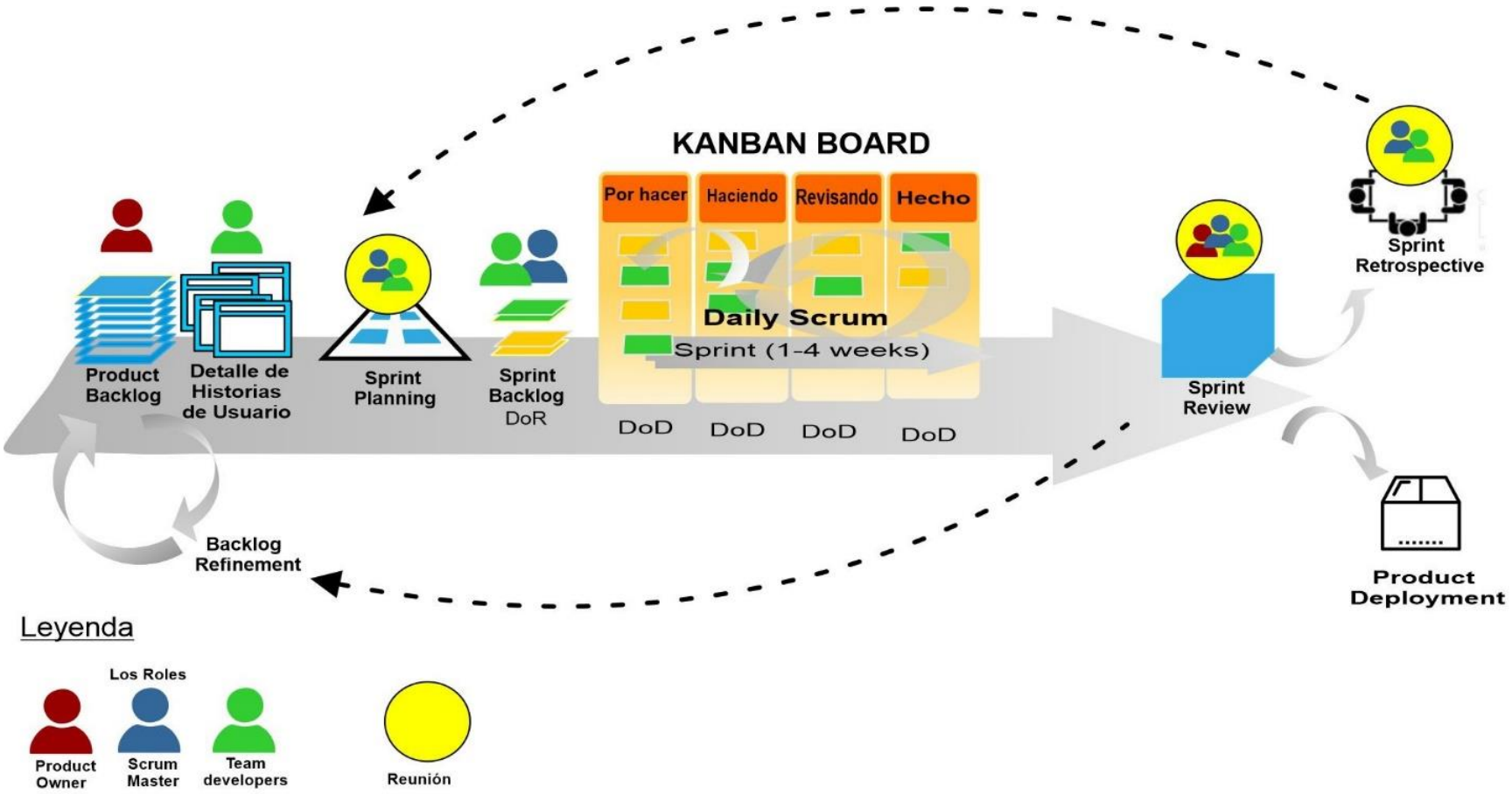
3.7.3 Planteamiento del modelo propuesto

En la figura 13, representa el flujo de la metodología de Scrumban propuesto mediante la adopción de elementos principales de Scrum y Kanban, además de insertar las buenas prácticas de la Programación Extrema.

El proceso comienza con la recopilación de requisitos que el **Product Owner** recoge del cliente sobre el producto deseado, estos requerimientos se escriben mediante el artefacto "**Product Backlog**", seguidamente el Product Owner proporciona la prioridad para cada requisito de acuerdo con el valor del cliente, además se refinan con la frecuencia que el cliente solicita los cambios. Así mismo, el team developers pasa a detallar estas Historias de Usuario para una descripción completa y organizada. El proceso de refinamiento consiste en la priorización de las Historias de Usuario según la regla del negocio y posteriormente pasan al evento **Sprint Planning**, en este evento acuerdan un objetivo de Sprint que define lo que se supone que alcanzará el próximo Sprint, las tareas que se debe realizar en el sprint y estimar el tiempo de cada sprint. En **Kanban Board**, las historias de usuario se denominan tareas donde se trasladan de izquierda a derecha, el Product Owner coloca las tareas que se han designado al backlog del producto en la columna titulada "Haciendo" del tablero Kanban, al mismo tiempo el Scrum Master y el team developers inician el sprint llenando el tablero con los detalles del sprint en cada proceso de desarrollo de software, así mismo en este evento, el cliente cuenta con una participación activa donde conforman que **DOD** (definición de listo) se considerará para el lanzamiento del producto. Los elementos completados que confirman el DOW (definición de flujo de trabajo) se trasladan al evento **Scrum Review** donde el equipo de Scrumban y las partes interesadas (cliente gerente, dueños del negocio, etc.) inspeccionan las tareas completadas.

Figura 13

Flujo de Scrumban Propuesto.



DOD: Definición de terminado

DOR: Definición de listo

3.7.4. Descripción de la técnica del modelo propuesto

I. Actores

El modelo planteado propone a los miembros que tienen como objetivo lograr el propósito de obtener un producto final satisfactorio al cliente. Para tal caso, el equipo de trabajo está garantizado por el Product Owner, ScrumMaster y Development Team. Así mismo, la cantidad ideal de integrantes es de 4 a 9 personas.

a. **Product Owner:** Su responsabilidad es garantizar el éxito del proyecto requerido y cumplir con el cliente, además debe traducir y listar los requerimientos del proyecto según la demanda del cliente. Así mismo, proporciona la prioridad adecuada para cada requisito según el cliente, ordenando el valor de mayor a menor. El propietario del producto debe facilitar la comunicación entre los clientes y el equipo de ser necesario.
b. **Scrum Master:** Es un líder que dirige al equipo hacia el logro del propósito. Su responsabilidad es lograr la efectividad del equipo eliminando cualquier dificultad que se presente. El equipo se organiza y se gestiona a sí mismo con un alto grado de autonomía y responsabilidad.
c. **Development Team:** Son los miembros que están trabajando en el proceso de desarrollo del producto. No hay una autoridad sobre los desarrolladores, el equipo es autoorganizado, ellos tienen la autoridad de organizarse de manera adecuada para lograr el propósito.

II. Reuniones de la metodología propuesta

Este modelo considera tres reuniones importantes: Planificación del sprint, Sprint Review y el Sprint Retrospective.

a. **Planificación del sprint:** Se debe realizar una reunión después de producir el product backlog para discutir y elegir las tareas apropiadas que se realizarán durante el próximo sprint. Así mismo, se discuten los detalles de las tareas y se estima el tiempo del sprint. Es importante la presencia de todos los miembros del equipo para compartir las sugerencias y opiniones sobre cómo realizar el trabajo.

b. **Sprint review:** Es una reunión donde concluyen todos los miembros del equipo Scrumban, incluyendo a los interesados claves; para presentar los resultados del trabajo al cliente y recopilar los comentarios sobre las funciones producidas. Esta reunión se lleva a cabo posterior a la prueba de calidad de la tabla Kanban.

c. **Sprint retrospective:** Esta reunión ocurre después de culminar el sprint y antes de empezar un nuevo sprint. El objetivo de esta reunión es discutir cómo fue su trabajo en el sprint anterior, cuáles fueron los errores que cometieron y cómo pueden superar dichos errores en el próximo sprint. Asimismo, esta reunión permite al Scrum Master y al Development Team, también puedan discutir las mejoras de su trabajo.

III. Artefactos

a. **Product backlog:** Es una lista de tareas dinámicas y visible para todos los implicados del proyecto. Así mismo, el dueño de producto es el encargado de actualizar esta lista de requerimientos funcionales del software. Los requerimientos del sprint son pequeños y detallados para que posteriormente sean refinados de forma progresiva en un conjunto partes más cortas y precisas, con la cooperación de los stakeholders, dueño del producto y el equipo de desarrollo.

b. **Historias de usuario:** Una historia de usuario es una forma de expresar una necesidad muy específica que tiene un usuario. Por lo general, se escribe en pocas oraciones, a

menudo en una ficha, a veces en una estructura estricta, pero otras veces en un formato flexible. (Stellman & Greene, 2014). Una historia de usuario también contiene información de confirmación en forma de condiciones de satisfacción (Gracia et al., 2014).

c. **Sprint backlog:** Este artefacto está compuesto por una lista de tareas enfocado al desarrollo, de la misma manera son estimadas en horas por el equipo de desarrollo, siendo modificada solo por el equipo de desarrollo durante el sprint. La plantilla para implementar el backlog del producto se muestra en el Anexo D.

d. **Tablero Kanban:** Es un tablero físico (comprendido por notas adhesivas) o digital, administrado por el Scrum Master donde se puede seguir el progreso del trabajo del proyecto en general, además del trabajo de cada sprint. Este tablero está dividido en tres columnas: Por hacer (contiene el trabajo pendiente del sprint), en proceso (contiene las tareas que se está realizando, además se puede dividir en columnas y filas según sea necesario, así mismo se limita el número de tareas (WIP) que se realizan al mismo tiempo) y terminado (contiene tareas completadas del sprint).

IV. Eventos

a. **Sprint Planning:** Se acuerdan los objetivos del sprint que va definir lo que se supone que alcanzará el próximo sprint, también establece las tareas de ingeniería de alta prioridad.

b. **Daily Scrum:** Los miembros del equipo tienen que reunirse diariamente para discutir el flujo de trabajo del sprint, lo que hicieron ayer, lo que harán hoy y lo que harán mañana.

c. **Sprint review:** Este evento es una ocasión para inspeccionar y adaptar el producto construido hasta el momento. La revisión del Sprint, facilita la percepción de la visión del estado actual del producto, además es el momento donde se resuelve los inconvenientes,

donde se realizan preguntas, observaciones o sugerencias y discutir el avance frente a la realidad actual.

d. **Sprint Retrospective:** Este evento es la ocasión para inspeccionar y adaptar lo construido hasta el momento. Así mismo, la revisión del sprint brinda una perspectiva clara del estado actual del producto, también es el momento donde se identifica los problemas mediante el planteamiento de preguntas, realización de observaciones, sugerencias y establecer discusiones de sobre el avance del desarrollo actual del producto. Así mismo, se puede plantear y refutar preguntas como: ¿Qué funcionó bien?, ¿Qué no funcionó bien? y ¿Qué debemos comenzar a mejorar o hacer de manera diferente?

V. Prácticas y principios

a. **Visualización de flujo de trabajo:** Es la visualización del tablero Kanban mediante tarjetas que describen tareas, a la vez están organizadas en columnas para percibir los elementos de flujo de trabajo.

b. **Limitar el trabajo en curso (WIP) Scrumban:** Consiste en respetar la capacidad de tareas desarrolladas en cada etapa de trabajo según la capacidad del equipo.

c. **Medir el tiempo de entrega:** Consiste en estimar un tiempo para terminar una tarea, de esta manera mejora el flujo y la culminación de proyecto en menor tiempo.

d. **Autoorganización:** El equipo reconoce el papel de la dirección dentro de límites específicos.

e. **Reunión:** Existe la presencia de reuniones con cambios significativos como, el número de veces que se desarrolla las reuniones.

f. **Colaboración:** Está relacionado con el trabajo colaborativo de la conciencia, articulación y la apropiación.

g. **Desarrollo iterativo:** Está enfocado en el desarrollo iterativo y destaca la administración de los cambios y la construcción de productos para alcanzar las expectativas del cliente.

h. **Simplicidad:** Un diseño simple se implementa más rápidamente que uno complejo.

i. **Metáfora:** Esta práctica permite explicar de manera simple y detallada el funcionamiento del sistema a los miembros del equipo, clientes y los stakeholders. Del mismo modo, la metáfora se implementa en el código fuente del sistema, para que este se pueda describir así mismo.

j. **Ritmo sostenido:** Reside en llevar un ritmo sostenido de trabajo. Anteriormente esta práctica se denominaba "semana de 40 horas".

k. **Estándares de codificación:** Normas definidas por los desarrolladores para tener un código legible.

VI. Valores

a. **Comunicación:** Consiste en la comunicación cara a cara entre los desarrolladores y el cliente, gracias a esto, el equipo puede realizar cambios que el cliente requiera.

b. **Respeto:** Consiste en promover el trabajo en equipo, donde cada integrante del proyecto forma parte integral del equipo encargado de desarrollar software de calidad. El equipo de trabajo como uno, sin realizar decisiones repentinas.

c. **Retroalimentación:** Este valor permite que los desarrolladores lleven y dirijan el proyecto en una dirección óptima donde el cliente lo requiera.

d. **Coraje:** Radica en que los desarrolladores estén preparados al cambio repentino y afrontar las actividades tempranamente; a través de apoyo y recursos a su disposición.

e. **Compromiso:** El equipo tiene control sobre sí mismo para lograr el éxito del proyecto.

f. **Empirismo:** Están enfocados en la teoría, como los resultados verificables de las afirmaciones.

g. **Flujo:** El equipo desarrolla su trabajo de manera productiva y motivada mediante la concentración que está desarrollando.

h. **Foco en el cliente:** Consiste en involucrarse en la perspectiva centrada de la actividad o en el producto.

Capítulo IV

Análisis y Resultados de la Investigación

4.1. Aplicación del Modelo de Desarrollo de la Programación Extrema sobre Scrumban

4.1.1. Fase N°1: Backlog del Producto

Esta fase es el inicio del proyecto, donde primero se definen las historias de usuario con apoyo del dueño del producto y posteriormente de crea el backlog priorizado del producto.

A. Definición de Historia de Usuario

Tabla 15

Definición de Historias Usuario mediante las preguntas ¿Como?, ¿Quiero? y ¿Para?

ID	Como	Quiero	Para
H_1	Usuario	Registrar colegiado	Utilizar el sistema
H_2	Usuario	Iniciar sesión	Validar registro
H_3	Usuario	Actualizar perfil	Corregir datos
H_4	Usuario	Filtrar los datos del colegiado.	Realizar búsquedas rápidas
H_5	Usuario	Seleccionar el tipo de págo.	Conocer el monto a págar
H_6	Usuario	Realizar págo	Conocer el estado de habilitad
H_7	Usuario	Consultar págos	Ver el estado de págos
H_8	Usuario	Generar reporte de págo.	acreditar el págo realizado.
H_9	Colegiado	Escanear código QR.	Verificar el estado de habilidad
H_10	Cajero	Generar reporte general de inscritos	Conocer la cantidad de colegiados por cada capítulo.
H_11	Cajero	Generar reporte de págo por día	Cierre de caja

B. Crear Product backlog

Tabla 16

Lista del Artefacto Producto Backlog según su Prioridad de Requisitos.

ID	Historia de usuario	Prioridad
H_06	Realizar págo	Alta
H_07	Consultar págos	Alta
H_08	Generar reporte de págo.	Alta
H_09	Escanear código QR.	Media
H_11	Generar reporte de págo por día	Media
H_01	Registrar colegiado	Media
H_02	Iniciar sesión	Media
H_03	Actualizar perfil	Media
H_05	Seleccionar el tipo de págo	Baja
H_04	Filtrar los datos del colegiado	Baja
H_10	Generar reporte general de inscritos	Baja

4.1.2. Fase N°2: Sprint

Esta fase, estará comprendido por la definición de roles del proyecto, el Sprint Backlog, la descripción de las historias de usuario y los sprints del proyecto.

En la siguiente tabla 17, se muestran los roles con sus respectivos integrantes que intervendrán en el proyecto.

Tabla 17

Definición de Roles del Proyecto.

Roles	Integrante
Scrum Master	Judith M. Vilca Alvear
Product Owner	Responsable del Colegio de Ingenieros
Equipo de desarrollo	✓ Programador 1 ✓ Programador 2

I. Sprint Backlog

Tabla 18

Sprint Backlog del proyecto

ID	Historias de Usuario	Responsable	Prioridad	Riesgo	Estimado
H_06	Realizar págo	Programador 1	Alta	Alta	10
H_07	Generar reporte de págo	Judith Vilca A.	Alta	Alta	10
H_08	Consultar págos	Programador 1	Alta	Alta	10
H_09	Escanear código QR	Judith Vilca A	Media	Media	10
H_10	Seleccionar el tipo de págo	Programador 2	Media	Alta	10
H_01	Registrar Colegiado	Programador 2	Media	Media	10
H_02	Iniciar sesión	Judith Vilca A.	Media	Media	10
H_03	Actualizar perfil	Programador 1	Media	Media	10
H_05	Generar reporte de págo por día	Judith Vilca A.	Baja	Media	10
H_04	Filtrar los datos del colegiado	Programador 1	Baja	Baja	10
H_11	Generar reporte general de colegiados.	Judith Vilca A.	Baja	Media	10

II. Detalle de la Historia de Usuario

En este evento, las tareas se detallarán en historias de usuario para su mejor comprensión y manipulación de los datos de cada sprint. Así mismo se refinarán utilizando el siguiente modelo donde cada campo tiene un significado: número de historia de usuario, nombre, usuarios relacionados, la prioridad que es la preferencia del desarrollo respecto a los demás, el riego que se trata de la importancia de la historia de usuario respecto al proyecto, cuantificando de esta manera el daño provocado en caso de fallo.

Figura 14

Descripción de Historia de Usuario "Registrar págos".

Historia de usuario	
Número:H_06	Usuario: Colegido, cajero
Nombre de historia: Registrar pagos.	
Riesgo: Alta	Riesgo: Alta
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Programador 1	
Descripción: • El administrador o cajero hacen clic en la pestaña "Pagar por cuotas" mostrando la interfaz "Realizar Pago". • El sistema muestra en la parte superior la opción de seleccionar en un combo box las opciones de montos acumulables de acuerdo a los meses por pagar. • El sistema muestra una tabla correspondiente que lista el monto y los meses a pagar. • El administrador o el cajero hacen clic el botón "Registrar Pagos" para consignar el pago. • El sistema muestra en la parte inferior derecha la opción "Cancelar", para cancelar la gestión. • El administrador o cajero seleccionan la opción "Pagar por certificado "mostrando la interfaz "Realizar Pago". • El sistema señala que el pago tiene un periodo de duración de tres meses. • El sistema ya te muestra la fecha actual para generar este pago. • Finalmente se muestra una opción "registrar" consignando el pago y además se muestra el botón "Close" para anular la gestión.	

Figura 15

Descripción de Historia de Usuario "Generar Reporte de Págo".

Historia de usuario	
Número: H_07	Usuario: Colegiado, cajero
Nombre de historia: Generar reporte de pago	
Riesgo: Alta	Riesgo: Alta
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Programador 1	
Descripción: • Al hacer clic en la pestaña "Principal". • El sistema muestra la interfaz "Reporte". • El administrador o cajero introduce en el campo "fecha inicio "y "fecha final", las fechas correspondientes para mostrar los pagos. • El administrador o cajero hacen clic en la opción "Generar Reporte Pago". • El sistema muestra el reporte en formato PDF en la opción de descargar e imprimir. • El reporte detalla en una tabla el código de CIP, la fecha de inscripción, el capítulo, el tipo de pago y el monto de pago.	

Figura 16

Descripción de Historia de Usuario "Consultar págos".

Historia de usuario	
Número: H_08	Usuario: Colegiado, cajero
Nombre de historia: Consultar pagos	
Riesgo: Alta	Riesgo: Alta
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Programador 2	
Descripción: • El administrador hace clic en la opción "ver pagos" del colegiado buscado. • Seleccionar el tipo de pago solicitado • El sistema muestra todos los pagos realizados detallando la fecha de pago, estado y tipo de pago.	

Figura 17

Descripción de Historia de Usuario "Escanear código QR".

Historia de usuario	
Número: H_09	Usuario: Colegiado
Nombre de historia: Escanear código QR	
Riesgo: Media	Riesgo: Media
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Judith Vilca	
Descripción: Los terceros (usuarios no registrados), escanean el código QR del recibo de pagos colegiado, lo cual redireccionará a un de un link, permitiendo la consulta de validación de habilidad del colegiado.	

Figura 18

Descripción de Historia de Usuario "Generar reporte General de Colegiados".

Historia de usuario	
Número: H_10	Usuario: Cajero
Nombre de historia: Generar reporte general de colegiados	
Riesgo: Media	Riesgo: Alta
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Judith Vilca	
Descripción: • Al hacer clic en la pestaña "Principal". • El sistema muestra la interfaz "Reporte". • El administrador o cajero hace clic en la opción "Reporte colegiado por capitulo" • El reporte muestra un cuadro donde lista los capítulos con las cantidades de colegiados que las integran.	

Figura 19

Descripción de Historia de Usuario: Registrar Colegiado.

Historia de Usuario	
Número: H_01	Usuario: Colegiado, cajero
Nombre de usuario: Registrar colegiado.	
Prioridad: Media	Riesgo: Media
Puntos estimados: 10	Iteración asignada: 2
Programador responsable: Programador 2	
Descripción: • La interfaz "Registrar colegiado" muestra dos opciones: "registrarse" e "ingresar". • El colegiado selecciona la opción "registrarse", el sistema verifica que los campos obligatorios (nombres, apellido paterno, apellido materno, código de colegiado CIP, tipo de documento, numero de documento, correo electrónico, fecha de inscripción del CIP, contraseña y conformación de esta) estén adecuadamente llenados de lo contrario, alerta el campo que no ha sido llenado mediante un mensaje de color rojo con el enunciado "Obligatorio". • El colegiado selecciona la opción "ingresar", el sistema muestra la interfaz Iniciar Sesión para acceder al sistema, previo éxito del registro del colegiado.	

Figura 20

Descripción de Historia de Usuario: Iniciar Sesión.

Historia de Usuario	
Número: H_02	Usuario: Colegiado, cajero
Nombre de Usuario: Iniciar sesión.	
Prioridad: Media	Riesgo: Media
Puntos estimados: 10	Iteración asignada: 2
Programador responsable: Judith Vilca	
Descripción: • El usuario ingresa su usuario y contraseña, siendo estos campos obligatorios. • El sistema busca y compara si existe el usuario y contraseña en la base de datos. Si los datos son correctos, el sistema muestra la interfaz de la página principal. • Si los datos ingresados son incorrectos o no existen, el sistema muestra un mensaje" Error de Usuario o contraseña" en un cuadro de color rojo. • El sistema restringe la cantidad de caracteres del campo usuario, mostrando un mensaje de color rojo "como máximo 6 caracteres".	

Figura 21

Descripción de Historia de Usuario "Actualizar Perfil de Colegiado".

Historia de usuario	
Número: H_03	Usuario: Colegiado, cajero
Nombre de historia: Actualizar perfil de colegiado.	
Prioridad: Media	Riesgo: Media
Puntos estimados: 10	Iteración asignada :2
Programador responsable: Programador 1	
Descripción: • El administrador hace clic en la opción "guardar". • El sistema actualiza los campos modificados. • El administrador realiza los filtros de los campos del colegiado. • Se percibe en seguida el cambio realizado.	

Figura 22

Descripción de Historia de Usuario "Generar Reporte de Págo".

Historia de usuario	
Número: H_05	Usuario: Colegiado, cajero
Nombre de historia: Generar reporte de pago.	
Riesgo: Baja	Riesgo: Media
Puntos estimados: 10	Iteración asignada :2
Programador responsable: Judith Vilca	
Descripción: • El administrador o cajero hace clic en la opción "imprimir" de la interface pagos, el sistema muestra un archivo PDF de los pagos realizados por el colegiado. • El reporte detalla el nombre de colegiado, el código de CIP, la fecha de inscripción, la especialidad, la fecha de pago, la hora, el monto de pago y el código QR.	

Figura 23

Descripción de Historia de Usuario "Filtrar los datos del colegiado".

Historia de usuario	
Número:H_04	Usuario: Colegiado, cajero
Nombre de historia: Filtrar los datos del colegiado	
Riesgo: Baja	Riesgo: Baja
Puntos estimados: 10	Iteración asignada: 2
Programador responsable: Programador 1	
Descripción: • El administrador puede filtrar por todos los campos del colegiado. • El administrador ingresa cualquier dato del atributo del colegiado que desea filtrar, en seguida se muestra los todos los demás campos que le corresponden al colegiado con respecto al valor buscado.	

Figura 24

Descripción de Historia de Usuario "Seleccionar el Tipo de Págo".

Historia de usuario	
Número: H_11	Usuario: Cajero
Nombre de historia: Seleccionar el tipo de pago.	
Riesgo: Baja	Riesgo: Media
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Programador 2	
Descripción: • El colegiado selecciona la pestaña "Colegiado Pago" • El sistema muestra la interfaz "Mis pagos". • El colegiado ingresa el tipo de pago seleccionando en el combo box las opciones de: "por cuota" o" por certificado" establecido como campo obligatorio. • El sistema muestra una tabla con los campos: fecha de pago, estado de pago y tipo de pago. • El colegiado al seleccionar el tipo de pago, el sistema lista los pagos correspondientes.	

III. Sprint 1

Se efectuarán los siguientes eventos: el Sprint Planning, el Sprint Backlog, el Daily Scrum, el Sprint Review y el Sprint Restrospective.

A. Sprint Planning

En este segmento se determinó la duración de los sprint del proyecto, el límite de trabajo en curso y las metas a alcanzar.

Previo debate del equipo Scrumban, se determinó que el Sprint 1 debe ser definido por 1 semana y 6 días de duración.

Se determinó respetar la capacidad de 2 tareas desarrolladas en el proceso “Haciendo”.

Metas del Primer Sprint:

- ✓ El colegiado será capaz de realizar el págo, seleccionando el tipo modalidad para cancelar el monto determinado.
- ✓ El colegiado será capaz de consultar sus págos y generar el reporte de este.
- ✓ El reporte de págo generará un código QR para comprobar el estado de habilidad del colegiado.

B. Sprint Backlog

Tabla 19

Sprint Backlog completado del Sprint 1

				Completado Diario		Pendientes Sprint		
Día	ID	Historias de Usuario	Responsable	Estimado	Real	Estimado	Real	Objetivo
0				0	0	0	0	130
1	H_06	Realizar págo	Programador 1	10	6	10	6	130
2	H_06	Realizar págo	Programador 1	10	9	20	15	130
3	H_06	Realizar págo	Programador 1	10	13	30	28	130
4	H_06	Realizar págo	Programador 1	10	8	40	36	130
5	H_07	Generar reporte de págo	Judith Vilca A.	10	8	50	44	130
6	H_07	Generar reporte de págo	Judith Vilca A.	10	12	60	56	130
7	H_08	Consultar págos	Programador 1	10	12	70	68	130
8	H_08	Consultar págos	Programador 1	10	15	80	83	130
9	H_08	Consultar págos	Programador 1	10	9	90	92	130

10	H_09	Escanear código QR	Judith Vilca A	10	10	100	102	130
11	H_09	Escanear código QR	Judith Vilca A	10	7	110	109	130
12	H_10	Seleccionar el tipo de págo	Programador 2	10	10	120	119	130
13	H_10	Seleccionar el tipo de págo	Programador 2	10	11	130	130	130

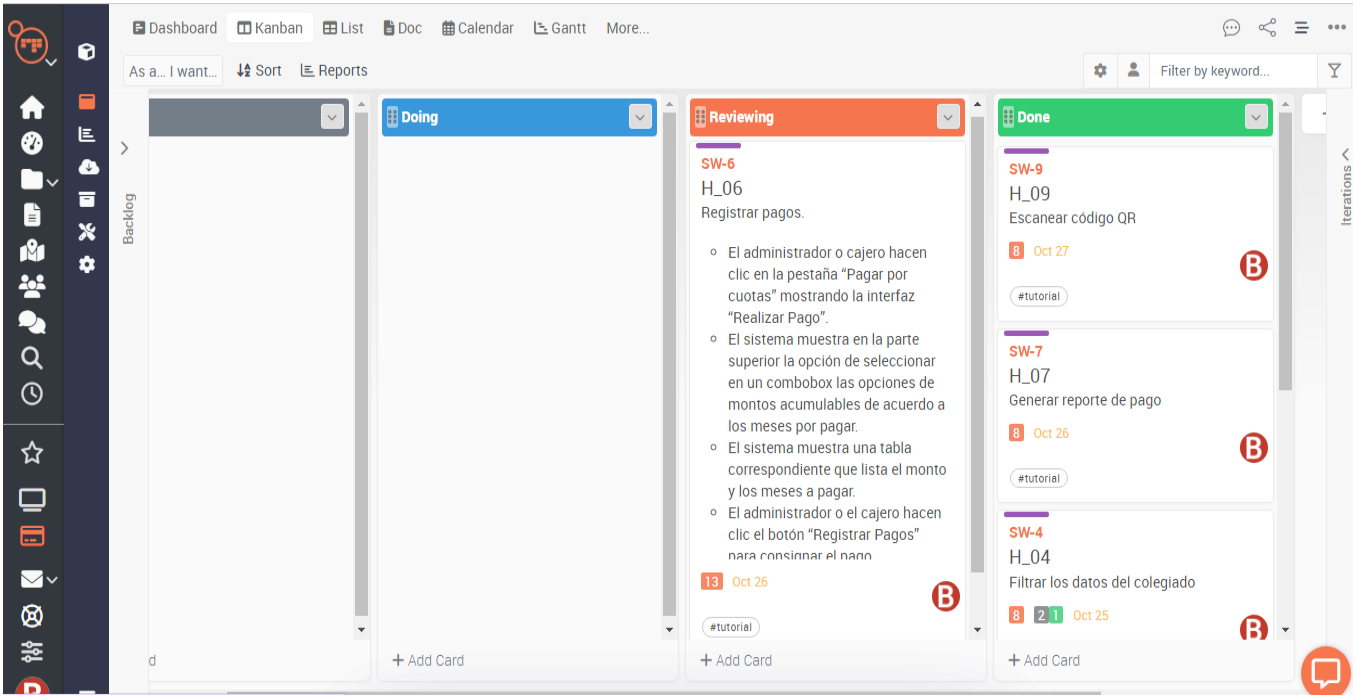
C. Daily Scrum

❖ Comunicación a través del Tablero Kanban

En la figura 25, el tablero Kanban da seguimiento al trabajo y las actividades de este primer sprint backlog para desarrollar los entregables del sprint 1 con un límite de trabajo en curso (WIP) de capacidad 2 en la fase “haciendo”.

Figura 25

Tareas completadas del Sprint 1 en el Tablero Kanban.



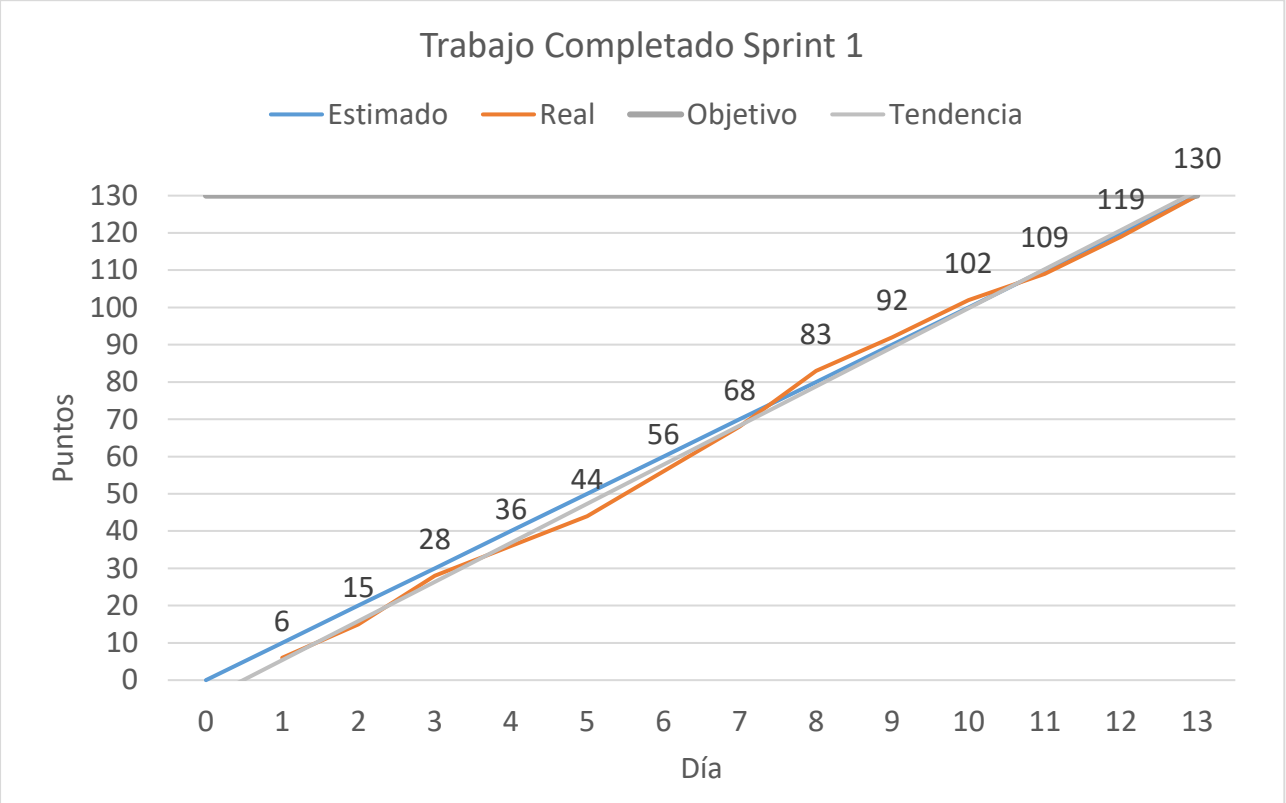
Fuente: ScrumDO

❖ **Trabajando con el cuadro Burndown**

La figura 26, muestra el cuadro burndown o gráfica de progreso para el sprint backlog 1, el cual tuvo una duración de 13 días.

Figura 26

Diagrama burndown para el Sprint backlog 1.



De la figura se puede observar el comportamiento de la gráfica lineal, la cual tiende al crecimiento, debido a las reuniones diarias que permitieron la evaluación del alcance final y la minimización de retrasos, logrando de esta manera alcanzar el objetivo en el tiempo establecido.

D. Resultados

El equipo de Scrumban muestra los resultados del primer sprint a lo largo del Sprint Review. Los entregables son el resultado del desarrollo de las tareas.

Figura 27

Interfaz Vista Lista de Págos.

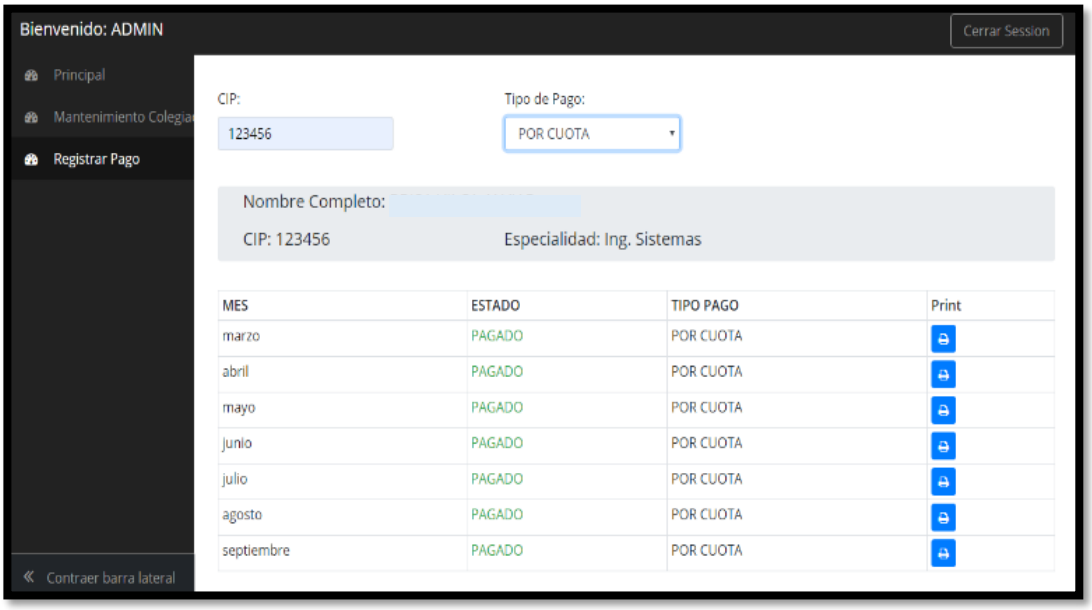


Figura 28

Interfaz "Realizar Págo" por cuota.

E. Sprint Review

El resumen de las tareas relacionadas con el Sprint 1 quedaría de la siguiente forma:

Tabla 20

Resumen de las tareas relacionadas con el Sprint 1.

ID	Historias de Usuario	Resultado
H_06	Realizar págo	Satisfactorio
H_07	Generar reporte de págo	Satisfactorio
H_08	Consultar págos	Satisfactorio
H_09	Escanear código QR	Satisfactorio
H_10	Seleccionar el tipo de págo	Satisfactorio

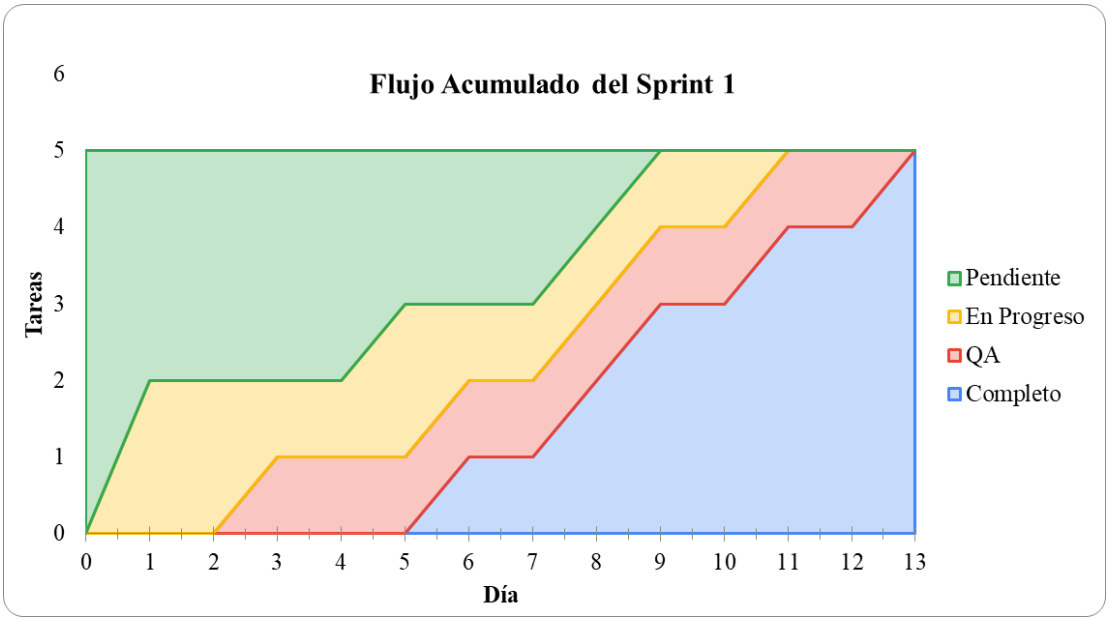
F. Sprint Retrospective

En esta reunión se debatió las lecciones aprendidas durante este primer sprint y la mejora continua para el siguiente.

Para una mejor comprensión del desarrollo del proyecto, se vió por conveniente respaldarse en el Flujo Acumulativo que nos da un panorama general, esta gráfica relaciona el tiempo y los lanzamientos de las tareas, mostrando el estado acumulativo de las tareas programadas.

Figura 29

Flujo Acumulativo del Sprint 1



De la figura se puede observar el estado acumulado de las tareas programadas, el cual tiende al crecimiento, cumpliendo con las tareas pendientes, en progreso, en evaluación y logrando completar las tareas en el tiempo programado. Así mismo, se pudo identificar que el proceso se llevó con fluidez, ya que no se generó cuellos de botellas ni retrasos en el proceso.

IV. Sprint 2

Se efectuarán los siguientes eventos: el Sprint Planning, el Sprint Backlog, el Daily Scrum, el Sprint Review y el Sprint Restrospective.

A. Sprint Planning

Previo debate del equipo Scrumban, se determinó que el Sprint 2 debe ser definido por 1 semana y 3 días de duración. Así mismo, se determinó respetar la capacidad de 2 tareas desarrolladas en el proceso “Haciendo”.

Metas del segundo sprint:

- ✓ El colegiado será capaz de registrarse e iniciar sesión para validar su identidad, así mismo debe poder actualizar sus datos personales.
- ✓ El sistema debe ser capaz de actualizar los págos y los registros de los colegiados para generar los repostes respectivo

B. Sprint Backlog

Figura 30

Sprint Backlog del Sprint 2.

				Completado Diario		Completados Sprint		
Día	N°	Historias de Usuario	Responsable	Estimado	Real	Estimado	Real	Objetivo
0				0		0		100
1	H_01	Registrar Colegiado	Programador 2	10	6	10	6	100
2	H_01	Registrar Colegiado	Programador 2	10	9	20	15	100
3	H_01	Registrar Colegiado	Programador 2	10	11	30	26	100
4	H_02	Iniciar sesión	Judith Vilca A.	10	12	40	38	100
5	H_02	Iniciar sesión	Judith Vilca A.	10	12	50	50	100
6	H_03	Actualizar perfil	Programador 1	10	10	60	60	100
7	H_05	Generar reporte de págo por día	Judith Vilca A.	10	12	70	72	100
8	H_05	Generar reporte de págo por día	Judith Vilca A.	10	10	80	82	100
9	H_04	Filtrar los datos del colegiado	Programador 1	10	9	90	91	100
10	H_11	Generar reporte general de colegiados.	Judith Vilca A.	10	9	100	100	100

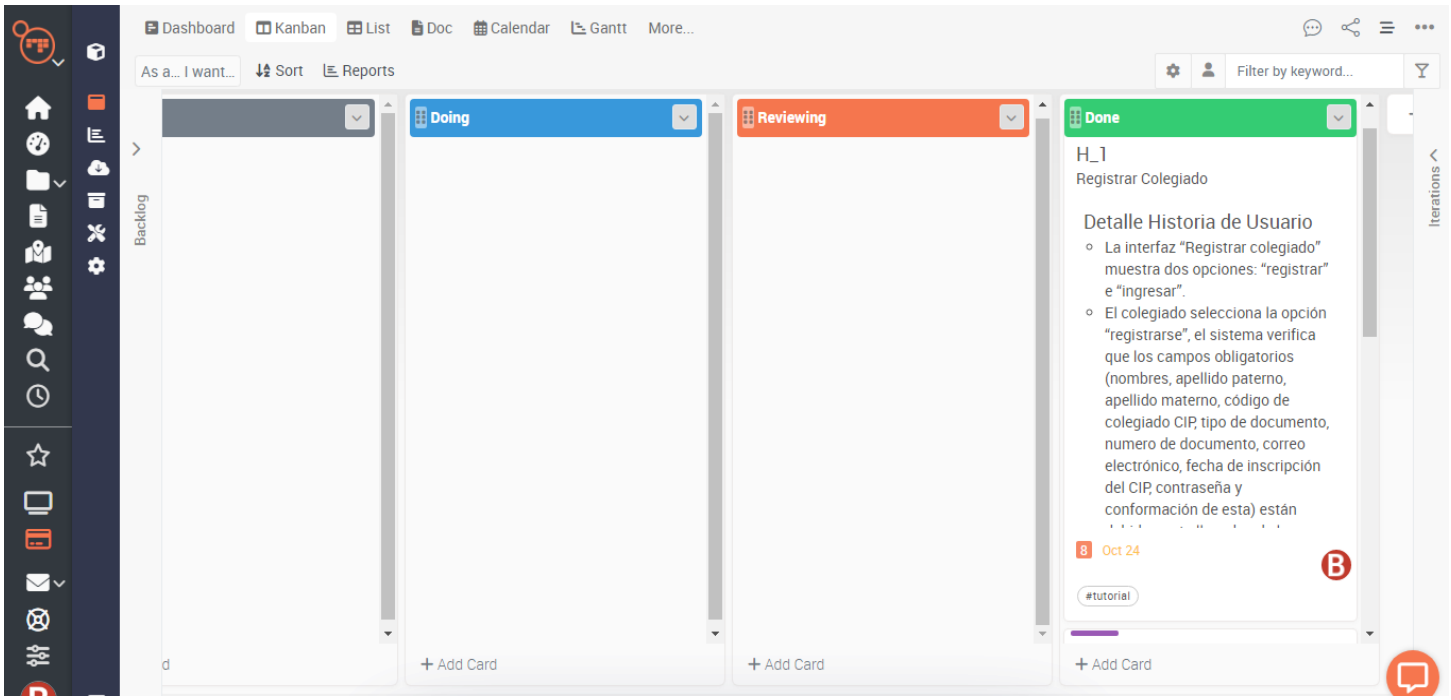
C. Daily Scrum

a. Comunicación a través del Tablero Kanban

En la figura 31, el tablero Kanban da seguimiento al trabajo y las actividades de este segundo sprint backlog para desarrollar los entregables del sprint 2 con un límite de trabajo en curso (WIP) de capacidad 2 en la fase “haciendo”.

Figura 31

Tareas completadas del Sprint 2 en el Tablero Kanban

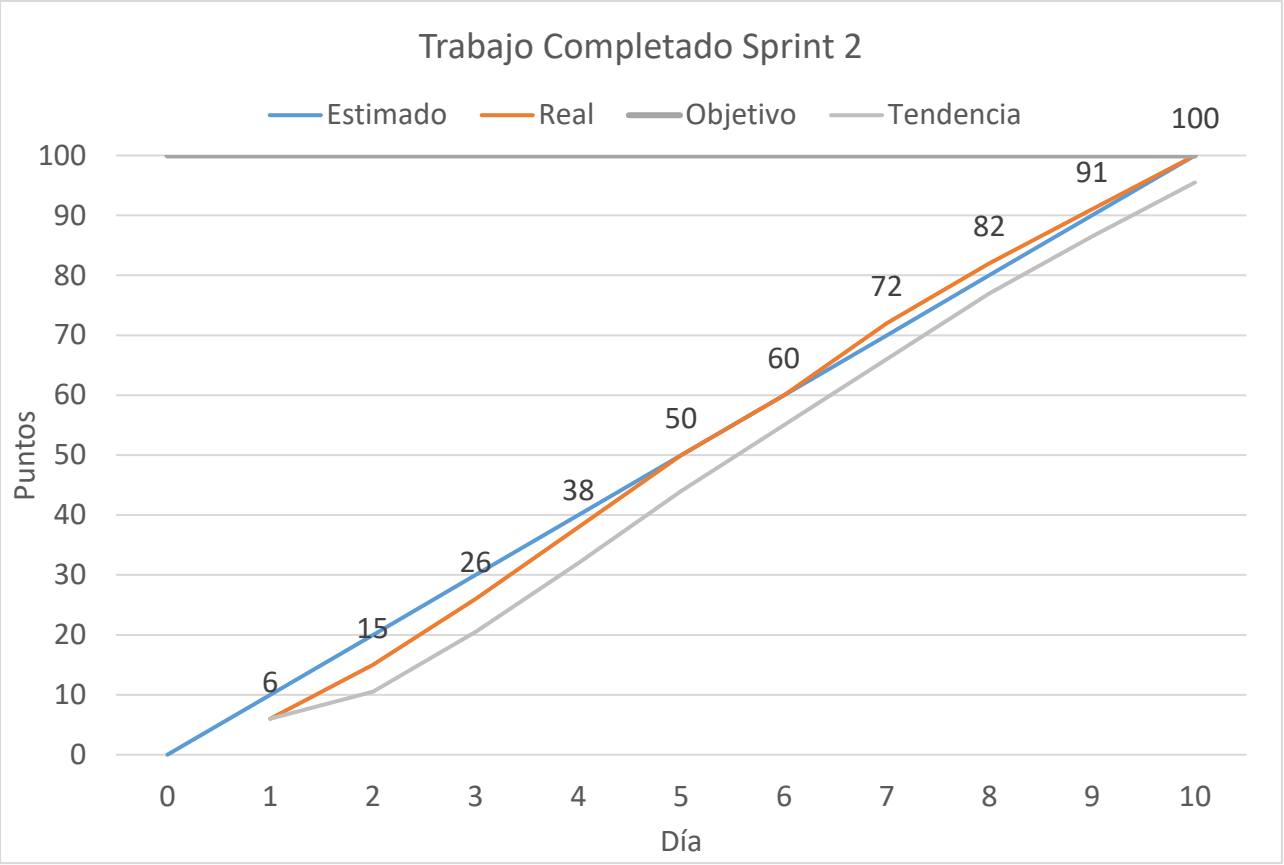


b. Trabajando con el cuadro Burndown

La figura 32 muestra el cuadro burndown o grafica de progreso para el sprint backlog 2, el cual tuvo una duración de 10 días.

Figura 32

Diagrama burndown para el Sprint backlog 2.



De la figura se puede observar el comportamiento de la gráfica lineal, la cual tiende al crecimiento, debido a las reuniones diarias que permitieron la evaluación del alcance final y la minimización de retrasos, el estimado y lo real no están tan alejado, quiere decir que se logró cumplir con las tareas programadas logrando de esta manera alcanzar el objetivo en el tiempo establecido de 10 días.

D. Resultados

El equipo de Scrumban muestra los resultados del segundo sprint a lo largo del Sprint Review.

Los entregables son el resultado del desarrollo de la tarea.

Figura 33

Interfaz. Vista registrar colegiado.

Figura 34

Interfaz. Vista iniciar sesión.

E. Sprint Review

El resumen de las tareas relacionadas con el segundo Sprint quedaría de la siguiente forma:

Tabla 21

Resumen de las tareas relacionadas con el Sprint 2

ID	Historias de Usuario	Resultado
H_01	Registrar Colegiado	Satisfactorio
H_02	Iniciar sesión	Satisfactorio
H_03	Actualizar perfil	Satisfactorio
H_05	Generar reporte de págo por día	Satisfactorio
H_04	Filtrar los datos del colegiado	Satisfactorio
H_11	Generar reporte general de colegiados	Satisfactorio

F. Sprint Retrospective

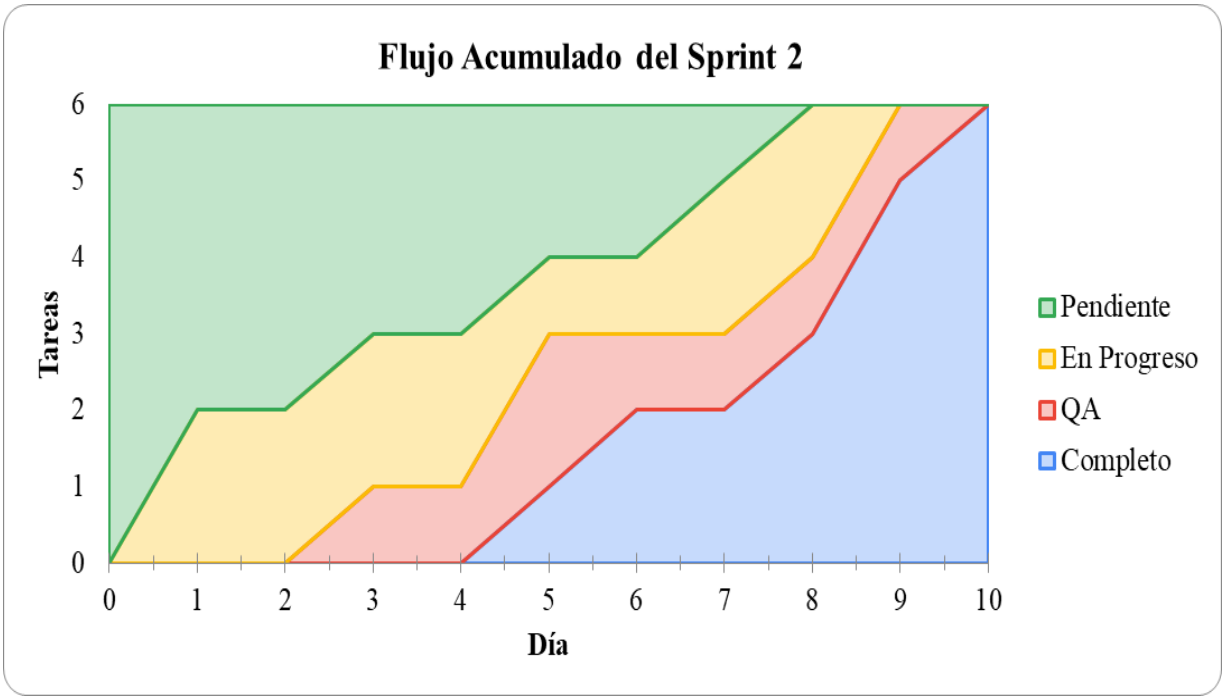
En esta reunión se debatió las lecciones aprendidas durante este segundo sprint y la mejora continua para el siguiente.

El tipo de flujo reflejado en la Figura 35, representa dos variaciones de Scrumban, relacionadas con los sprint y los lanzamientos, en la que se muestra el estado acumulado de las tareas programadas. En este el eje Y representa los incrementos totales de trabajo (días) para el sprint. El equipo puede usar este diagrama para identificar los cuellos de botella, pronosticar el proceso y administrar el alcance.

El Flujo Acumulativo indica retrasos en la entrega o inicialmente, la obtención de aprobaciones, la recepción del trabajo pendiente y eventos similares. La cantidad de trabajo en la cola de revisión aumenta significativamente hacia el final del periodo de tiempo.

Figura 35

Reporte del Flujo Acumulativo



Capítulo V

Conclusiones y Recomendaciones

5.1. Conclusiones

A. Conforme al capítulo II, numeral 2.2.3, Scrumban sobre la Programación Extrema, a través de sus Iteraciones; numeral 2.2.4, Gestión de Proyectos de Software en entornos Ágiles, Estrategia de desarrollo incremental, usando técnicas e instrumentos respaldados en el capítulo III, sección 3.6, se logró comprobar que al aplicar Scrumban a través de iteraciones sobre la programación extrema, efectivamente permite la gestión de proyectos de software en entornos ágiles para adoptar de esta manera una estrategia de desarrollo incremental.

B. Conforme al capítulo II, numeral 2.2.3, Scrumban sobre la Programación Extrema, planificación bajo demanda; numeral 2.2.4, Gestión de Proyectos de Software en entornos Ágiles, calidad de resultado, usando técnicas e instrumentos respaldados en el capítulo III, sección 3.6, se logró comprobar que al aplicar Scrumban a través de una planificación bajo demanda sobre la programación extrema, efectivamente ayuda a la gestión de proyectos de software en entornos ágiles permitiendo mejorar la calidad en el resultado.

C. Conforme al capítulo II, numeral 2.2.3, Scrumban sobre la Programación Extrema, priorización; numeral 2.2.4, Gestión de Proyectos de Software en entornos Ágiles, conocimiento en equipos auto organizados, usando técnicas e instrumentos respaldados en el capítulo III, sección 3.6, se logró comprobar que al aplicar Scrumban a través de la priorización sobre la programación extrema, efectivamente ayuda a la gestión de proyectos de software en entornos ágiles para generar conocimiento en equipos auto organizados.

5.2. Recomendaciones

a) El resultado de esta investigación recomienda a Scrumban para desarrollar y gestionar proyectos de software ya que este marco de trabajo está conformado por las fortalezas de Scrum y Kanban. El resultado también representa que Scrumban necesita de los artefactos y prácticas de la metodología XP para aumentar la robustez del proyecto, así mismo, el uso de la herramienta ScrumDo para administrar de manera eficiente nuestros proyectos.

b) Continuar con la investigación enfocados en aquellos problemas específicos que no pudieron cubrirse usando Scrumban, realizando investigaciones con otras metodologías.

c) Se recomienda enfatizar al dueño del producto, el compromiso en la totalidad del proyecto mediante las reuniones estratégicas de Scrumban.

5.3. Referencia Bibliográficas

Beck, K. & Martin, F. (2004). *Planning Extreme Programming 2da. Edición.* Boston:: Person.

Bedini, A. (2006). *Gestión de Proyectos de Software.*

Bernal, A. (2010). *Metodología de la Investigacíon (6ta Ed.).* Mexico: McGraw Hill Interamericana.

Borja, Y. (2013). *Metodología ágil de desarrollo de software – xp. espe,.* MEVAST.

Cagley, T. (s.f.). *¿Qué hace Scrumban?* Obtenido de Cagley, T. (http://tcagley.wordpress.com/2013/09/24/what-makes-scrumban-scrumban-daily-process-Thoughts /

Carrasco, S. (2019). *Metodología de la investigación científica.Pautas metodológicas para diseñar y elaborar el proyecto de investigacíon.* San Marcos E.I.R.LTDA.

Diaz, G. (2019). *Metodologías de gestion de proyectps de sistemas, una revisión de la literatura científica.* Lima: UNIVERSIDAD PRIVADA DEL NORTE S.A.C.

Flores Leyva, L., & Molina Espinoza, J. (2014). *"Integración de incidentes a la metodología Scrumban para la administración efectiva de Proyectos de TI: El Caso de la Implementación en Sistemas Financieros en Mexico".* México: Escuela de Graduados en Ingeniería y Arquitectura.

Gonzales del Rio, J., & Perez Leal, R. (2015). *Estudio de la Aplicación de las Metodologias Ágiles para proyectos software en el ámbito de las TI.* Leganes: Universidad Carlos III de Madrid.

Hernández , R., Fernández, C., & Baptista, P. (2014). *Metodología de la Investigacíon.* Mexico: McGraw Hill Interamericana.

Kendal, K., & Kendal, J. (2011). *Analisi y Diseño de Sistemas(Octava edicion).* México: Pearson Educación de México, S.A. de C.V.

Kniberg, H., & Skarin, M. (2010). *Kanban y Scrumban-aprovechando al máximo ambos.* Obtenido de C4 me-dia -Publisher of InfoQ: http://www.infoq.com/minibooks/kanban-scrum-minilibro

Leiter , P., & Sanchez, E. (2003). *Metodologías Ágiles en el desarrollo de software.* Alicante: Grupo ISI.

Letelier, P., & Sanchez, E. (2003). *Métodologías Ágiles en el Desarrollo de Software.* Alicante: Grupo ISSI.

Pahuja, S. (Abril de 2012). *¿Qué es Scrumban ?* Obtenido de http://www.solutionsiq.com/resources/agileiq-blog/bid/87799/What-is-Scrumban

Peréz, M. (2011). *Guía Comparativa de Metodologías Ágiles.* España:Universidad de Valladolid: Tesis de Grado en Ingeniería Informática de servicios y aplicaciones.

Pressman. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.Séptima edición.* Mexico: The McGraw-Hill.

Quonext. (12 de Febrero de 2018). Obtenido de https://www.quonext.com/blog/metodologias-ágiles-scrum-kanban-xp/

Radics, S. (2013). *Scrum, Kanban, Scrumban: una descripción general rápida y una categorización aproximación cuándo usar qué método*. Obtenido de http://www.ontheagilepath.net/2013/09/scrum-kanban-scrumban-fast-overview-and.html

Reddy, A. (2016). *The Scrumban [r]evolution : getting the most out of Agile, Scrum, and lean Kanban.* United States of America: Pearson Education,. Obtenido de https://www.pdfdrive.com/the-scrumban-revolution-d19397027.html

Robin, K. (2017). *Essential Scrum.* USA: Pearson Education.

Saiz, J. (24 de Diciembre de 2020). *Jorge Saiz*. Obtenido de https://jorgesaiz.com/blog/scrumban/

Salvay, J. E. (2017). *Kanban y Scrumban.* Córdoba: Instituto Universitario Aeronáutico.

Schwaber, K., & Sutherla, J. (2013). *La Guía de Scrum.* España: Madrid: Pearson Education.

Schwaber, K., & Sutherland, J. (2017). *La Guía Definitiva de Scrum.* España: Madrid: Pearson Education.

Sommerville, I. (2005). *Ingeniería del Software (7ma. Ed.).* Madrid:Pearson Educación S.A.

Sugimori, Y. (1977). 'Sistema de producción Toyota y sistema Kanban: Materialización de just-in-time y respeto por el sistema humano. *Revista internacional de producción*investigación, 553-564.

Yodiz, E. d. (25 de setiembre de 2015). *Yodiz*. Obtenido de Metodología ágil , Kanban , scrumban: https://www.yodiz.com/blog/scrumban-an-amalgamation-of-scrum-and-kanban/

Zumba, J., & Leon, C. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal* .

Bhavsar, K., Shah, V. & Gopalan, S. (2020). Scrumban: An Agile Integration of Scrum and Software Engineering. *International Journal of Innovate Technology and Exploring Engineering (IJITEE), 9*,1226-1634. https://www.ijitee.org/wp-content/uploads/papers/v9i4/D1566029420.pdf

Schwaber, K., & Beedle, M. (2002). Agilè Software Development with Scrum

Sáez, P. (2013). *Identificación y valoración de técnicas ágiles de gestión de proyectos software.* Oviedo: Universidad de Oviedo.

Josep Rodriguez, F. M. (2014). *Desarrollo de componentes con las tecnologías emergentes.*

Lima. Judith, M. M. (2007). *RFID*

Montoya L., Jimenez L. y Sepulveda, J. (2016). *Análisis comparativo de las metodologías ágiles en el desarrollo de software aplicadas en Colombia.* Cimted Corporacion.

Letelier, P. y_ M. Penadés, (2006) Metodologías Ágiles para el Desarrollo de Software: extreme Programming (XP). Investigación. España, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.

Beck, K. (1999) Extreme Programming Explained: Embrace Change. Primera Edición. Reading, Massachusetts, Addison-Wesley.

Z. Bougroun &A. Zeaaraoui & T. Bouchentouf (2016) . Scrumban/XP: un nuevo enfoque para cubrir el tercer nivel del modelo CMMI, Springer International Publishing Suiza, DOI

10.1007 / 978-3-319-30298-0_40

Canty, D. (2015). *Agile for Project Managers.* U.S: Taylor & FrancisGroup.

Cohn, M. (2009). User Stories Applied for Agile Software Development. U.S: Pearson Education Inc.

Salazar, J., Tovar, A., Linares, J., Lozano, A. y Valbuena, L. (2018). Scrum versus XP: similitudes y diferencias. TIA, 6(2), 29-37.

Ladas (2008). Scrumban Essays on Kanban systems for Lean Software Development. Disponible: https://books.google.co.ve/books?hl=es&lr=&id=SQFdAgAAQBAJ&oi=fnd&pg=PA7&dq=Scrumban-Essays+on+Kanban+Systems+for+Lean+Software+Development&ots=c89XRBXGNg&sig=I59JUW5uUe2KDdWgc-LuJnkQKd8#v=onepáge&q=ScrumbanEssays%20on%20Kanban%20Systems%20for%20Lean%20Software%20Development&f=false

Mancl, D., & Fraser, S.D. (2019). XP 2019 Panel: Agile Manifesto – Impacts on Culture, Education, and Software Practices. In: Hoda, R. (ed) Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2019. Lecture Notes in Business Information Processing, vol 364. *Springer*, Cham. https://doi.org/10.1007/978-3-030-30126-2_17

Stol, K. J, & Fitzgerald, B. (2018). The ABC of software engineering research. *ACM Trans Softw Eng Methodol*, 27(3), 1-51. https://doi.org/10.1145/3241743

RV Anand y Dr. M. Dinakaran, "Métodos ágiles populares en el desarrollo de software: revisión y análisis" Revista internacional de avances científicos y técnicos, vol. 2, número 4, págs.

147-150, 2016. ISSN: 2454-1532.

Anexo A

Plantilla de Guia para el análisis de contenido

Guia para el Análisis Documental
"SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022" **I. Finalidad del Instrumento** El presente instrumento tiene como finalidad obtener información referente las variables: "SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022", contempladas en esta investigación.
II. Detalle del contenido Título: Autor: Edición: Editorial: N° de página: Fecha de publicación: Lugar de publicación:
III. Descripción del contenido:

Anexo B

Plantilla para la Historia de Usuario

Figura 36

Tarjeta de Historia de Usuario.

HISTORIA DEL USUARIO	
Número:	***Usuario:***
Nombre de historia:	
Prioridad en negocio:	***Riesgo de desarrollo:***
Puntos estimados:	***Iteración asignada:***
Programador responsable:	
Descripción:	
Observación:	

Fuente: Universidad Politécnica de Valencia (2006).

Anexo C

Plantilla del Backlog del Producto

Figura 37

Plantilla del Backlog del Producto

ID	Historia de usuario	Prioridad

Anexo D

Plantilla del Sprint Backlog

Figura 38

Plantilla del Sprint Backlog General

ID	Historias de Usuario	Responsable	Prioridad	Riesgo	Estimado

Figura 39

Plantilla del Sprint Backlog de Sprint planteados

				Completado Diario		**Pendientes Sprint**		
Día	ID	Historias de Usuario	Responsable	Estimado	Real	Estimado	Real	Objetivo

CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 001-2023-FIMGC

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 281-2022-FIMGC- UNSCH-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Apellidos y Nombres	:	VILCA ALVIAR, Judith Maribel
Escuela Profesional	:	INGENIERÍA DE SISTEMAS
Título de la Tesis	:	"SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022"
Evaluación de la Originalidad	:	**12 %** Índice de Similitud
Identificador de la entrega	:	1987985421

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 02 de enero del 2023

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
Facultad de Ingeniería de Minas, Geología y Civil

Firmado digitalmente por LEZAMA CUELLAR CHRISTIAN

Mg. Ing. Christian LEZAMA CUELLAR
Verificador de Originalidad de Trabajos de Tesis de Pregrado

Con depósito para Sustentación y Tramites
Cc. Archivo

FACULTAD DE INGENIERIA DE MINAS, GEOLOGIA Y CIVIL
Av. Independencia S/N Ciudad Universitaria
Central Tel. 066 312510
Anexo 151

ACTA DE SUSTENTACIÓN DE TESIS

ACTA N° 001-2023-FIMGC

En la ciudad de Ayacucho, en cumplimiento a la **RESOLUCIÓN DECANAL N° 019-2023-FIMGC-D**, siendo los once días del mes de enero del 2023, a horas 8:00 am.; se reunieron los jurados del acto de sustentación, en el Auditórium virtual google meet del Campus Universitario de la Universidad Nacional de San Cristóbal de Huamanga.

Siendo el Jurado de la sustentación de tesis compuesto por el presidente el **Dr. Ing. Efraín Elías PORRAS FLORES,** Jurado el **Mg. Ing. Eloy VILA HUAMAN,** Jurado el **Ing. José Antonio GUERRERO HINOSTROZA,** Jurado - Asesor el **Mg. Ing. Hubner JANAMPA PATILLA** y secretario del proceso el **Mg. Ing. Christian LEZAMA CUELLAR,** con el objetivo de recepcionar la sustentación de la tesis denominada **"SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022"**, presentado por la señorita, **Judith Maribel VILCA ALVIAR,** Bachiller en **Ingeniería de Sistemas**.

El Jurado luego de haber recepcionado la sustentación de la tesis y realizado las preguntas, el sustentante al haber dado respuesta a las preguntas, y el Jurado haber deliberado; califica con la nota aprobatoria de **15 (Quince).**

En fe de lo cual, se firma la presente acta, por los miembros integrantes del proceso de sustentación.

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS GEOLOGÍA Y CIVIL
Dr. Efraín Elías Porras Flores
DECANO

Firmado digitalmente por Efraín Elías Porras Flores
Fecha: 2023.01.16 14:45:35 -05'00'

Dr. Ing. Efraín Elías PORRAS FLORES
Presidente

Mg. Ing. Eloy VILA HUAMAN
Jurado

Mg. Ing. Hubner JANAMPA PATILLA
Jurado Asesor

Ing. José Antonio GUERRERO HINOSTROZA
Jurado

Firmado digitalmente por LEZAMA CUELLAR CHRISTIAN

Mg. Ing. Christian LEZAMA CUELLAR
Secretario del Proceso

c.c.:
Bach. Judith Maribel VILCA ALVIAR
Jurados (4)
Archivo

FACULTAD DE INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL
Av. Independencia S/N
Ciudad Universitaria
Central Tel 066 312510
Anexo 151

“SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022”

por Judith Maribel Vilca Alviar

Fecha de entrega: 02-ene-2023 11:02a.m. (UTC-0500)
Identificador de la entrega: 1987985421
Nombre del archivo: Tesis_Judith_Maribel_Vilca_Alviar_EPIS.pdf (1.64M)
Total de palabras: 19137
Total de caracteres: 111028

“SCRUMBAN SOBRE LA PROGRAMACIÓN EXTREMA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN ENTORNOS ÁGILES, 2022”

INFORME DE ORIGINALIDAD

12%	11%	2%	6%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Nacional de San Cristóbal de Huamanga Trabajo del estudiante	4%
2	hdl.handle.net Fuente de Internet	1%
3	sedici.unlp.edu.ar Fuente de Internet	1%
4	repositorio.upn.edu.pe Fuente de Internet	1%
5	www.risti.xyz Fuente de Internet	1%
6	rdu.iua.edu.ar Fuente de Internet	1%
7	repositorio.unsch.edu.pe Fuente de Internet	<1%
8	documentop.com Fuente de Internet	<1%

9	repositorioacademico.upc.edu.pe Fuente de Internet	<1 %
10	repositorio.upao.edu.pe Fuente de Internet	<1 %
11	es.scribd.com Fuente de Internet	<1 %
12	dadospdf.com Fuente de Internet	<1 %
13	dspace.sheol.uniovi.es Fuente de Internet	<1 %
14	qdoc.tips Fuente de Internet	<1 %
15	www2.deloitte.com Fuente de Internet	<1 %
16	Submitted to Unviersidad de Granada Trabajo del estudiante	<1 %
17	dspace.aeipro.com Fuente de Internet	<1 %
18	es.slideshare.net Fuente de Internet	<1 %
19	repositorio.autonoma.edu.co Fuente de Internet	<1 %
20	www.soe.uagrm.edu.bo Fuente de Internet	<1 %

21	María Consuelo Franky. "Agile management and development of software projects based on collaborative environments", ACM SIGSOFT Software Engineering Notes, 2011 Publicación	<1 %
22	Submitted to Universidad Cientifica del Sur Trabajo del estudiante	<1 %
23	dspace.unitru.edu.pe Fuente de Internet	<1 %
24	www.eaeprogramas.es Fuente de Internet	<1 %
25	www.scrum.org Fuente de Internet	<1 %
26	pdfcoffee.com Fuente de Internet	<1 %

Excluir citas	Activo	Excluir coincidencias	< 30 words
Excluir bibliografía	Activo		