



**UNIVERSIDAD PRIVADA TELESUP**

**FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERIA DE  
SISTEMAS**

**TESIS**

**“METODOLOGÍAS ÁGILES PARA LA OPTIMIZACIÓN  
EN LA DIRECCIÓN DE PROYECTOS DE SOFTWARE EN  
LA EMPRESA BITS & SYSTEMS SOLUTIONS S.A.C –  
LIMA – 2017”**

**PARA OBTENER EL TÍTULO DE:**

**INGENIERO DE SISTEMAS**

**AUTOR:**

**BACH. CARLOS ENRIQUE SOTO CONDORI**

**LIMA- PERÚ**

**2018**

## **ASESOR DE TESIS**

.....  
**Ing. Wilver Auccahuasi Aiquipa**  
**Asesor**

# **JURADO EXAMINADOR**

---

**Presidente**

---

**Secretario**

---

**Vocal**

## **DEDICATORIA**

**A Dios por todas las bendiciones en mi vida, a mis padres por el incansable esfuerzo en brindarme la educación, formarme con valores dentro de un hogar lleno de amor y a mi compañera de vida Carol que estuvo conmigo en todo momento.**

## **AGRADECIMIENTO**

**A Dios por haberme dado la vida y cuidarme cada instante de ella; a los docentes por todo el asesoramiento y haber colaborado con aportes que tuvieron particular relevancia en el desarrollo de la investigación. A la Universidad Privada Telesup por brindarme toda la facilidad para la culminación del proyecto.**

## RESUMEN

La tesis planteada busca mejorar los procesos en la metodología de trabajo dentro de la empresa Bits & System Solution (BSS), en base a las metodologías ágiles como: XP (Xtreme Programing) y Scrum, la empresa Bits & System Solution no cuenta con un esquema estandarizado para dirigir y controlar los proyectos de software que desarrolla, esto ocasiona el retraso de los entregables de los proyectos a los clientes y el aumento del costo en las actividades que se realizan. Se propone plantear un marco de referencia para la dirección de proyectos de desarrollo de software en la empresa Bits & System Solution, con el fin de encontrar soluciones a los problemas antes mencionados, mejorando los resultados que obtiene la empresa a través de sus proyectos de desarrollo de software. Finalmente, se realizó el desarrollo metodología de trabajo para la dirección de proyectos de desarrollo de software basado en las metodologías ágiles SCRUM y XP, que en conjunto se fusionan para proponer una alternativa de solución a los problemas identificados, la misma que contribuirá a optimizar la gestión de desarrollo de software dentro de las restricciones de tiempo, costo; que permita obtener la satisfacción del cliente.

**Palabras Clave:** Gestión de proyectos, desarrollo de software, metodología ágil, SCRUM y XP.

## **ABSTRACT**

The thesis proffer search improve on the process in the methodology of work inside at the enterprise Bits & System Solution (BSS), take to model the work what use agile methodologies as: XP (Xtreme Programing) and Scrum, BSS don't have a standardized methodology to steer and verify the projects what have, helped so at accomplish the objectives and successful completion. Propose a guideline of Project Management for software development in Bits & System Solution. company, providing solutions to problems that exist when developing a software project. Finally, it became the foundation guide for the direction of software development projects based on the agiles methodologies SCRUM and XP, merge together to propose an alternative to solve the problems identified, this will contribute to optimize the management of software development within the constraints of time, cost; to obtain customer satisfaction.

**Key Words:** Project management, software development, agile methodologies, SCRUM, XP.

# INDICE DE CONTENIDOS

CARÁTULA.....	I
ASESOR DE TESIS.....	II
JURADO EXAMINADOR.....	III
DEDICATORIA .....	IV
AGRADECIMIENTO .....	V
RESUMEN .....	VI
ABSTRACT.....	VII
INDICE DE TABLAS.....	XI
INDICE DE FIGURAS .....	XII
INTRODUCCIÓN .....	XIII
<b>I. PROBLEMA DE INVESTIGACIÓN .....</b>	<b>14</b>
<b>1.1. Planteamiento del Problema .....</b>	<b>14</b>
<b>1.2. Formulación del Problema .....</b>	<b>16</b>
1.2.1. Problema General .....	16
1.2.2. Problemas Específicos.....	16
<b>1.3. Justificación y Aportes del Estudio .....</b>	<b>16</b>
<b>1.4. Objetivos de la Investigación.....</b>	<b>18</b>
1.4.1. Objetivo General.....	18
1.4.2. Objetivos Específicos .....	18
<b>II. MARCO TEÓRICO .....</b>	<b>19</b>
<b>2.1. Antecedentes de la Investigación.....</b>	<b>19</b>
2.1.1. Antecedentes Nacionales .....	19
2.1.2. Antecedentes Internacionales.....	23

<b>2.2. Bases Teóricas de las Variables</b> .....	28
2.2.1. Métodos Ágiles.....	28
2.2.2. Dirección de Proyectos de Software.....	46
<b>2.3. Definición de Términos Básicos</b> .....	49
2.3.1. Dirección de Proyectos.....	49
2.3.2. Dirección de Proyectos de Software.....	49
2.3.3. Scrum.....	50
2.3.4. Sprint.....	50
2.3.5. Extremme Programming (XP).....	50
2.3.6. Entrenador (Coach).....	50
2.3.7. Encargado Seguimiento(Tracker).....	51
2.3.8. Encargado Pruebas (Tester).....	51
<b>III. MÉTODOS Y MATERIALES</b> .....	<b>52</b>
<b>3.1. Hipótesis de la Investigación</b> .....	52
<b>3.2. Variables de Estudio</b> .....	52
3.2.1. Definición Conceptual.....	52
3.2.2. Definición Operacional .....	53
<b>3.3. Tipo y Nivel de Investigación</b> .....	54
<b>3.4. Diseño de la Investigación</b> .....	54
<b>3.5. Población y Muestra de Estudio</b> .....	55
3.5.1. Población.....	55
3.5.2. Muestra .....	55
3.5.3. Muestreo .....	55
<b>3.6. Técnicas e Instrumentos de Recolección de Datos</b> .....	56
3.6.1. Técnicas de Recolección de Datos .....	56
3.6.2. Instrumentos de Recolección de Datos .....	56
3.6.3. Validez del Instrumento .....	57
<b>3.7. Métodos de Análisis de Datos</b> .....	57
<b>IV. RESULTADOS</b> .....	<b>58</b>
<b>4.1. Resultados del Procesamiento Estadístico del Instrumento</b> .....	58

4.2. <i>Contrastación de la Hipótesis</i> .....	64
V. DISCUSIÓN .....	67
VI. CONCLUSIÓN .....	69
VII. RECOMENDACIÓN .....	71
REFERENCIA BIBLIOGRÁFICA.....	73
ANEXOS .....	75
1. <i>Matriz de Consistencia</i> .....	75
2. <i>Matriz de Operacionalización</i> .....	76
3. <i>Validación de Instrumentos de Recolección de Datos</i> .....	77
4. <i>Matriz de Datos</i> .....	83
5. <i>Propuesta de Valor</i> .....	84

## INDICE DE TABLAS

<b>Tabla 1.</b> <i>Principales empresas que siguen una Metodología Ágil.....</i>	<b>28</b>
<b>Tabla 2.</b> <i>Compatibilidad SCRUM - XP .....</i>	<b>44</b>
<b>Tabla 3.</b> <i>Compatibilidad Prácticas Técnicas SCRUM - XP .....</i>	<b>45</b>
<b>Tabla 4.</b> <i>Matriz de operacionalización de la variable .....</i>	<b>53</b>
<b>Tabla 5.</b> <i>Validez del Instrumento.....</i>	<b>57</b>
<b>Tabla 6.</b> <i>Primera Medición CPI .....</i>	<b>58</b>
<b>Tabla 7.</b> <i>Segunda Medición CPI .....</i>	<b>59</b>
<b>Tabla 8.</b> <i>Puntuación de requerimientos para la elección - CPI .....</i>	<b>60</b>
<b>Tabla 9.</b> <i>Primera Medición SPI .....</i>	<b>61</b>
<b>Tabla 10.</b> <i>Segunda Medición SPI.....</i>	<b>62</b>
<b>Tabla 11.</b> <i>Puntuación de requerimientos para la elección - SPI.....</i>	<b>63</b>
<b>Tabla 12.</b> <i>Historia del Usuario.....</i>	<b>89</b>
<b>Tabla 13.</b> <i>Tabla Burn-Down .....</i>	<b>90</b>
<b>Tabla 14.</b> <i>Dirigir el Proyecto de Desarrollo de Software .....</i>	<b>94</b>

## INDICE DE FIGURAS

<b>Figura 1:</b> Perú: Empresas por Tipo de Inversión en Ciencia y Tecnología, según Actividad económica, 2014.....	15
<b>Figura 2.</b> Scrum: Procesos del Scrum.....	37
<b>Figura 3.</b> “8 Steps to Scrum”.....	39
<b>Figura 4.</b> Características de las Metodologías Ágiles y Tradicionales.....	41
<b>Figura 5.</b> Actividades PMI y Herramientas de Desarrollo Ágil .....	41
<b>Figura 6.</b> T Student – Máxima Demanda CPI.....	65
<b>Figura 7.</b> T Student – Máxima Demanda SPI.....	66
<b>Figura 8.</b> Beneficios de la Metodología .....	86
<b>Figura 9.</b> Gráfico Burn-Down.....	91
<b>Figura 10.</b> Tablero Scrum.....	92
<b>Figura 11.</b> Supervisión Programación XP.....	93
<b>Figura 12.</b> Retrospectiva Sprint.....	93
<b>Figura 13.</b> Diagrama Dirección del Proyecto.....	96

## INTRODUCCIÓN

Desde las últimas décadas del siglo XX estamos sometidos a un conjunto de transformaciones económicas-sociales y culturales, cuya complejidad no admite precedentes y nuestro país Perú no se encuentra ajeno a estos eventos.

Teniendo en cuenta la nueva escena sociocultural en el Perú y la creciente industria, economía y sobre todo Tecnología, esta última se volvió indispensable para todas las empresas y organizaciones que buscan mantenerse a la vanguardia y ser competitivas en este “nuevo mercado” necesitan Softwares que sean de Calidad, cumplan con todos los requerimientos de la organización y se Adapten a los cambios en el tiempo.

Efectivamente, estos Softwares tan buscados, necesitan contar con un trabajo arduo, dinámico y cualificado que se refleja en el proyecto, donde la etapa más complicada y que determinará la Calidad y resultado final es la Dirección con la que se maneje el Proyecto, por este motivo la presente Tesis busca unir esta necesidad básica y fundamental a través de una metodología Ágil que permita de forma Dinámica, Adaptable y Rápida conseguir la Calidad en el software tan buscada y requerida por todas las organizaciones.

En el capítulo I, Se presenta el problema el cual es que la empresa no cuenta con una metodología de trabajo para poder dirigir los proyectos de software, en el capítulo II, Se presenta el Marco teórico en donde ubicamos los antecedentes de los investigadores nacionales e internacionales referentes al tema de investigación, en el capítulo III, Describe el Marco Metodológico el cual se ubica la investigación y el tratamiento de los datos la descripción de la hipótesis, variables, población y muestra, las técnicas de recolección de datos, y los métodos de análisis descripción de la situación actual y nuestra propuesta de valor entre otros, en el capítulo IV, Se muestran los Resultados de la implementación del sistema web para la proyección de la máxima demanda en la cual se estará presentando el análisis e interpretación de los datos obtenidos a través de la matriz de observación, en el capítulo V, Encontramos la discusión de los datos de investigación, en el capítulo VI, Encontramos las conclusiones de la tesis, en el capítulo VII, Se muestra las referencias de las tesis relacionadas al tema de investigación.

# **I. PROBLEMA DE INVESTIGACIÓN**

## **1.1. Planteamiento del Problema**

Se observó que en la empresa Bits & Systems Solutions se trabaja de forma rutinaria y poco organizada ya que esta no cuenta con una metodología de trabajo estandarizada y formal para la dirección y de proyectos de desarrollo de software, perjudicando notablemente al avance de los proyectos, calidad en los resultados, pérdida de recursos e insatisfacción de los clientes.

El retraso que se genera en los proyectos para terminar los entregables acordados con los clientes provenientes de una mala dirección de proyectos genera pérdida de recursos y la insatisfacción por parte de los clientes quienes perciben un mal servicio.

La dirección de proyectos en la empresa Bits & Systems Solutions no está basada bajo ningún modelo o esquema de trabajo que permite realizar una correcta gestión a los proyectos, esto es debido a que no se tiene una visión real de los proyectos, por ende, no se refleja una evolución concreta del proyecto, ya que en muchos de estos se usan los recursos sin tener ningún control o seguimiento para poder determinar los tiempos y calidad que finalmente se plasmara en el producto final que es el Software.

El retraso de los entregables de los proyectos a los clientes, muchas veces es por la sobrecarga de trabajo que existe en el personal, se trabaja hasta en 4 proyectos de forma simultánea, sin tener un objetivo claro, o indicadores puntuales que les permitan medir el trabajo de cada uno, esta falta de orientación muchas veces genera duplicidad o ambigüedad laboral, ocasionando que el proyecto de software presentado a los clientes no cumpla con los requerimientos específicos que estos solicitaron.

La falta de comunicación en la empresa a pesar de que es una MYPE, ocasiona que en el desarrollo del proyecto no se establezcan los objetivos claros en cuanto a la funcionalidad del software, y al no haber una supervisión continua a los programadores, estos a su vez que se encuentran desmotivados y abrumados ante la carga laboral, retrasa mucho más el proyecto, la calidad final del software

se ve claramente mermada, ocasionando conflictos con los clientes, siendo muchos de estos clientes importantes y vitales para la empresa.

La motivación del personal es un factor importante, ya que estos al no sentirse comprometidos con la empresa, y ante la evidente falta de liderazgo, muchas veces realizan el trabajo según el criterio personal, sin cumplir con los requerimientos especificados por los clientes.

**PERÚ: EMPRESAS POR TIPO DE INVERSIÓN EN CIENCIA Y TECNOLOGÍA, SEGÚN ACTIVIDAD ECONÓMICA, 2014**  
(Porcentaje)

Actividad económica	Total de la inversión	Patentes y propiedad industrial	Programas de computadoras y software	Costo de exploración y desarrollo	Fórmulas, diseños y prototipos	Reservas de recursos extraíbles	Otros tipos de intangibles	Capacitación en investigación científica, tecnológica y de gestión	Asesoría y consultoría
<b>Total</b>	<b>100,0</b>	<b>3,6</b>	<b>18,0</b>	<b>14,3</b>	<b>0,8</b>	<b>0,0</b>	<b>6,6</b>	<b>1,5</b>	<b>55,2</b>
Agrícola y pesca	100,0	0,8	24,7	2,3	0,0	0,0	9,6	2,5	60,1
Hidrocarburos	100,0	0,5	1,1	90,8	0,0	0,0	5,6	0,2	1,8
Industrias manufactureras	100,0	11,1	13,0	1,4	3,9	0,0	12,2	3,2	55,2
Suministro de electricidad	100,0	0,1	67,0	0,2	0,0	0,0	2,6	1,4	28,7
Suministro de agua y alcantarillado	100,0	0,0	47,3	0,5	0,0	0,0	12,4	0,9	38,9
Construcción	100,0	0,5	6,1	2,0	0,0	0,0	0,8	0,8	89,8
Comercio al por mayor y al por menor	100,0	3,8	15,9	0,7	0,7	0,0	17,6	1,9	59,4
Transporte y almacenamiento	100,0	3,0	13,1	0,2	0,0	0,0	4,1	1,9	77,7
Alojamiento y servicio de comidas	100,0	7,1	21,7	0,7	0,4	0,0	2,8	4,4	62,9
Información y comunicaciones	100,0	2,2	40,8	7,3	0,0	0,0	0,4	0,4	48,9
Actividades inmobiliarias	100,0	0,0	5,7	0,0	0,0	0,0	9,2	1,8	83,3
Actividades profesionales, científicas y técnicas	100,0	1,4	7,9	0,4	0,3	0,0	2,0	1,0	87,0
Actividades de servicios administrativos y de apoyo	100,0	0,8	21,2	0,8	0,4	0,0	11,4	2,1	63,3
Enseñanza privada	100,0	12,4	21,5	2,9	0,2	0,0	3,7	5,1	54,2
Atención de la salud humana y de asistencia social	100,0	0,3	16,5	0,0	0,0	3,9	1,2	0,6	77,5
Actividades artísticas, entretenimiento y recreativas	100,0	0,4	5,0	0,0	1,4	0,0	3,4	1,0	88,8
Otras actividades de servicios	100,0	0,6	5,9	2,7	0,0	0,0	0,0	1,6	89,2

**Nota:** El valor de la inversión corresponde al ejercicio económico 2014.

**Fuente:** Instituto Nacional de Estadística e Informática - Encuesta Económica Anual 2015.

**Figura 1:** Perú: Empresas por Tipo de Inversión en Ciencia y Tecnología, según Actividad económica, 2014

Fuente: Instituto Nacional de Estadística e Informática – Encuesta Económica Anual 2015

## **1.2. Formulación del Problema**

### **1.2.1. Problema General**

¿De qué manera las Metodologías ágiles influye en la optimización de la dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C. Lima, 2017?

### **1.2.2. Problemas Específicos.**

¿De qué manera las metodologías ágiles influyen en la reducción en los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C. Lima, 2017?

¿De qué manera las metodologías ágiles influyen en la optimización del tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C. Lima, 2017?

## **1.3. Justificación y Aportes del Estudio**

La justificación de esta tesis se sustenta principalmente en lo fundamental que es para el país la existencia de organizaciones, empresas o entes no gubernamentales que contribuyan al crecimiento económico y desarrollo del país. Toda sociedad, organización y personas, necesitan fuentes de ingresos para subsistir y desarrollar sus actividades. Un país desarrollado cuenta con diferentes empresas e instituciones organizadas, profesionales capacitados, tecnología y sistemas de punta, además de ciertos indicadores que garantizan el desarrollo.

Actualmente las empresas de desarrollo de software buscan posicionamiento y diferenciación en el mercado. Esto conlleva a que los profesionales se preparen y actualicen constantemente en temas relacionados con la gestión de proyectos del sector de las TIC, con el fin de generar productos y servicios de calidad dándole un valor agregado al cliente, para esto las empresas están realizando mejoras continuas en sus procesos empleando metodologías y estándares; sin embargo muchas organizaciones al tener diferentes alternativas para la gestión de proyectos no deciden cuáles usar, además muchas de la empresas no saben cómo utilizarlas o mejorarlas. Es por esta razón que se propone la mejora de los procesos en la metodología de trabajo que ayude a la empresa Bits & Systems

Solutions.SAC a optimizar la dirección de proyectos de desarrollo de Software, con el fin de que los clientes como la empresa se vean beneficiados.

La presente tesis desarrollará en la empresa Bits & Systems Solutions.SAC tiene como finalidad evaluar y mejorar la dirección de proyectos que se realiza en la empresa, de esta manera solucionar completamente las falencias que puedan ocurrir en la gestión de proyectos ejemplo retrasos en los entregables, insatisfacción de los clientes, falta de compromiso por parte de los trabajadores, además estos proyectos establecen un límite de costo y una fecha de finalización máxima especificada. Se requiere establecer la metodología de trabajo que tiene la empresa la cual será a corto, mediano y largo plazo, se tomará como indicador y avisará que acciones ejecutar para lograr mejores prácticas, de esta manera optimizar la dirección en los proyectos de la empresa.

Al estudiar las Metodologías Agiles, conocer sus buenas prácticas para luego con criterio seleccionar los procesos y recomendaciones más apropiados, estos se ajustaran a la realidad de los proyectos de desarrollo de software de la empresa Bits & Systems Solutions S.A.C., en un esquema ágil, además de esto se realizara un estudio de las principales metodologías ágiles que existe para el desarrollo de los proyectos de software, se discernirá los procesos más importantes y las herramientas que usaremos para la creación de nuestra metodología de trabajo en la dirección de proyecto de desarrollo de software en la empresa Bits & System Solution.

Con este propósito se analizaron casos de estudio y de éxito que permitieron conocer las buenas prácticas y metodologías de dirección de proyectos mundialmente utilizadas en los últimos tiempos. Además, debido a los buenos resultados y la gran acogida que han tenido por su flexibilidad y facilidad de uso, la metodología ágil híbrida SCRUM/XP.

(Gómez Arias et al., 2009); indican “Las grandes decisiones no se toman sin antes no tener un previo estudio que nos indique que, cuando, donde, como y sobre todo porque se deben llevar a cabo las acciones que queremos realizar, este análisis no es otra cosa que el inicio de un proyecto, este es un emprendimiento temporario realizado para crear un producto o servicio único. Temporario ya que tiene un comienzo y un fin definido y único ya que el producto

o servicio es diferente de alguna manera que lo distingue de otros productos o servicios.”

(Romano & Yacuzzi, 2011); señalan que “Los proyectos nos permiten resolver problemas identificados, los cuales de una u otra forma van a mejorar las condiciones de vida del grupo en estudio, además de ello permiten acceder a diversas fuentes de financiación las cuales basan sus decisiones en diseño de estrategias y flujo de recursos que concluidos de la forma correcta van a permitir obtener resultados que nos llevan a la recuperación del capital, en cuanto a gestión, los proyectos nos permiten tener una adecuada forma de gestionar y organizar los recursos invertidos en base a los resultados esperados, todo esto estableciendo una lógica de ejecución.”

(En & Tocto, 2014) señala que los Procesos de Negocio involucra todo un conjunto de actividades ordenadas. Modelar un proceso de negocio te ayuda a tener una visión clara de cómo fluye la información y a cumplir a cabalidad los objetivos estratégicos de la empresa. En el modelado de BPMN, se pueden percibir distintos niveles de modelado de procesos.

(Roberts & Wallace, 2011); señalan que “La dirección del proyecto tiene la responsabilidad total de todos los aspectos técnicos del proyecto - diseño, código, selección de la tecnología, asignación del trabajo, aspectos de programación que bien podría encuadrar como un tipo mitad directivo mitad técnico. Esto significa que tienen que expresarse a dos sectores - los directivos, y el equipo de trabajo.”

#### **1.4. Objetivos de la Investigación**

##### **1.4.1. Objetivo General**

Establecer una metodología ágil que optimice la dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

##### **1.4.2. Objetivos Específicos**

- Establecer una metodología ágil que reduzca los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.
- Establecer una metodología ágil que optimice el tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

## **II. MARCO TEÓRICO**

### **2.1. Antecedentes de la Investigación**

#### **2.1.1. Antecedentes Nacionales**

Las empresas en el Perú están viviendo una bonanza, el crecimiento de casi todas las industrias es el reflejo de una demanda ascendente y del crecimiento de la competitividad. Es por ello, que las empresas de alto rendimiento están optando como ventaja competitiva las metodologías que se adapten mucho mejor a las necesidades del cliente.

Malpica (2014); en la investigación titulada “Aplicación de la metodología Scrum para incrementar la productividad del proceso de desarrollo de software en la empresa CCJ S.A.C. lima”.

Conclusión: La presente tesis intitulada “Aplicación de la Metodología Scrum para incrementar la productividad del proceso de desarrollo de software en la Empresa CCJ S.A.C. Lima”, se ha enfocado en la Unidad de Negocio de Tecnologías de Información de CCJ, una corporación con la visión de servicio en el diseño, desarrollo y ejecución de proyectos en ingeniería. Sin embargo, el modelo con el que viene desarrollando los proyectos de software era adaptado del Ciclo de Vida en Cascada, el cual es inadecuado por su carencia de agilidad y flexibilidad, generando desfases en tiempos y costos. Bajo este contexto, la Metodología Scrum se presenta como una atractiva posibilidad debido a su naturaleza ágil, lo cual implica un carácter adaptable, orientado a las personas más que a los procesos y que emplea la estructura de desarrollo ágil. A diferencia del Ciclo de Vida en Cascada, posee agilidad, flexibilidad permitiendo el incremento de la calidad y la reducción notable de tiempo y costos. El modelo aplicativo Scrum consta de cinco fases: Definición del backlog del producto, Planificación del sprint, Scrum diario, Revisión del sprint y Retrospectiva del sprint. Para su correcta aplicación se comenzó con la visión general del producto, proporcionado por el product owner, ésta información se estructura en el backlog del producto, que contiene los sprint backlogs que son especificaciones funcionales de las partes con mayor prioridad de desarrollo. Estos sprints fueron planificados uno a uno y se llevaron a cabo en un periodo de 1 a 4 semanas mediante reuniones diarias donde participaron el Scrum master y el product owner. Cada periodo de desarrollo atravesó por la revisión del sprint, y al existir

ciertas variaciones respecto al requerimiento inicial se dio la retrospectiva del sprint. Todo el proceso finalizó con la producción de un incremento operativo del producto validado por el cliente, lo que Scrum denomina “potentially shippable”. Con la aplicación de Scrum, se obtuvo como resultado: la reducción del número de días de retraso a cero días, pudiéndose cumplir con los entregables del proyecto en el plazo establecido, no incurriendo en costos adicionales. Además, se logró mejorar el clima laboral con las reuniones diarias que establece Scrum. Concluyendo, se plantea que Scrum al ser un proceso de desarrollo iterativo e incremental se puede usar para cualquier área de una empresa como Tecnologías de Información, Administración, Operaciones, entre otras, ya que sirve para planificar, ordenar, reportar el trabajo del día a día, semanal, mensual, anual. Por lo que, se recomienda implementar Scrum en su empresa, porque permite la creación de equipos auto-organizados impulsando la participación activa de todos los miembros del equipo, y la comunicación verbal entre todos los integrantes y disciplinas involucradas en el proyecto.

Salvador (2013); “Una revisión sistemática de usabilidad en metodologías ágiles”

Conclusión: En los últimos años, se han aplicado técnicas de evaluación de usabilidad en el desarrollo de software. En las metodologías ágiles, éstas técnicas se están considerando, puesto que siempre han propuesto mejorar la calidad del producto. El objetivo de este trabajo de tesis es sintetizar el conocimiento existente referente a los métodos de evaluación de usabilidad que se han aplicado en las metodologías a través de la realización de una revisión sistemática. La estrategia de búsqueda identificó un total de 307 artículos, de los cuales se seleccionaron 32. Los resultados muestran que las técnicas de usabilidad utilizadas con mayor frecuencia son el prototipado+ rápido (40%), la indagación individual (37%), las pruebas formales de usabilidad (25%) y las evaluaciones heurísticas (18%). Los resultados obtenidos han permitido conocer el estado actual de las técnicas de evaluación de usabilidad en metodologías ágiles, esto contribuye a identificar lagunas de investigación, información que pueda ayudar a otros profesionales interesados en el tema de investigación.

Castillo (2016); “Desarrollo e implementación de un sistema web para generar valor en una pyme aplicando una metodología ágil. Caso de estudio: Manufibras Pérez SRL”.

Conclusión: Las empresas en la actualidad se apoyan cada vez más en la tecnología para la mejora de sus procesos y productos. Por lo que la adopción de un sistema web que automatice procesos del negocio, está dejando de ser una alternativa para pasar a ser un requerimiento en las pymes, debido a que tienen que estar adaptándose rápidamente a los cambios que puedan presentarse en su entorno por causa de la alta competencia de los productos que elaboran y el poder competir dentro del mercado. En este contexto, es viable mejorar la situación actual para la pyme de caso de estudio, pues al momento todo es un proceso manual, el cual trae como consecuencia pérdidas económicas por errores manuales y la alta inversión de tiempo en sus actividades. Por lo que el objetivo del presente trabajo es la generación de valor para la pyme, debido a que es importante mejorar la situación económica de las pymes ya que investigaciones previas señalan que son las que aportan un mayor crecimiento al país y son generadoras de empleo. Al finalizar el proyecto se demuestra como con la consecución del sistema para la promoción de productos, gestión de pedidos y registro de ventas, se genera valor para la pyme con la reducción de tiempo, costos operativos y el mejorar el servicio a los clientes, los cuales permitirán que los beneficios sean mayores a la inversión del proyecto. Además, también se comprueba con la revisión de la literatura que estudios previos sobre el desarrollo web inciden en el uso de las metodologías ágiles, las cuales referencian a la Extreme Programming (XP) y Scrum como las más destacadas metodologías ágiles para el desarrollo de software.

Palao (2014); “Modelo de sistema de información de registro y monitoreo socio ambiental participativo del proyecto de exploración minero chucapaca comparando metodologías ágiles scrum y kanban”.

Conclusión: El presente estudio se realizó durante los años 2011 – 2012 en la ciudad de Puno en gerencia Programa de Monitoreo Socio Ambiental Participativo (PMSAP) del Proyecto Minero Chucapaca que se desarrolla en la provincia de Ichuña región Moquegua. Con el objetivo de Implementar un

sistema de registro y procesamiento de datos del Programa de Monitoreo Socio Ambiental Participativo en el proyecto de exploración minera Chucapaca, comparando las metodologías ágiles Scrum y Kanban. La metodología utilizada fue de investigación de tipo cuantitativo - descriptivo y experimental; para la recolección de la información se realizó un cuestionario con preguntas abiertas, las cuales fueron aplicada a los monitores encargados del Programa de Monitoreo Socio Ambiental Participativo (PMSAP) a pedido del Proyecto Minero Chucapaca. Para implementar un sistema de registro y procesamiento de datos del Programa de Monitoreo Socio Ambiental Participativo en el proyecto de exploración minera Chucapaca, se realizó la secuencia de pasos las metodologías ágiles Scrum y Kanban, para posteriormente comparar las metodologías Scrum y Kanban, adoptando lo mejor de ambas, evaluando el nivel de optimización de los resultados del monitoreo del PMSAP en el proyecto de exploración minera Chucapaca. De los resultados obtenidos se ha llegado a la siguiente conclusión: El sistema de registro y procesamiento de datos del Programa de Monitoreo Socio Ambiental Participativo en el proyecto de exploración minera Chucapaca, comparando las metodologías ágiles Scrum y Kanban, logra optimizar los resultados en tiempo y calidad. El análisis, diseño e implementación permitió poner en funcionamiento las funcionalidades requeridas. La comparación de ambas metodologías aplicado en el proceso de desarrollo del sistema. permitió conocer la medida en que Scrum y Kanban son muy adaptables, pero en términos relativos Scrum es más restrictivo que Kanban Como cualquier herramienta, Scrum y Kanban no son ni perfectas ni completas, son herramientas que se deben juzgar a la herramienta por el éxito del trabajo realizado con ella.

Palza (2013); “Desarrollo de una herramienta de soporte a la gestión de proyectos ágiles para equipos distribuidos”.

Conclusión: La industria de software, es una industria globalizada, por esta razón resulta cada vez más común trabajar con equipos distribuidos, en diferentes locaciones geográficas. Estas organizaciones de Tecnologías de Información continuamente tienen que adaptar sus procesos, reducir costos e incrementar la calidad de sus productos. Es por esto que muchas organizaciones optaron por la adaptación de los procesos de desarrollo para que sean ágiles y sencillos. El

presente proyecto de fin de carrera se realiza con el objetivo de proponer una solución informática de soporte a la gestión de proyectos ágiles de desarrollo de software para equipos distribuidos, sin añadirle una burocracia innecesaria. En primer lugar, se presenta un estudio sistemático de la literatura existente, sobre metodologías ágiles, y de las prácticas recomendadas para ambientes distribuidos globalmente. Y también, se presenta una comparación de la tecnología existente que pretende solucionar el problema encontrado. El producto final propuesto, se desarrolló mediante iteraciones continuas, de análisis, diseño e implementación, haciendo uso de prácticas ágiles de desarrollo de software y bajo la supervisión del asesor del proyecto de fin de carrera. La planificación del proyecto se realizó tomando en cuenta los lineamientos de prácticas ágiles. Al inicio del proyecto se realizó una planificación considerando las limitaciones de tiempo con las que se cuenta, pero, la planificación detallada de las actividades en cada iteración se resolvió al inicio de la misma.

### **2.1.2. Antecedentes Internacionales**

Núñez (2010); “Usabilidad en metodologías ágiles”

Conclusión: La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. Es considerada la usabilidad también, como la medida en que un producto puede ser usado por usuarios específicos, para lograr los objetivos especificados con efectividad, eficiencia y satisfacción en un contexto específico de uso. En el campo de la Ingeniería de software, hablar de usabilidad hoy en día, es relacionar con la interfaz de usuario, por lo tanto, la usabilidad afectaría a la UI y no a la funcionalidad básica del sistema. De acuerdo a esto, la usabilidad debe ser tratada al final del desarrollo del sistema, como un atributo más de calidad y que no necesita demasiadas pruebas (Hipótesis de la separación). En contraposición, a la idea que se tiene hoy en día de la usabilidad, existen investigaciones en el campo de la ingeniería de Software, donde, se describe que las Isuues de usabilidad son limitaciones estáticas y dinámicas a los componentes de software y que la separación de la interfaz de usuario de la funcionalidad básica, no garantiza que se entregue un producto usable al cliente final. Confirmada la relación entre diseño de software y usabilidad, la revisión del

costo para alcanzar un nivel aceptable de usabilidad debería ser mayor que lo planteado por la hipótesis de separación, por lo tanto, la facilidad de uso debe ser tratada con anterioridad al proceso de desarrollo, con el fin de evaluar su impacto en el diseño tan pronto como sea posible. Esto es una estrategia ya aplicada con anterioridad a algunos de los atributos de calidad como, por ejemplo: fiabilidad, disponibilidad y mantenibilidad, donde, algunos autores propusieron en sus momentos, técnicas para hacer frente a estos atributos en tiempo de diseño arquitectónico. El proceso de desarrollo de software, hoy en día, se vienen utilizando en proyecto de desarrollo de software tradicional, es decir, proyectos que siguen lineamientos de metodologías de desarrollo tradicional. Es de acuerdo a esta premisa, que el presente trabajo tiene como objetivo principal, la inclusión de los principios de usabilidad, en proyectos Software, que se desarrollen en base a metodologías ágiles. Para cumplir con este objetivo, el presente trabajo de fin de Master, se basará en estudios previos, donde se detallan mecanismo para considerar la usabilidad antes del proceso de desarrollo, es decir, en las etapas, de toma de requisitos y de diseño (ver anexos). Estos mecanismos, deberán de ser tomados en cuenta, en los planteamientos de la metodología o metodologías ágiles seleccionadas, para que así, el ciclo de la vida del software a desarrollar en base a estos lineamientos ágiles, tome en cuenta los principios de usabilidad.

Gimson (2012); “Metodologías ágiles y desarrollo basado en conocimiento”.

Conclusión: Este trabajo busca realizar una investigación bibliográfica tendiente a exponer los fundamentos de diferentes metodologías ágiles propuestas para el desarrollo de sistemas y también realizar paralelamente una recopilación bibliográfica sobre el desarrollo basado en conocimiento. Metodologías ágiles Hace casi dos décadas que se comenzó a buscar una alternativa a las metodologías formales o tradicionales que estaban sobrecargadas de técnicas y herramientas y que se consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo Las metodologías ágiles conllevan una filosofía de desarrollo de software liviana, debido a que hace uso de modelos ágiles. Se considera que un modelo es ágil o liviano cuando se emplea para su construcción una herramienta

o técnica sencilla, que apunta a desarrollar un modelo aceptablemente bueno y suficiente en lugar de un modelo perfecto y complejo. Existen actualmente una serie de metodologías que responden a las características de las metodologías ágiles y cada vez están teniendo más adeptos. Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con sus postulados y principios, cada metodología ágil tiene características propias y hace hincapié en algunos aspectos más específicos. Actualmente se pretende poder desarrollar software en el menor tiempo posible y con el menor costo. Para tratar de reducir el tiempo de programación, la solución no está relacionada tanto en mejorar más todavía los lenguajes de programación sino en la programación en sí. En los desarrollos de sistemas tradicionales se desarrolla y se realiza el mantenimiento con programación manual. Si se "describe" en vez de "programar", se pueden maximizar las descripciones declarativas y minimizar las especificaciones procedurales, haciendo desarrollo basado en conocimiento y no en programación. Esta pretensión constituye un cambio esencial de paradigma e implica un choque cultural. La Base del conocimiento inicialmente tiene asociado un conjunto de mecanismos de inferencia y contiene reglas generales que son independientes de cualquier aplicación particular. Al describir la realidad del usuario objeto se almacenan las descripciones en el Modelo Externo. El sistema, automáticamente, captura todo el conocimiento contenido en el Modelo Externo y lo sistematiza, agregándolo también a la Base del conocimiento. Adicionalmente, sobre el conocimiento anterior, el sistema infiere lógicamente un conjunto de resultados que ayudan a mejorar la eficiencia de las inferencias posteriores. En este tipo de desarrollo el foco está en ocuparse únicamente del Modelo Externo (el "qué") y abstenerse de tratar la Base del Conocimiento, que lo contiene y lo mantiene, (y que forma parte del "cómo").

Izaurralde (2013); "Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario".

Conclusión: El presente trabajo revisa las prácticas de ingeniería para el desarrollo de requerimientos, con foco en la especificación de los requerimientos de software y las metodologías ágiles de desarrollo de software, en particular Scrum. Se analizan las características que poseen los requerimientos

especificados según el estándar IEEE Std. 830-1998 para luego desarrollar una comparación con las historias de usuario. Se efectúa el análisis de la estructura de las historias de usuario, se analizan las ventajas, desventajas y también se describen las características que han sido definidas para ayudar a pensar y escribir historias de usuario. Por último, se revisa el caso en donde los documentos tradicionales de especificación de requerimientos y las historias de usuario coexisten, aun cuando se ha adoptado Scrum como metodología para el desarrollo de software. El mismo tiene por objetivo evaluar el impacto de no establecer mecanismos formales de trazabilidad de requerimientos a lo largo del ciclo de desarrollo y proponer un modelo de trazabilidad de requerimientos en metodologías ágiles. Dado que existen opiniones encontradas respecto de la utilidad de las matrices de trazabilidad de requerimientos de software (Ambler, Agile Requirements Modeling, 2010) (Ramesh & Jarke, 2001), el primer objetivo del plan de tesis de maestría es medir el impacto de no utilizar dichas matrices en proyectos ágiles de gran escala (Ambler, The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments, 2009). Pero para poder dar comienzo a dicho análisis es preciso determinar cómo se especifican los requerimientos en metodologías ágiles como Scrum (Schwabe, 2004) y qué características deben poseer dichas especificaciones para cumplir con las características de un buen requerimiento según la IEEE (Software Engineering Standards Committee of the IEEE Computer Society, 1998) (Software Engineering Standards Committee of the IEEE Computer Society, 1998). En base a lo mencionado anteriormente se plantea como objetivo del presente trabajo determinar los principales atributos y características que debe cumplir un requerimiento ágil, revisando la bibliografía existente sobre técnicas de especificación de requerimientos de software en Scrum (Schwabe, 2004) e identificando las características de un buen requerimiento según IEEE (Software Engineering Standards Committee of the IEEE Computer Society, 1998) (Software Engineering Standards Committee of the IEEE Computer Society, 1998). El trabajo será elaborado en diferentes secciones, una primera sección que definirá los requerimientos de software, la ingeniería de requerimientos y el proceso de especificación de requerimientos de software. Una segunda sección en donde se hará una introducción a las metodologías ágiles, a Scrum (Schwabe, 2004) puntualmente y luego a las historias de usuario. Una tercera sección donde

se hará una revisión de la bibliografía existente acerca de la práctica de especificación de requerimientos en Scrum (Schwabe, 2004) y se analizarán las diferencias respecto de las metodologías tradicionales. Por último, se desarrollarán las conclusiones del trabajo y detallarán los pasos a seguir en las próximas etapas (Izaurrealde, 2013).

Bacca (2015); "Una metodología ágil para el desarrollo de software en una compañía financiera".

Conclusión: En este artículo se plantea una alternativa para mejorar el proceso de desarrollo de una compañía que actualmente no utiliza una metodología para la implementación de proyectos de software, lo que ha generado retrasos en las entregas, mala calidad de los productos terminados y sobrecarga laboral para algunos empleados. Las metodologías a plantear son metodologías de desarrollo existentes que ya han sido comprobadas como exitosas para la ejecución de este tipo de proyectos y que permiten dar cumplimiento a los requerimientos del cliente tanto en tiempo como en calidad aprovechando al máximo los recursos de la organización. En el desarrollo de software existen muchas falencias, una de ellas y la más importante es la calidad con la que se entrega el producto al cliente y esto a causa de la dificultad para encontrar fallas en el desarrollo de los requerimientos (Merchán , Urrea, & Rebollar, 2008) y una de las soluciones a esto son las metodologías de desarrollo ágil ya que estas hacen énfasis en la entrega de valor al cliente y en la generación de negocio (Díaz, 2009) La compañía a evaluar es una empresa que provee tecnología a entidades financieras, tiene varios proyectos de desarrollo de software, los cuales forman parte de la gerencia de soluciones informáticas, estos proyectos hacen parte de diferentes áreas que se dividen dependiendo del producto que se le ofrece a las entidades financieras, cada área está conformada por varios equipos que se encargan de los diferentes proyectos que están asignados, estos equipos tiene unos requerimientos o necesidades que provienen de un cliente externo (entidad financiera de un grupo financiero) u otra área; al momento de recibir estos requerimientos, se entregan al equipo y cada líder realiza una planificación (en la algunos de casos haciendo una reunión con su equipo) de tiempos de las diferentes etapas, el desarrollador se encarga de organizar su tiempo de acuerdo a los requerimientos que se le sean asignados, pero muchas

veces por el afán de las entregas y las emergencias que se presentan por el camino, los tiempos tienden a no cumplirse, esto puede generar una entrega fuera del tiempo estipulado, o a tiempo pero en un estado de calidad no esperado por el cliente, además de generar una sobrecarga laboral para algunos empleados. Analizando esta situación, se propone utilizar metodologías de desarrollo ágiles, ya que una de las ventajas de esta estrategia está sobre el producto final el cual se podrá entregar en el tiempo y con la calidad esperada, utilizando los recursos de manera eficiente y generando mayor productividad, a su vez, se disminuye el riesgo de futuros defectos o errores que pueden aparecer por la falta de tiempo para realizar las pruebas correspondientes.

**Tabla 1.** Principales empresas que siguen una Metodología Ágil

Nº	Sectores	Usuario
1	Media y Telcos	BBC, BellSouth, British Telecom, DoubleYou, Motorola, Nokia, Palm, Qualcomm, Schibsted, Sony/Ericsson, Telefonica I+D, TeleAtlas, Verizon
2	Software, Hardware	Adobe, Autentia, Biko2, Central Desktop, Citrix, Gailén, IBM, Intel, Microfocus, Microsoft, Novell, OpenView Labs, Plain Concepts, Primavera, Proyectalis, Softhouse, Valtech, VersionOne
3	Internet	Amazon, Google, mySpace, Yahoo
4	ERP	SAP
5	Banca e Inversión	Bank of America, Barclays Global Investors, Key Bank, Merrill Lynch
6	Sanidad y Salud	Patientkeeper, Philips Medical
7	Defensa y Aeroespacial	Boeing, General Dynamics, Lockheed Martin
8	Juegos	Blizzard, High Moon Studios, Crytek, Ubisoft, Electronic Arts
9	Otros	3M, Bose, GE, UOC, Ferrari

Nota. Fuente: (Samamé Silva, 2013)

## 2.2. Bases Teóricas de las Variables

### 2.2.1. Métodos Ágiles

(Sánchez, 2013) Menciona que, “Las metodologías Ágiles han demostrado en los últimos años mayor eficacia en el control del avance y manejo de expectativas, sin embargo, debe aceptarse que, aunque no son la solución a todos los problemas, si constituye una buena alternativa para el desarrollo de proyectos informáticos”.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

(Bahit Eugenia, 2011); Expresa, “el enfoque ágil, plantea la definición de un alcance global al comienzo, para luego ir incrementándolo en las diversas iteraciones, cada una de las cuales, supone la entrega de un software 100% funcional.”.

(Schwaber, 2013); Las metodologías ágiles son aquellas que impulsan generalmente una gestión de proyectos que promueve el trabajo en equipo, la organización y responsabilidad propia, un grupo de buenas prácticas de ingeniería de software que brindan una entrega rápida de software de alta calidad, y un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía.

(Carvajal, 2008), Menciona los principales valores del desarrollo Ágil al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas, La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también pueden afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el

equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

Desarrollar software que funciona más que conseguir una buena documentación, Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.

La colaboración con el cliente más que la negociación de un contrato, Las características particulares del desarrollo de software hacen que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Responder a los cambios más que seguir estrictamente un plan, La habilidad de responder a los cambios que puedan surgir al largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

### **Valores y Principios del Manifiesto Ágil**

(Deemer, 2012), Menciona que el agilísimo se basa en 4 valores y 12 principios, que se desprenden del Manifiesto Ágil.

#### **Valores**

1. Valora Individuos e interacciones del equipo sobre procesos y herramientas que se tendrán en cuenta en el proceso.
2. Software funcionando sobre documentación extensiva.

3. Exista una interacción constante entre el cliente y el equipo de desarrollo, esta colaboración asegura el éxito del proyecto.
4. Respuesta ante el cambio antes que seguir estrictamente un plan que conlleve a la monotonía empresarial.

### **Principios**

1. Nuestra mayor prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
2. Aceptamos requisitos cambiantes, incluso en etapas avanzadas dentro del proyecto.
3. Entregamos software frecuentemente.
4. Los responsables de negocio y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
5. Construimos proyectos con profesionales motivados.
6. Conversación cara a cara.
7. Software que funciona es la principal medida de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible.
9. La atención continua a la excelencia técnica y los buenos diseños mejoran la agilidad.
10. Simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de equipos que se auto-organizan.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo.

(Franco, 2012); Expresa, “Aunque la iniciativa nació del desarrollo de software, bien puede ser aplicada a cualquier industria. El objetivo primordial de las metodologías ágiles es justamente entregar soluciones rápida y continuamente, con un equipo multidisciplinario, auto organizado y con la colaboración constante del cliente. Recordemos que la prioridad de Agile es la entrega constante de valor.”.

### **Principales Referencias de la Gestión Ágil de Proyectos**

(Guerrero Cando, Renán Mauricio; María Fernanda, 2015) Los métodos ágiles están ganando popularidad en el sector de desarrollo de software. Varias empresas están reportando grandes ganancias de productividad y la satisfacción

del incremento del negocio con los sistemas desarrollados utilizando estos enfoques.

Un estudio previo en la cual se aplicó gestión comercial y metodología ágil se tiene: “Desarrollo de un sistema web de comercio electrónico B2C, para la promoción compra on-line y gestión de stock de artículos de cuero”, en la cual se concluye que la definición de requerimientos es una tarea importante, pues de ellos son los que depende la satisfacción de los usuarios del sistema. Además, se debe conocer lo que el público objetivo espera del sistema, con lo cual el diseño final debe ser un sistema intuitivo para que sea de fácil acceso y esto pueda traducirse en una compra y publicidad asegurada. Además, Scrum es una metodología que se acomoda de la mejor manera al equipo de desarrollo, ya que deja de lado ciertos formalismos el cual permite que realizar las tareas de cada miembro del equipo sea en base a su perfil. Y que las metodologías de desarrollo ágil se caracterizan por su flexibilidad y rápida respuesta al cambio, por lo que la flexibilidad implica alteraciones, que a veces es mínima, en la planificación de tareas y el orden en que se ejecutan. Es por tal motivo que la documentación no debe ser excesiva, sino puntual ya que estas pueden variar durante el tiempo de desarrollo.

### **Programación Extrema (Extreme Programming, XP)**

(Larman, 2014), Indica que, “es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.”.

### **Roles XP**

**Cliente**, Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y

decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

**Programador**, Escribe las pruebas unitarias y produce el código del sistema una vez empezado el proyecto.

**Encargado de pruebas (Tester)**, Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

**Encargado de seguimiento (Tracker)**, Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

**Entrenador (Coach)**, Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

**Consultor**, Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

**Gestor (Big boss)**, Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

## **Proceso XP**

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

## **El Ciclo de Vida Ideal**

1. Exploración
2. Planificación de la Entrega

3. (Release), Iteraciones
4. Producción
5. Mantenimiento
6. Cierre del Proyecto.

### **Prácticas XP**

Para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas

**El juego de la planificación,** Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

**Entregas pequeñas,** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

**Metáfora,** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

**Diseño simple,** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

**Pruebas,** La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.

**Refactorización (Refactoring),** Es una actividad constante de reestructuración del código para hacerlo más flexible y facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.

**Programación en parejas,** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores.).

**Propiedad colectiva del código,** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

**Integración continua,** Cada pieza de código es integrada en el sistema una vez que esté lista.

**40 horas por semana,** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.

**Cliente in-situ,** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

**Estándares de programación,** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. La mayoría de las prácticas propuestas por XP no son novedosas, sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

## **Scrum**

(Benefield, 2014), Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos

El desarrollo de software se realiza mediante iteraciones, denominadas *Sprint*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.

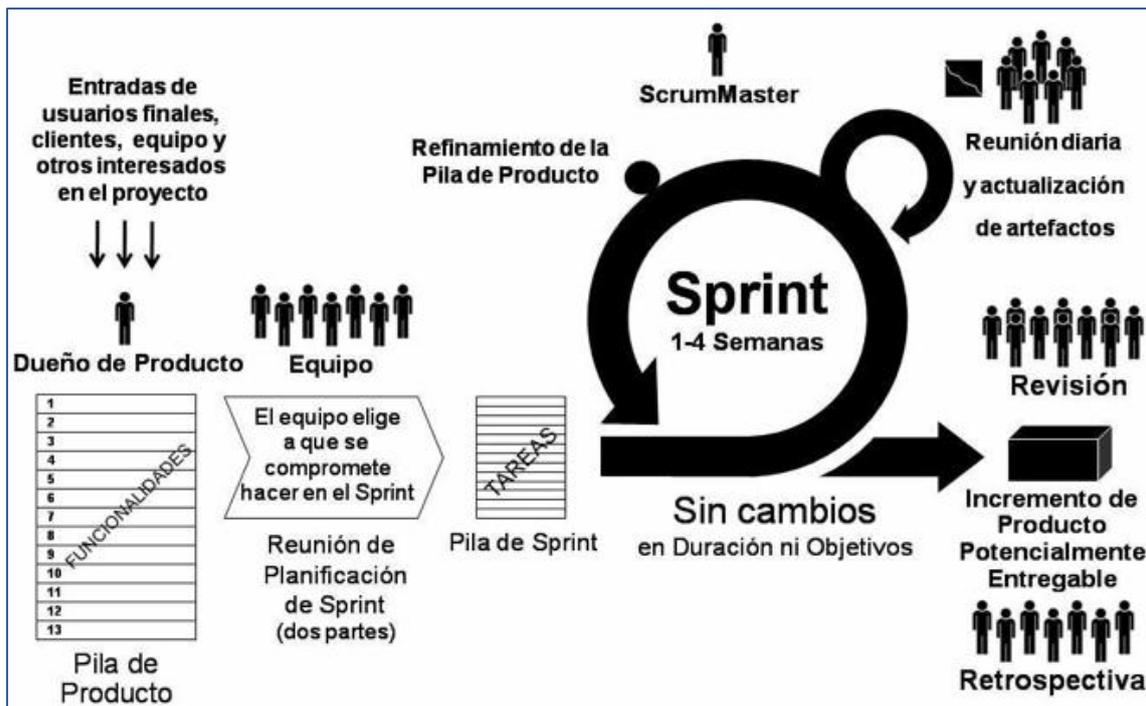
La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración que se debe realizar.

Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprint. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra.

Los Sprint son de duración fija terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint.

Durante el Sprint no se pueden cambiar los elementos elegidos. Todos los días el equipo se reúne brevemente (aprox. 15 minutos) para informar del progreso, y actualizan unas gráficas sencillas que les orientan sobre el trabajo restante. Al final del Sprint, el equipo revisa el Sprint con los interesados en el proyecto, y les enseña lo que han construido

La gente obtiene comentarios y observaciones que se puede incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar.



**Figura 2. Scrum: Procesos del Scrum**

Fuente: Deemer P., Benefield G., Larman C., & Vodde B., "The Scrum Primer", 2009

## Roles

### Product Owner- Dueño de Producto (DP)

Es el responsable de maximizar el retorno de inversión ROI, identificando las funcionalidades del producto, poniéndolas en una lista priorizada de funcionalidades, decidiendo cuales deberían ir al principio de la lista para el siguiente Sprint, y re priorizando y refinando continuamente la lista.

El DP tiene la responsabilidad de las pérdidas y ganancias del producto, asumiendo que es un producto comercial. En el caso de una aplicación interna, el DP no es responsable del ROI en el mismo sentido de un producto comercial (que dará beneficio), pero es responsable de maximizar el ROI en el sentido de elegir en cada Sprint los elementos de más valor de negocio y menos coste.

En algunas ocasiones el DP y el cliente son la misma persona; esto es muy común en aplicaciones internas. En otras, el cliente podría ser millones de personas con diferentes necesidades, en cuyo caso el rol de DP es parecido al rol de jefe de producto o jefe de marketing del producto que hay en muchas empresas. Sin embargo, el Dueño de Producto es diferente al tradicional jefe de producto porque interactúa activa y frecuentemente con el equipo, estableciendo

personalmente las prioridades y revisando el resultado en cada iteración de 1 a 4 semanas, en vez de delegar las decisiones de desarrollo en el jefe de proyecto. Es importante destacar que en Scrum hay una persona y sólo una, que hace y tiene la autoridad final de Dueño de Producto.

## **El Equipo**

Construye el producto que va a usar el cliente, por ejemplo, una aplicación o un sitio web. El equipo en Scrum es “multi-funcional” tiene todas las competencias y habilidades necesarias para entregar un producto potencialmente distribuible en cada Sprint y es “auto organizado” (auto gestionado), con un alto grado de autonomía y responsabilidad. En Scrum, los equipos se auto-organizan en vez de ser dirigidos por un jefe de equipo o jefe de proyecto.

El equipo en Scrum consta de siete personas más menos dos, y para un producto de software el equipo podría incluir analistas, desarrolladores, diseñadores de interface, y testers. En Scrum, el equipo debería estar dedicado al 100% al trabajo en el producto durante el Sprint; intentando evitar hacer varias tareas en diferentes productos o proyectos. A los equipos estables se les asocia con una productividad más alta, así que evita cambiar miembros del equipo.

A los grupos de desarrollo de aplicaciones con mucha gente se les organiza en varios equipos Scrum, cada uno centrado en diferentes funcionalidades del producto, coordinando sus esfuerzos muy de cerca. Dado que el equipo hace todo el trabajo (planificación, análisis, programación y pruebas) para una funcionalidad completa centrada en el cliente, a los equipos de Scrum también se les llama equipos por funcionalidades.

## **ScrumMaster**

Ayuda al grupo del producto a aprender y aplicar Scrum para conseguir valor de negocio. El ScrumMaster hace lo que sea necesario para ayudar a que el equipo tenga éxito. El ScrumMaster no es el jefe del equipo o jefe de proyecto; el ScrumMaster sirve al equipo, le protege de interferencias del exterior, y enseña y guía al DP y al equipo en el uso fructífero de Scrum.

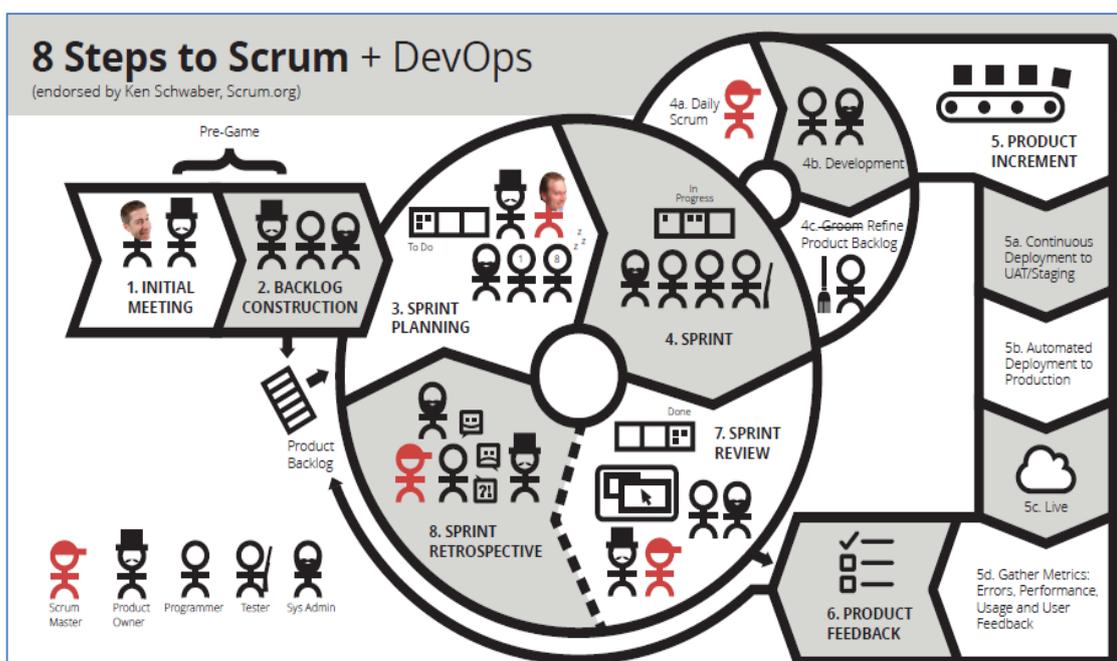
(Carvajal Riola, 2010). El ScrumMaster se asegura de que todo el mundo en el equipo (incluyendo al DP y la gerencia) entienda y siga las prácticas de

Scrum. Los equipos de Scrum deberían tener un ScrumMaster a tiempo completo, aunque en un equipo más pequeño podría ser un miembro del equipo (llevando una carga de trabajo más ligera). Un gran ScrumMaster puede venir de cualquier experiencia o disciplina previa: ingeniería, diseño, testing, gestión de productos, gestión de proyectos o gestión de calidad.

El ScrumMaster y el Dueño de Producto no pueden ser la misma persona; a veces el ScrumMaster necesitará parar al DP cuando si intenta meter nuevas funcionalidades en mitad de un Sprint. Y al contrario de un jefe de proyecto, el ScrumMaster no le dice a gente las tareas que tienen asignadas lo que hace es facilitar el proceso, apoyando al equipo que se organiza y gestiona solo. Se debe tomar en cuenta que no existe el rol de jefe de proyecto en Scrum.

- Ayudan a quitar los impedimentos identificados por el equipo.
- Ponen su experiencia y conocimiento a disposición del equipo.

En Scrum, los jefes cambian el tiempo que dedicaban a hacer de “niñeras” (asignar tareas, pedir informes de estado y otras formas de micro gestión) por tiempo como “gurús” o “sirvientes” del equipo (mentoring, coaching, ayudar a quitar obstáculos, ayudar a resolver problemas, dar ideas creativas y guiar el desarrollo de habilidades de los miembros del equipo.



**Figura 3. “8 Steps to Scrum”**

Fuente: (Schwaber, 2012)

## **Convivencia Entre las Metodologías Tradicionales y Ágiles**

(Norberto, 2012). Existen proyectos en los cuales los objetivos del negocio, los requerimientos, y la solución a implementarse están más o menos claros de entrada por lo que se podría planificar todo al principio y sin mayores inconvenientes (lo que Wysocki denomina planificación predictiva o tradicional). En los otros extremos cuando no existe tanta claridad debería utilizarse una aproximación adaptativa o ágil y cuando hay mucha incertidumbre se recurre a la programación extrema.

Para algunos las metodologías ágiles vienen a triunfar donde la metodología “tradicional” había fracasado. Para otros, las metodologías ágiles son sólo una excusa para desarrollar software sin planes, diseño ni documentación. Ambos extremos son erróneos; por una parte, varias prácticas ágiles tienen nichos donde funcionan muy bien, pero hay otros ambientes donde pierden su efectividad o deben ser modificadas para lograr el objetivo.

Las prácticas ágiles y las más tradicionales han demostrado tener nichos en los cuales se comportan bien y favorecen el éxito. Lo que sucede ahora con mayor frecuencia en el vertiginoso mundo en que vivimos, es que estos nichos se entrecruzan frecuentemente, y por lo tanto ningún esquema resulta completo o favorable por sí sólo. No hay que quedarse en ningún extremo. Eso puede llevarnos a ignorar los objetivos del proyecto y conducirnos ciegamente al fracaso.

Por el contrario, reconociendo el valor que una mezcla adecuada puede generar, aumentaremos notablemente la probabilidad de éxito de nuestros proyectos no sólo en los parámetros usuales de plazo, alcance y costo, sino también en el desarrollo del conocimiento y experiencia para el equipo y la organización.

Se observa las características de las Metodologías Ágiles respecto de las Metodologías Tradicionales que no sólo se refieren al proceso en sí, sino también al contexto de equipo y organización en los procesos de desarrollo de software.

METODOLOGÍA ÁGIL	METODOLOGÍA TRADICIONAL
Pocos artefactos	Más artefactos
El modelo es prescindible, modelos desechables	El modelo es esencial, mantenimiento de modelos.
Pocos roles, genéricos y flexibles	Más roles, más específicos
No existe contrato tradicional, debe ser flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (in situ)	Cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños, corta duración (entregables frecuentes)	Aplicables a proyectos de cualquier tamaño especialmente grandes
Equipos pequeños hasta 10 integrantes trabajando en el mismo sitio	Equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Énfasis en los aspectos humanos, el individuo y el trabajo en equipo	Énfasis en la definición del proceso, roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguido por el entorno de desarrollo
Se espera cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

**Figura 4. Características de las Metodologías Ágiles y Tradicionales**

Fuente:(Palacios, 2014)

ELABORACIÓN PROGRESIVA (PMI)			DESARROLLO ÁGIL		
FASE	ACTIVIDAD	HERRAMIENTA	FASE	ACTIVIDAD	HERRAMIENTA
INICIO	Definición del Alcance	Definición del alcance	Product Backlog Review	Revisión del alcance del producto y descomposición en sus partes	Backlog con la características del producto desagregadas
	Elaboración de Estructura Desagregada de Trabajo (EDT)	Estructura Desagregada de Trabajo (EDT)			
PLANIFICACIÓN	Definición de actividades	Lista de actividades	Planificación de la Iteración	Definición de las historias de usuario	Backlog producto
	Secuenciamiento de actividades	Diagrama de precedencia		Priorización de las historias de usuario	Backlog producto
	Estimación de duración de actividades	Estimaciones de duración de actividades		Estimación del esfuerzo para fabricar componentes	Plan de la Iteración
EJECUCIÓN	Dirección y Ejecución del trabajo	Sistema de asignación de tareas	Ejecución de la Iteración	Ejecución de la Iteración	El equipo usa el radiador para seleccionar el trabajo a realizar
MONITOREO Y CONTROL	Control de Calidad	Inspecciones	Ejecución de la Iteración	Ejecución de la Iteración	QA trabaja de la mano con personal de desarrollo construyendo la calidad del producto.
CIERRE	Cierre de fase	Lecciones aprendidas	Retrospectiva	Reunión del Equipo	Lecciones aprendidas
	Autorización de cierre de fase	Aprobación	Cierre iteración	Revisión del Backlog	Backlog de producto

Fuente: (Palacios, 2014)

**Figura 5. Actividades PMI y Herramientas de Desarrollo Ágil**

Muchos de los artefactos PMI serán utilizados sólo al principio de proyectos ágiles, debido a lo siguiente:

- Los costos y tiempos son fijos, definidos al principio del proyecto, por lo que el foco de cada iteración es definir la duración de las actividades y recursos necesarios para completar el alcance y seleccionar una cantidad tal que se ajuste a los tiempos y costos de la iteración.
- El Proceso de Control de Cambios está integrado a la metodología ágil, pues al final de cada iteración, se agregan, descartan, desagregan o reordenan las prioridades de los elementos del Back log de producto, definiendo y aprobando el alcance de próximas iteraciones.
- Actividades PMI de Identificación de implicados, definición de objetivos y acta de constitución, bajo un enfoque ágil deberían realizarse previo al comienzo de las iteraciones o en una iteración cero.

### **Procesos en el Proyecto de Software**

(Kniberg, 2007). Es importante definir en los proyectos de software que procesos, áreas, interacciones, entregables, entre otros tanto del PMI como de las metodologías ágiles se pueden fusionar en la gestión, ya que estos determinarán el éxito del proyecto.

### **Procesos de Inicialización**

Permite identificar las características iniciales preliminares del alcance del proyecto (tradicional) y adicional identificar los métodos, procedimientos que se utilizarán para entregar el proyecto (ágil), los conceptos de desarrollo iterativo y la producción de los incrementos regulares de software debe explicarse claramente a todos los interesados, debido a que puede representar una forma diferente de trabajar para algunas personas.

### **Procesos de Planificación**

Se aplica cuándo existen modificaciones continuas, la planificación es crítica en proyectos tradicionales y ágiles. Sin embargo, no debe concentrarse mucho tiempo debido a que al principio es cuándo menos se conoce del proyecto. Las necesidades de planificación deben ser iterativas y continuas a lo largo del proyecto, guiado por retroalimentación real de desempeño del proyecto, los

cambios del negocio y el crecimiento de los conocimientos técnicos. La elaboración progresiva habla de la necesidad de perfeccionar y desarrollar planes de alcance a medida que más información esté disponible. El concepto relacionado con la planificación de la onda del balanceo que establece la planificación debe ser iterativo y continuo basado en horizontes de tiempo cortos.

**Empoderamiento del equipo:** Lo tradicional es la asignación de trabajo de arriba hacia abajo, esto es problemático para el software, debido a que por lo general existen dependencias impredecibles y asignar a los desarrolladores tareas individuales ocasionan que sean propensos a fallar, en su lugar es mejor aprovechar la capacidad de las personas para gestionar complejidad, relacionados con objetivos de alto nivel asignando la responsabilidad al equipo. De esta manera el equipo podrá seleccionar y desarrollar las características de esa iteración como mejor crean conveniente.

### **Procesos de Control**

Usar técnicas ágiles, en proyectos de software es aconsejable debido a que sufren muchas desviaciones, además de variar el plan del proyecto, por lo cual se requiere intervención. El Control del proyecto y Control de cambios se realiza progresivamente en cada iteración.

### **Procesos de Cierre**

Los procesos de cierre definidos proporcionan buenas pautas generales para actividades a tener en cuenta al cierre de los proyectos de desarrollo de software como lineamientos o requisitos de cierre del proyecto (por ejemplo, auditorías finales del proyecto, evaluaciones del proyecto, validaciones del producto y criterios de aceptación).

### **Combinando Scrum y Extreme Programming**

Es necesario compatibilizar ambas metodologías, es con un buen análisis comparativo de cada propuesta se logra obtener, el grado de compatibilidad entre ambas. Haremos se mostrará cada uno de los valores y prácticas técnicas de Extreme Programming, analizándolos de forma separada a fin de deducir cuán compatibles resultan con el Scrum.

## Compatibilidad Scrum – XP

La siguiente tabla muestra una comparación entre los valores de XP y su correlación con Scrum, obteniendo el grado de compatibilidad en una escala de 1 a 3, siendo 1, poco compatible y requerirá decidir si se debe implementar o no; 2 algo compatible, pero debe adaptarse y 3, absolutamente compatible.

**Tabla 2. Compatibilidad SCRUM - XP**

VALOR	XP	SCRUM	COMPATIBILIDAD
<b>Comunicación</b>	Los impedimentos son resueltos entre los miembros del equipo y cliente	Los impedimentos son relevados por el equipo al Scrum Master y éste se encarga resolverlos (incluyendo o no, al dueño de producto)	1 / 3
<b>Simplicidad</b>	Pretende desarrollar aquellos aspectos mínimos necesarios para cumplir con los objetivos.	Se priorizan las Historias de Usuario, desarrollando solo aquellas más prioritarias.	3 / 3
<b>Retroalimentación</b>	Pretende lograr un feedback continuo con el cliente, a través de las entregas tempranas.	Se planifica y revisa en forma conjunta entre el equipo y dueño de producto (cliente) en períodos regulares. Si surgen impedimentos, son relevados al Scrum Master y éste se encarga de resolverlos.	2 / 3
<b>Respeto</b>	Se respeta la idoneidad del cliente con respecto a las decisiones que aportan valor al negocio y la del equipo con respecto al cómo esas decisiones deben ser desarrolladas.	Solo el dueño de producto (cliente) puede decidir que Historias de Usuario serán desarrolladas en el Sprint y únicamente el equipo puede definir la forma en la cual serán desarrolladas.	3 / 3
<b>Coraje</b>	El equipo se compromete a estimar con sinceridad y comentar con honestidad el avance del proyecto.	Las estimaciones se realizan frente al dueño de producto y el equipo se compromete a relevar los impedimentos al Scrum Master y mostrar el avance del proyecto en el tablero.	3 / 3

**Nota.** Fuente: Elaboración Propia del Autor.

Como se puede apreciar en la tabla anterior, encontramos que la comunicación, es el valor que menor compatibilidad posee. Esto significa, al momento de unificar metodologías, se deberá decidir la forma en la cual se va a realizar la comunicación, para esta tesis se tomará como base la siguiente metodología.

**Estilo XP** → serán resueltos entre los miembros del equipo.

Cabe resaltar que esta retroalimentación solo será para el aspecto de la comunicación ya que para los demás valores se tomará en cuenta ambas Metodologías por igual.

### **Compatibilidad Prácticas Técnicas Scrum – XP**

Para este aspecto técnico el análisis exhaustivo de cada práctica no será fundamental, ya que Scrum, no propone ninguna de forma directa puesto que este es un Framework muy adaptable. Solo nos concentraremos en aquellas prácticas que puedan llegar a generar ciertas dudas. Utilizaremos el mismo método de análisis de compatibilidad, empleado en la tabla anterior.

**Tabla 3.** Compatibilidad Prácticas Técnicas SCRUM - XP

<b>VALOR</b>	<b>XP</b>	<b>SCRUM</b>	<b>COMPATIBILIDAD</b>
<b>Cliente in-situ</b>	Lo integra al desarrollo de forma directa (el cliente está físicamente presente durante todo el proceso de desarrollo)	Lo integra de forma indirecta, en las ceremonias de planificación y revisión y opcionalmente, en las de revisión y retrospectiva.	1 / 3
<b>Entregas cortas</b>	Cada 1 o 2 semanas	Cada 2 o 4 semanas	2 / 3
<b>Testing</b>	El cliente define los test de aceptación	El dueño de producto define los criterios de aceptación	2 / 3

<b>Juego de Planificación</b>	No propone una técnica de estimación de forma explícita.	Indefectiblemente propone el uso de al menos una técnica de estimación (Planning Pocker, TShirt Sizing, Columns o una combinación de varias)	3 / 3
-------------------------------	--	--	-------

**Nota.** Fuente: Elaboración Propia del Autor.

Como podemos apreciar en la tabla anterior, son 3 las prácticas técnicas que deberán ser adaptadas. En el caso de cliente in-situ, se deberá adoptar esta práctica, con un modelo fijado y establecido, para esta tesis se tomará como base la siguiente metodología

**Estilo Scrum** → Contar con el cliente solo en las ceremonias de planificación y revisión del avance del Proyecto a través de los Sprint, esto está completamente dentro de los parámetros del Scrum y tiene parte del estilo XP.

Cabe resaltar que esta retroalimentación solo será para el aspecto técnico del Cliente In-Situ. Con respecto a las entregas cortas, son perfectamente compatibles entre ambas metodologías. Establecer Sprint de 2 semanas, sería el equilibrio ideal para los clientes y programadores encargados del proyecto.

En cuanto al Testing, la diferencia encontrada, es que en XP, el cliente define los Test de Aceptación y en Scrum, solo indica los criterios de aceptación (Requerimientos a Cumplir). La forma de compatibilizar ambas metodologías, es simplemente, sumar al cliente (dueño de producto), la facultad de definir Test de Aceptación para cada uno de los Criterios de Aceptación definidos en una Historia de Usuario (Anexos).

### 2.2.2. Dirección de Proyectos de Software

(Maigua, López, 2012) La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo.

Evidentemente, supone gozar de una visibilidad más amplia sobre los recursos y los objetivos, y difícilmente estaremos dispuestos a renunciar a la gestión de nuestros propios recursos, aunque, en muchos casos, implique cierta responsabilidad adicional y, por qué no decirlo, algún que otro dolor de cabeza.

Asumido lo anterior, ¿por qué los jóvenes profesionales son tan reacios a dirigir y a gestionar los proyectos en los que participan? Es evidente que, al optar por una carrera profesional no vinculada con la gestión, el individuo se decanta por actividades técnicas, pero eso no ha de significar renunciar a la perspectiva que supone participar en las tareas de dirección y gestión de los proyectos en los que trabaje.

Participar en la gestión y dirección de un proyecto supone conocer los recursos y objetivos del mismo, así como las limitaciones a tener en cuenta, y este conocimiento nos permite optimizar nuestro trabajo y adaptarlo a los requisitos más específicos y más relevantes del proyecto. La dirección pasa de ser una tarea aislada a constituirse en una herramienta para mejorar el desarrollo de las actividades técnicas.

### **Proyecto Informático**

Un Proyecto Informático es un sistema de cursos de acciones simultáneas y/o secuenciales que incluye personas, equipamientos de hardware, software y comunicaciones, enfocadas en obtener uno o más resultados deseables sobre un sistema de información.

Los recursos más frecuentemente utilizados que caracterizan a un sistema de información, son los componentes de la Tecnología de la Información (TI) como puede ser el uso de Hardware, Software y Comunicaciones. Considerando entonces, la importancia que la informática tiene en los planes estratégicos de cualquier empresa moderna, no solamente se debe tener en cuenta la evolución de los recursos de la tecnología de la información, sino también las distintas metodologías para el desarrollo de los sistemas de información.

### **Director del Proyecto**

(Ríos, 2016) El director del proyecto es la persona asignada por la organización ejecutora para liderar al equipo responsable de alcanzar los objetivos del proyecto. El rol del director del proyecto es diferente del de un

gerente funcional o del de un gerente de operaciones. Por lo general, el gerente funcional se dedica a la supervisión gerencial de una unidad funcional o de negocio y la responsabilidad de los gerentes de operaciones consiste en asegurar que las operaciones de negocio se llevan a cabo de manera eficiente.

Dependiendo de la estructura de la organización, un director de proyecto puede estar bajo la supervisión un gerente funcional. En otros casos, el director del proyecto puede formar parte de un grupo de varios directores de proyecto que dependen de un director de programa o del portafolio, que es el responsable en última instancia de los proyectos de toda la empresa. En este tipo de estructura, el director del proyecto trabaja estrechamente con el director del programa o del portafolio para cumplir con los objetivos del proyecto y para asegurar que el plan para la dirección del proyecto esté alineado con el plan global del programa. El director del proyecto también trabaja estrechamente y en colaboración con otros roles, como los de analista de negocio, director de aseguramiento de la calidad y expertos en materias específicas.

Sin embargo, el trabajo del proyecto de software puede ser organizado por el componente del producto (por ejemplo, la interfaz de usuario, base de datos, computación y software de comunicación), funcionalmente por componente de proceso (por ejemplo, el análisis, el diseño, la construcción, los procesos de pruebas, y la instalación / formación), o por subsistema (por ejemplo, tiempo, radar, pantalla del tráfico aéreo). El equipo de proyecto podrá realizarse de una manera funcional, por ejemplo, el director del proyecto de software puede ser el gerente de las unidades funcionales del proyecto durante la planificación y ejecución de ese proyecto. Además, un proyecto de software grande puede ser tratado como un programa de software que se descompone en múltiples proyectos; cada una con un jefe de proyecto cuyos productos trabajo se fusionan un solo producto.

### **Características**

Dado que el software es un producto intangible, hay muchas posibilidades para los ciclos de vida de proyectos de software y muchos enfoques diferentes para la aplicación de las técnicas de dirección de proyectos de software a los ciclos de vida.

(Ríos, 2016) El principio de "la mayoría de los proyectos, la mayor parte del tiempo" es modificarse para adaptarse a las variaciones dentro de los continuos ciclos de vida de desarrollo de software; los directores de proyectos de software se adaptan los métodos, herramientas y técnicas para la gestión de proyectos y a los aspectos particulares de diversos ciclos de vida del proyecto de software.

No se espera que los directores de proyectos de software tengan el conocimiento y las habilidades de sus miembros del equipo en profundidad, pero deben entender los problemas y preocupaciones de sus miembros del equipo, tratar y deben de estar familiarizados con la terminología utilizada por los miembros del equipo. Los directores de proyectos de software y también deben entender diferentes enfoques para la dirección de proyectos de software dentro de la continuidad de los ciclos de vida de los proyectos de software.

Los directores de proyectos de software pueden tener antecedentes técnicos, pero no siempre en el software. Esos directores de proyectos que no tienen fuertes habilidades de software pueden tener que trabajar en estrecha colaboración con un líder técnico en su proyecto de software. Las personas con conocimientos de software fuertes podrían tener que concentrarse en el desarrollo de sus negocios, gestión de proyectos y habilidades interpersonales.

## **2.3. Definición de Términos Básicos**

### **2.3.1. Dirección de Proyectos**

(PMBOK, 2017) Es la aplicación del conocimiento, de las habilidades, y de las técnicas para ejecutar los proyectos en forma eficiente y efectiva. Es una competencia estratégica para las organizaciones, y les permite atar los resultados de los proyectos a las metas del negocio, y así competir mejor en su mercado.

### **2.3.2. Dirección de Proyectos de Software**

(Carrasco, 2017) Es más que la capacidad de reconocer los desafíos que te proporciona el cliente o la Empresa, para a través de ellos encontrar, revisar y evaluar las múltiples soluciones, seleccionando la que más responda a las

definiciones de eficiencia y calidad, para después ponerla en práctica, acorde a los objetivos y planificación establecidos.

La dirección de proyectos simplemente en conducir un proyecto desde el comienzo hasta un final satisfactorio, haciendo uso conjunto de procesos, conocimientos, habilidades, herramientas y técnicas que orienten y motiven al personal a realizar satisfactoriamente su trabajo dentro del proyecto.

### **2.3.3. Scrum**

(Eugenia, 2012), Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

### **2.3.4. Sprint**

(Nateevo, 2012), Es un subconjunto de requerimientos, extraídas del product backlog, para ser ejecutadas durante un periodo entre 1 y 4 semanas de trabajo. El sprint backlog sería el documento que describa las tareas que son necesario realizar para abordar los dichos subconjuntos de los requerimientos.

### **2.3.5. Extremme Programming (XP)**

(Luna, 2012), Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

### **2.3.6. Entrenador (Coach)**

(Palacios, 2014), Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

### **2.3.7. Encargado Seguimiento(Tracker)**

(Palacios, 2014), Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

### **2.3.8. Encargado Pruebas (Tester)**

(Palacios, 2014), Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

### **III. MÉTODOS Y MATERIALES**

#### **3.1. Hipótesis de la Investigación**

##### **Hipótesis General**

Las metodologías ágiles influirán en la optimización de la dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

##### **Hipótesis Específica**

Las metodologías ágiles influirán en la reducción de los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

Las metodologías ágiles influirán en la optimización del tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

#### **3.2. Variables de Estudio**

##### **3.2.1. Definición Conceptual**

##### **Variable Independiente (VI): Metodologías Ágiles**

(Bahit Eugenia, 2011); Expresa, “el enfoque ágil, plantea la definición de un alcance global al comienzo, para luego ir incrementándolo en las diversas iteraciones, cada una de las cuales, supone la entrega de un software 100% funcional.”.

(Schwaber, 2013); Las metodologías ágiles son aquellas que impulsan generalmente una gestión de proyectos que promueve el trabajo en equipo, la organización y responsabilidad propia, un grupo de buenas prácticas de ingeniería de software que brindan una entrega rápida de software de alta calidad, y un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía.

##### **Variable Dependiente (VD): Dirección de Proyectos de Software**

(Carrasco, 2017); Es más que la capacidad de reconocer los desafíos que te proporciona el cliente o la Empresa, para a través de ellos encontrar, revisar y evaluar las múltiples soluciones, seleccionando la que más responda a las definiciones de eficiencia y calidad, para después ponerla en práctica, acorde a los objetivos y planificación establecidos.

La dirección de proyectos simplemente en conducir un proyecto desde el comienzo hasta un final satisfactorio, haciendo uso conjunto de procesos, conocimientos, habilidades, herramientas y técnicas que orienten y motiven al personal a realizar satisfactoriamente su trabajo dentro del proyecto.

### 3.2.2. Definición Operacional

#### Variable Independiente (VI): Metodologías Ágiles

Las Metodologías ágiles tienen sus propios instrumentos y herramientas que permitirán controlar y medir el avance de cada una de estas dentro del Proyecto de Desarrollo de Software, como el diagrama de Burn-Down Chart aplicado por la metodología Scrum o el Refactoring aplicado por la metodología XP.

#### Variable Dependiente (VD): Dirección de Proyectos de Software

La Variable Dependiente se medirá a través de los indicadores de proyectos que serán el Costo y Tiempo que estos usan, ya que son los indicadores determinantes para una empresa al determinar si este fue exitoso o aún existen lecciones por aprender.

#### 3.2.2.1. Operacionalización de la Variable

**Tabla 4.** Matriz de operacionalización de la variable

VARIABLES	DIMENSIONES	INDICADORES	ESCALA DE MEDICIÓN	NIVELES Y RANGOS
VD: Dirección de Proyectos de Software	Costo	CPI (Índice de desempeño del costo).	% de CPI	0-1
	Tiempo	SPI(Índice del desempeño del programa)	% DE SPI	0-1

**Nota.** Fuente: Elaboración Propia del Autor.

### **3.3. Tipo y Nivel de Investigación**

El tipo de investigación es aplicada porque el estudio está en la resolución práctica de problemas. Se centra específicamente en cómo se pueden llevar a la práctica las teorías generales. La motivación va hacia la resolución de los problemas que se plantean en un momento dado.

(José Lozada, 2008) La investigación aplicada busca la generación de conocimiento con aplicación directa a los problemas de la sociedad o el sector productivo. Esta se basa fundamentalmente en los hallazgos tecnológicos de la investigación básica, ocupándose del proceso de enlace entre la teoría y el producto. El presente ensayo presenta una visión sobre los pasos a seguir en el desarrollo de investigación aplicada, la importancia de la colaboración entre la universidad y la industria en el proceso de transferencia de tecnología, así como los aspectos relacionados a la protección de la propiedad intelectual durante este proceso.

El nivel de la investigación es correlacional, permite establecer el grado de relación de la variable independiente sobre la variable dependiente.

Es importante tener una visión clara de la situación actual de la empresa Bits & Systems Solutions S.A.C con el fin de conocer las fortalezas y debilidades de su proceso de gestión de proyectos de desarrollo de software, este conocimiento permitirá proponer mejoras a sus prácticas internas.

### **3.4. Diseño de la Investigación**

El diseño de investigación será de tipo no experimental porque se usará una Primera Medición y Segunda Medición para demostrar que los usos de las Metodologías Ágiles optimizaran la dirección de proyectos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C.

(Hernández y Baptista, 2003) Son los estudios que se realizan sin la manipulación deliberada de variables y en los que se observan los fenómenos en su ambiente natural para después analizarlos.

### 3.5. Población y Muestra de Estudio

#### 3.5.1. Población

En la presente Tesis La empresa Bits & Systems Solutions, es una pequeña empresa de desarrollo de software a medida y estandarizado, por tal motivo el objeto de estudio estará conformado por 10 proyectos de desarrollo de Software que se realizan de forma simultánea.

#### 3.5.2. Muestra

Se elaboró una prueba de consulta al Gerente de la Empresa Bits & Systems Solutions, se le pregunto si conoce las metodologías ágiles y si estas se podrían aplicar a sus proyectos

En este caso se obtuvo la respuesta de que el 80% (P) de los proyectos podrían ser aplicados las metodologías ágiles y en el otro 20%(q) manifestó que no se podría trabajar con estas metodologías.

#### 3.5.3. Muestreo

Para el cálculo de la muestra se utilizó la fórmula de proporciones, es decir:

$$n_0 = \frac{PQ}{\left[\frac{d}{Z}\right]^2}$$

Dónde:

$n_0$ : Tamaño de la muestra.

Z:  $1 - \alpha = 99,9\%$

P: (Porcentaje del Problema).

Q: (Porcentaje del no Problema).

d:  $e * P$ .

E: Error relativo;  $e \in <0,01>$ .

Si consideramos como error relativo del 1% y el nivel de confianza  $1 - \alpha = 99,9\%$

$$n_0 = \frac{0.8 * 0.2}{\left[\frac{0.01 * 0.8}{0.5}\right]^2} = 625$$

Por regla entonces tenemos que la n buscada será:

$$n_0 = \frac{625}{1 + \left[\frac{625}{10}\right]} = 9.84 \cong 10$$

Por lo tanto, el tamaño de la muestra será: 10 Proyectos.

### **3.6. Técnicas e Instrumentos de Recolección de Datos**

#### **3.6.1. Técnicas de Recolección de Datos**

La principal técnica que se utilizara en la investigación es análisis documental.

(Castillo, 2005) Es una de las operaciones fundamentales de la cadena documental. Se trata de una operación de tratamiento. El análisis documental es un conjunto de operaciones encaminadas a representar un documento y su contenido bajo una forma diferente de su forma original, con la finalidad posibilitar su recuperación posterior e identificarlo.

El análisis documental es una operación intelectual que da lugar a un subproducto o documento secundario que actúa como intermediario o instrumento de búsqueda obligado entre el documento original y el usuario que solicita información. El calificativo de intelectual se debe a que el documentalista debe realizar un proceso de interpretación y análisis de la información de los documentos y luego sintetizarlo.

#### **3.6.2. Instrumentos de Recolección de Datos**

El instrumento de recolección de datos aplicado será de ficha de observación.

(Herrera, 2016) Son instrumentos de la investigación de campo. Se usan cuando el investigador debe registrar datos que aportan otras fuentes como son personas, grupos sociales o lugares donde se presenta la problemática.

Son el complemento del diario de campo, de la entrevista y son el primer acercamiento del investigador a su universo de trabajo.

Estos instrumentos son muy importantes, evitan olvidar datos, personas o situaciones, por ello el investigador debe tener siempre a la mano sus fichas para competir el registro anecdótico que realiza cuando su investigación requiere trabajar directamente con ambientes o realidades.

### 3.6.3. Validez del Instrumento

**Tabla 5.** Validez del Instrumento

<b>Expertos</b>	<b>Institución</b>	<b>Promedio de Valorización</b>
<b>Barrantes Ríos Edmundo José</b>	<b>UTELESUP</b>	<b>100 %</b>
<b>Ovalle Paulino Christian</b>	<b>UTELESUP</b>	<b>100 %</b>

**Nota.** Fuente: Elaboración Propia del Autor.

### 3.7. Métodos de Análisis de Datos

En la presente tesis se realizaron los procesamientos de los datos y sus síntesis mediante el programa Excel en el marco de la estadística descriptiva y la estadística inferencial para los análisis estadísticos, al realizar las estimaciones de las medidas de tendencia central para la comparación de las muestras del resultado de los instrumentos a fin de la verificación de las hipótesis planteadas en la investigación.

Se utilizó los siguientes estadígrafos:

- La estadística descriptiva: Promedio y desviación estándar.
- La estadística inferencial, para la prueba de hipótesis se utilizó la “t” de Student.
- Los análisis se realizaron con un nivel de significancia estadística del 95%.

## IV. RESULTADOS

### 4.1. Resultados del Procesamiento Estadístico del Instrumento

CPI (Índice de desempeño del costo).

*Tabla 6. Primera Medición CPI*

<b>PRIMERA MEDICIÓN</b>			
<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Costo Actual (AC)</b>	<b>CPI (EV / AC)</b>
ITE1-P1	500	680	0.7353
ITE2-P1	340	500	0.6800
ITE3-P1	400	590	0.6780
<b>Total CPI (P1)</b>			<b>2.0933</b>
ITE1-P2	200	350	0.5714
ITE2-P2	250	370	0.6757
ITE3-P2	250	390	0.6410
<b>Total CPI (P2)</b>			<b>1.8881</b>
ITE1-P3	335	460	0.7283
ITE2-P3	390	440	0.8864
ITE3-P3	325	470	0.6915
<b>Total CPI (P3)</b>			<b>2.3061</b>
ITE1-P4	227	365	0.6219
ITE2-P4	190	250	0.7600
ITE3-P4	196	222	0.8829
<b>Total CPI (P4)</b>			<b>2.2648</b>
ITE1-P5	413	490	0.8429
ITE2-P5	410	520	0.7885
ITE3-P5	436	501	0.8703
<b>Total CPI (P5)</b>			<b>2.5016</b>
ITE1-P6	520	560	0.9286
ITE2-P6	514	580	0.8862
ITE3-P6	538	599	0.8982
<b>Total CPI (P6)</b>			<b>2.7129</b>
ITE1-P7	120	190	0.6316
ITE2-P7	140	197	0.7107
ITE3-P7	115	230	0.5000
<b>Total CPI (P7)</b>			<b>1.8422</b>
ITE1-P8	327	356	0.9185
ITE2-P8	365	421	0.8670
ITE3-P8	369	495	0.7455

<b>Total CPI (P8)</b>			<b>2.5310</b>
ITE1-P9	284	369	0.7696
ITE2-P9	245	362	0.6768
ITE3-P9	230	400	0.5750
<b>Total CPI (P9)</b>			<b>2.0214</b>
ITE1-P10	428	560	0.7643
ITE2-P10	432	528	0.8182
ITE3-P10	472	640	0.7375
<b>Total CPI (P10)</b>			<b>2.3200</b>

**Nota.** Fuente: Elaboración Propia del Autor.

Como se aprecia en la Tabla Anterior se está tomando el total del CPI por cada proyecto de la Primera Medición, ya que los proyectos recogidos con el Muestreo fueron 10, estos datos usaremos para realizar la Tabla Final.

**Tabla 7. Segunda Medición CPI**

<b>SEGUNDA MEDICIÓN</b>			
<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Costo Actual (AC)</b>	<b>CPI (EV / AC)</b>
ITE4-P1	530	520	1.0192
ITE5-P1	463	456	1.0154
ITE6-P1	482	430	1.1209
<b>Total CPI (P1)</b>			<b>3.1555</b>
ITE4-P2	269	295	0.9119
ITE5-P2	250	240	1.0417
ITE6-P2	241	242	0.9959
<b>Total CPI (P2)</b>			<b>2.9494</b>
ITE4-P3	312	303	1.0297
ITE5-P3	356	340	1.0471
ITE6-P3	397	382	1.0393
<b>Total CPI (P3)</b>			<b>3.1160</b>
ITE4-P4	210	208	1.0096
ITE5-P4	140	143	0.9790
ITE6-P4	157	152	1.0329
<b>Total CPI (P4)</b>			<b>3.0215</b>
ITE4-P5	435	438	0.9932
ITE5-P5	427	423	1.0095
ITE6-P5	460	417	1.1031
<b>Total CPI (P5)</b>			<b>3.1057</b>
ITE4-P6	511	462	1.1061
ITE5-P6	563	520	1.0827

ITE6-P6	574	560	1.0250
Total CPI (P6)			3.2138
ITE4-P7	103	105	0.9810
ITE5-P7	174	167	1.0419
ITE6-P7	193	186	1.0376
Total CPI (P7)			3.0605
ITE4-P8	350	333	1.0511
ITE5-P8	392	389	1.0077
ITE6-P8	363	365	0.9945
Total CPI (P8)			3.0533
ITE4-P9	246	250	0.9840
ITE5-P9	273	279	0.9785
ITE6-P9	222	203	1.0936
Total CPI (P9)			3.0561
ITE4-P10	490	462	1.0606
ITE5-P10	483	497	0.9718
ITE6-P10	471	424	1.1108
Total CPI (P10)			3.1433

**Nota.** Fuente: Elaboración Propia del Autor.

Como se aprecia en la Tabla Anterior se está tomando el total del CPI por cada proyecto de la Segunda Medición, ya que los proyectos recogidos con el Muestreo fueron 10, estos datos usaremos para realizar la Tabla Final.

**Tabla 8.** Puntuación de requerimientos para la elección - CPI

PUNTUACIÓN DE REQUERIMIENTOS PARA LA ELECCIÓN				
MUESTRA	PRIMERA MEDICIÓN	TEST	SEGUNDA MEDICIÓN	D
1	2.0933		3.1555	-1.0623
2	1.8881		2.9494	-1.0613
3	2.3061		3.1160	-0.8099
4	2.2648		3.0215	-0.7567
5	2.5016		3.1057	-0.6041
6	2.7129		3.2138	-0.5008
7	1.8422		3.0605	-1.2183
8	2.5310		3.0533	-0.5223
9	2.0214		3.0561	-1.0346
10	2.3200		3.1433	-0.8233
<b>PROMEDIO</b>				-0.8394
<b>DESVIACIÓN</b>				0.2492
<b>T0</b>				-10.65

**Nota.** Fuente: Elaboración Propia del Autor.

En la tabla presentada se muestra el resumen estadístico para las dos muestras de datos del Índice de desempeño de costo. Para evaluar si las diferencias entre las estadísticas de las dos muestras son significativas. De particular interés es la curtosis estandarizada que pueden usarse para comparar si las muestras provienen de distribuciones normales. Valores de estos estadísticos fuera del rango de -1.96 a +1.96 indican desviaciones significativas de la normalidad, lo que tendería a invalidar las pruebas que comparan las desviaciones estándar. En este caso, los valores de ambas curtosis estandarizadas se encuentran dentro del rango esperado.

### SPI (Índice del desempeño del programa)

*Tabla 9. Primera Medición SPI*

<b>PRIMERA MEDICIÓN</b>			
<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Valor Planificado (PV)</b>	<b>SPI (EV / PV)</b>
ITE1-P1	500	555	0.9009
ITE2-P1	340	422	0.8057
ITE3-P1	400	520	0.7692
<b>Total SPI (P1)</b>			<b>2.4758</b>
ITE1-P2	200	421	0.4751
ITE2-P2	250	342	0.7310
ITE3-P2	250	350	0.7143
<b>Total SPI (P2)</b>			<b>1.9203</b>
ITE1-P3	335	490	0.6837
ITE2-P3	390	495	0.7879
ITE3-P3	325	480	0.6771
<b>Total SPI (P3)</b>			<b>2.1486</b>
ITE1-P4	227	350	0.6486
ITE2-P4	190	370	0.5135
ITE3-P4	196	280	0.7000
<b>Total SPI (P4)</b>			<b>1.8621</b>
ITE1-P5	413	510	0.8098
ITE2-P5	410	530	0.7736
ITE3-P5	436	520	0.8385
<b>Total SPI (P5)</b>			<b>2.4219</b>
ITE1-P6	520	650	0.8000
ITE2-P6	514	610	0.8426
ITE3-P6	538	670	0.8030
<b>Total SPI (P6)</b>			<b>2.4456</b>
ITE1-P7	120	250	0.4800
ITE2-P7	140	240	0.5833

ITE3-P7	115	230	0.5000
<b>Total SPI (P7)</b>			<b>1.5633</b>
ITE1-P8	327	405	0.8074
ITE2-P8	365	475	0.7684
ITE3-P8	369	500	0.7380
<b>Total SPI (P8)</b>			<b>2.3138</b>
ITE1-P9	284	420	0.6762
ITE2-P9	245	390	0.6282
ITE3-P9	230	385	0.5974
<b>Total SPI (P9)</b>			<b>1.9018</b>
ITE1-P10	428	570	0.7509
ITE2-P10	432	590	0.7322
ITE3-P10	472	605	0.7802
<b>Total SPI (P10)</b>			<b>2.2632</b>

**Nota.** Fuente: Elaboración Propia del Autor.

Como se aprecia en la Tabla Anterior se está tomando el total del CPI por cada proyecto del Primera Medición, ya que los proyectos recogidos con el Muestreo fueron 10, estos datos usaremos para realizar la Tabla Final.

**Tabla 10.** Segunda Medición SPI

<b>SEGUNDA MEDICIÓN</b>			
<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Valor Planificado (PV)</b>	<b>SPI (EV / PV)</b>
ITE4-P1	530	510	1.0392
ITE5-P1	463	450	1.0289
ITE6-P1	482	430	1.1209
<b>Total SPI (P1)</b>			<b>3.1890</b>
ITE4-P2	269	270	0.9963
ITE5-P2	250	230	1.0870
ITE6-P2	241	240	1.0042
<b>Total SPI (P2)</b>			<b>3.0874</b>
ITE4-P3	312	305	1.0230
ITE5-P3	356	340	1.0471
ITE6-P3	397	385	1.0312
<b>Total SPI (P3)</b>			<b>3.1012</b>
ITE4-P4	210	215	0.9767
ITE5-P4	140	135	1.0370
ITE6-P4	157	155	1.0129
<b>Total SPI (P4)</b>			<b>3.0267</b>
ITE4-P5	435	430	1.0116
ITE5-P5	427	435	0.9816
ITE6-P5	460	455	1.0110
<b>Total SPI (P5)</b>			<b>3.0042</b>
ITE4-P6	511	505	1.0119

ITE5-P6	563	570	0.9877
ITE6-P6	574	580	0.9897
<b>Total SPI (P6)</b>			<b>2.9893</b>
ITE4-P7	103	110	0.9364
ITE5-P7	174	170	1.0235
ITE6-P7	193	190	1.0158
<b>Total SPI (P7)</b>			<b>2.9757</b>
ITE4-P8	350	340	1.0294
ITE5-P8	392	385	1.0182
ITE6-P8	363	360	1.0083
<b>Total SPI (P8)</b>			<b>3.0559</b>
ITE4-P9	246	240	1.0250
ITE5-P9	273	265	1.0302
ITE6-P9	222	215	1.0326
<b>Total SPI (P9)</b>			<b>3.0877</b>
ITE4-P10	490	485	1.0103
ITE5-P10	483	475	1.0168
ITE6-P10	471	465	1.0129
<b>Total SPI (P10)</b>			<b>3.0401</b>

**Nota.** Fuente: Elaboración Propia del Autor.

Como se aprecia en la Tabla Anterior se está tomando el total del CPI por cada proyecto del Segunda Medición, ya que los proyectos recogidos con el Muestreo fueron 10, estos datos usaremos para realizar la Tabla Final.

### **SPI (Índice del desempeño del programa)**

**Tabla 11.** Puntuación de requerimientos para la elección - SPI

<b>PUNTUACIÓN DE REQUERIMIENTOS PARA LA ELECCIÓN</b>				
<b>MUESTRA</b>	<b>PRIMERA MEDICIÓN</b>	<b>TEST</b>	<b>SEGUNDA MEDICIÓN</b>	<b>D</b>
1	2.4758		3.1890	-0.7132
2	1.9203		3.0874	-1.1671
3	2.1486		3.1012	-0.9525
4	1.8621		3.0267	-1.1646
5	2.4219		3.0042	-0.5824
6	2.4456		2.9893	-0.5436
7	1.5633		2.9757	-1.4123
8	2.3138		3.0559	-0.7421
9	1.9018		3.0877	-1.1859
10	2.2632		3.0401	-0.7768
<b>PROMEDIO</b>				<b>-0.9241</b>
<b>DESVIACIÓN</b>				<b>0.2954</b>
<b>T0</b>				<b>-9.89</b>

**Nota.** Fuente: Elaboración Propia del Autor.

En la tabla presentada se muestra el resumen estadístico para las dos muestras de datos del Índice de desempeño del programa. Para evaluar si las diferencias entre las estadísticas de las dos muestras son significativas. De particular interés es la curtosis estandarizada que pueden usarse para comparar si las muestras provienen de distribuciones normales. Valores de estos estadísticos fuera del rango de -1.96 a +1.96 indican desviaciones significativas de la normalidad, lo que tendería a invalidar las pruebas que comparan las desviaciones estándar. En este caso, los valores de ambas curtosis estandarizadas se encuentran dentro del rango esperado.

## 4.2. Contrastación de la Hipótesis

### Hipótesis Estadística

#### CPI (Índice de desempeño del costo).

**H<sub>0</sub>:** Las metodologías ágiles no influirán en la reducción de los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

**H<sub>1</sub>:** Las metodologías ágiles no influirán en la reducción de los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

$$t_o = \frac{\bar{D} \sqrt{n}}{SD} \quad t_o = \frac{-0.84\sqrt{10}}{0.25} \quad t_o = -10.65$$

Donde

D = Promedio

n = La Población

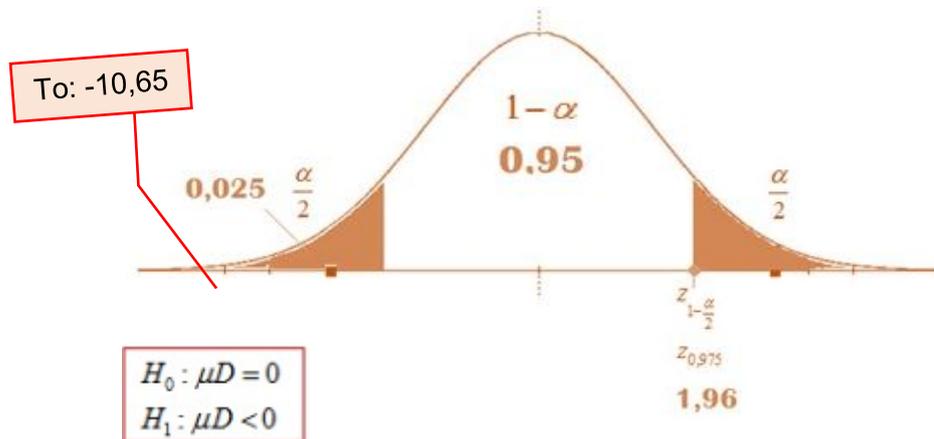
SD = Desviación Estándar

Rechazamos la hipótesis nula

Ho: MD = 0

Hi: MD  $\neq$  0

Con un nivel de confianza del 95%



**Figura 6.T Student – Máxima Demanda CPI**

Fuente: Elaboración Propia.

**Decisión:** Considerando que la T obtenida es -10,65 Se rechaza la hipótesis nula (Ho) y se acepta la hipótesis alterna (H1).

Se demuestra con un nivel de confianza del 95% que Las metodologías ágiles Si influirán en la reducción de los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

### **SPI (Índice de desempeño del programa).**

**Ho:** Las metodologías ágiles no influirán en la optimización del tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

**H1:** Las metodologías ágiles influirán en la optimización del tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.

$$t_o = \frac{\bar{D} \sqrt{n}}{SD} \quad t_o = \frac{-0.92\sqrt{10}}{0.30} \quad t_o = -9.90$$

Donde

D = Promedio

n = La Población

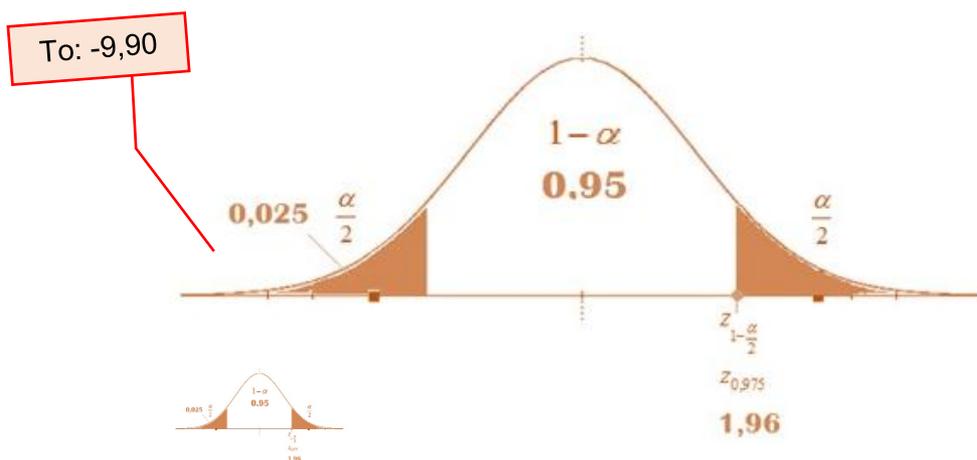
SD = Desviación Estándar

Rechazamos la hipótesis nula

Ho: MD = 0

Hi: MD  $\neq$  0

Con un nivel de confianza del 95%



**Figura 7.** T Student – Máxima Demanda SPI

Fuente: Elaboración Propia.

**Decisión:** Considerando que la T obtenida es -9,90 Se rechaza la hipótesis nula (Ho) y se acepta la hipótesis alterna (H1).

Se demuestra con un nivel de confianza del 95% que Las metodologías ágiles SI influirán en la optimización del tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017

## V. DISCUSIÓN

Uno de los principales Objetivos de esta tesis es que la gestión de proyectos deje de ser un tema aparte al de sistemas y más por el contrario sea un medio por el cual las PYMES puedan obtener una ventaja competitiva, al brindar a sus clientes esa confianza y seguridad de que el software que van a comprar sea de calidad, en el tiempo justo y con los requerimientos presentados, además de ser un gran beneficio para la Pyme que reducirá los costos y tiempos en cada uno de sus proyectos al aplicar la dirección de proyectos bajo esta metodología ágil presentada.

Según (Castillo, 2016) Como conclusión, para distinguir si una empresa genera valor no basta con observar la gestión financiera, sino también otros aspectos como la innovación tecnológica y una estrategia administrativa. Con lo cual en el presente proyecto se muestra que la orientación a la innovación tecnológica es importante, ya que es un elemento vital en el desarrollo de la pyme para hacer frente a la fuerte competencia. Así como también se demuestra que una estrategia administrativa para operar el negocio y dirigir sus operaciones apoyándose en herramientas tecnológicas hace crecer al negocio

A pesar de que las metodologías ágiles están más sonadas estos últimos años aún no se logra aplicar con un gran porcentaje en el país, ya que la cultura organizacional y la desorganización debido al trabajo instintivo y empírico prima sobre una estructura estandarizada de la dirección de proyectos, las metodologías tradicionales de desarrollo de software son elementales para todo profesional, sin embargo existe un gran valor agregado en las metodologías ágiles puesto que estas son flexibles en todo aspecto, crecen constantemente con cada cierre de proyecto, ya que dejan las lecciones aprendidas que son fuente de experiencias y conocimiento para futuros proyectos que se tengan que realizar, brindando una madurez organizacional y profesional para el equipo de trabajo que use estas metodologías. Es tarea de cada líder, gerente o dueño del proyecto determinar cuál es la estructura que se ajuste a la realidad organizacional, y estas nuevas metodologías planteadas son fáciles de adaptarse a todo tipo de entorno, sean pequeños grupos de trabajo o grandes equipos segmentados con roles específicos.

Según (Gimson, 2012) Hace casi dos décadas que se comenzó a buscar una alternativa a las metodologías formales o tradicionales que estaban sobrecargadas de técnicas y herramientas y que se consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Las metodologías ágiles conllevan una filosofía de desarrollo de software liviana, debido a que hace uso de modelos ágiles. Se considera que un modelo es ágil o liviano cuando se emplea para su construcción una herramienta o técnica sencilla, que apunta a desarrollar un modelo aceptablemente bueno y suficiente en lugar de un modelo perfecto y complejo. Existen actualmente una serie de metodologías que responden a las características de las metodologías ágiles y cada vez están teniendo más adeptos. Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con sus postulados y principios, cada metodología ágil tiene características propias y hace hincapié en algunos aspectos más específicos.

## VI. CONCLUSIÓN

Las Metodologías Tradicionales han tenido buenos resultados en proyectos que no tienen cambios frecuentes en su alcance y están sujetos a un plan. Mientras que las Metodologías Ágiles tienen buenos resultados en proyectos que son muy cambiantes debido al entorno del negocio, si bien fue concebido originalmente para proyectos de software donde se ha tenido éxito, en la actualidad lo utilizan en empresas de varios sectores.

Se puede aplicar metodologías ágiles a proyectos que tienen personal experimentado al menos en puestos claves, ya que estos son equipos auto organizados y multidisciplinarios, los miembros del equipo pueden desarrollar sus habilidades, conocimientos y experiencias de tal forma que comparten el mismo nivel de responsabilidades dentro del proyecto, y sobre todo donde exista una alta participación y comunicación entre el cliente y el equipo de trabajo. En las metodologías ágiles el Scrum Master es un colaborador antes que un jefe del proyecto.

Las actitudes del equipo ágil es son más proactiva desde el instante que empieza el Sprint, debido a que al finalizar la iteración los responsables deben entregar un producto funcional que cumpla los requerimientos del cliente, mientras que un equipo de un proyecto que aplique una metodología tradicional libera sus entregables por lo general al final de todo el proyecto y es recién ahí cuando el equipo se ve presionado para cumplir con las fechas del cronograma inclusive realizando horas extras para alcanzar el plan. Además, las pruebas se ven afectadas ya que es una de las últimas fases, cuando los problemas salen a cubierto es demasiado tarde para corregirlos ya que los requerimientos pueden haber cambiado y las funcionalidades ya no son útiles para el cliente.

El tratamiento de los riesgos está implícito en la misma metodología ágil ya que todos los días se tratan los impedimentos o riesgos que el equipo detecta en el transcurso del Sprint para que sean gestionados por el Scrum Master en su función de colaborador con el equipo de trabajo.

Contrario a lo que creen los equipos ágiles si planifican y documentan su trabajo este será mucho más sencillo ya que el nivel de detalle necesario se encuentra

en las Historias de Usuario y los criterios de aceptación que son definidos en cada iteración.

Las metodologías ágiles Scrum y XP son muy usadas en la actualidad debido a la flexibilidad que tienen al momento de aplicarlas juntas en cada iteración, la tendencia es emplearlas juntas como una metodología híbrida ya que Scrum se orienta a la gestión de proyectos mientras que XP interviene en el desarrollo mismo del producto.

La situación actual de la gestión de proyectos en Perú viene creciendo paulatinamente, ya que la mayoría de empresas tienen una cultura informal tanto a nivel de proveedores como de clientes, por tal motivo los proyectos se enfocan más en la elaboración técnica antes que en la gestión, dando menor importancia a temas relacionados con la comunicación, integración, riesgos, costos, seguimiento y control que se necesita para que un proyecto de desarrollo de software sea exitoso.

A nivel mundial se va notando la tendencia de las empresas que buscan y cuentan con personal capacitado en la dirección de proyectos con certificaciones en metodologías ágiles como Scrum y también organizaciones como Project Management International, el profesional debe de poder trabajar con una metodología de gestión tradicional o ágil ya que estas pueden convivir para generar resultados efectivos y eficientes.

Bits & Systems Solutions S.A.C no tiene una metodología de trabajo adecuada que se adapte a las necesidades y cambios constantes que presentan sus proyectos, situación que ha ocasionado problemas considerables tanto en la gestión como en el desarrollo de proyectos de software, por tal motivo los clientes estas insatisfechos, y genera un sobre costo de dinero y tiempo en los proyectos

## VII. RECOMENDACIÓN

Antes de tomar una decisión sobre la metodología a utilizar en sus proyectos las empresas de desarrollo de software deben hacer un análisis de las características propias del negocio, así como la estructura organizacional de la empresa y sobre todo la relación que se tiene con el cliente, puesto que gracias a ese análisis se podrá determinar la aplicabilidad o no de una metodología y el desarrollo normal del proyecto de Software.

Las metodologías ágiles no hablan de un tratamiento formal de los riesgos pero tampoco descartan el uso de técnicas o herramientas que permitan detectar inconvenientes que se vayan presentando a lo largo del proyecto, para así poder corregirlos en cada Sprint, el factor del CPI actual y el SPI es una de las herramientas más rápidas, económicas y efectivas para determinar el estado del proyecto, y como se encuentran los riesgos, esto permitirá al Scrum Master realizar el seguimiento y control de para ser atendidos oportunamente.

Se recomienda a la empresa el uso de herramientas de apoyo para el proceso de las metodologías ágiles que permitan consolidar información para una mejor planificación, seguimiento y control del proyecto como por ejemplo para versionar el código, controlar la calidad del software con el uso de pruebas unitarias e integradoras, analizadores de código, buscadores de errores para la administración de proyectos ágiles que incorporan indicadores y gráficos de control.

Se recomienda con énfasis el registro de las Lecciones Aprendidas para temas relacionados con riesgos, contracción de recursos, adquisiciones, control de la calidad, desempeño del equipo, a pesar de que las metodologías ágiles no detallan específicamente este proceso, es de vital importancia realizar una retroalimentación de las experiencias obtenidas a los largo del desarrollo del proyecto, y el trabajo que se realizó en cada Sprint, de tal manera que todos estos conocimientos adquiridos sirvan de apoyo al equipo para la toma de decisiones basados en situaciones similares de proyectos anteriores que les ayude mejorar los resultados encaminados al éxito del proyecto de desarrollo de Software.

Se recomienda a la empresa Bits& Systems Solutions S.A.C usar la propuesta metodológica planteada en esta tesis para la dirección de sus proyectos de desarrollo de software, estos pueden ser pequeños o grandes ya que es una metodología flexible que permitirá gestionar proyectos de forma interactiva e incremental, puesto que en cada entrega del Sprint el cliente obtendrá resultados tempranos que satisfagan sus necesidades, cubran los requerimientos, el equipo alcance una mejor productividad y calidad en el desarrollo de los productos de software al estar comprometidos y motivados, la empresa reduzca los costos y tiempos de entrega de los proyectos asegurando y posicionando la empresa en el mercado.

Se recomienda a la empresa Bits & Systems Solutions S.A.C una vez que haya aplicado la metodología ágil planteada en esta tesis para la dirección de proyectos de desarrollo de software realice una evaluación de los resultados de la efectividad de misma y de esta manera sugerir la aplicabilidad bajo testimonio real a otras empresas que se encuentren en situaciones similares.

## REFERENCIA BIBLIOGRÁFICA

**Toro, F.** (2013). *Administración de Proyectos de Informática*. Primera edición: Bogotá: Eco Ediciones.

**Mitaritonna, A.** (2010). *Una innovadora metodología para el desarrollo de software en Ambientes de trabajo virtuales*. Recuperado diciembre 2017.

**Carvajal R. José.** (2008). *Tesis, Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones java EE como metodología empresarial*. Recuperado noviembre 2017.

**Evangelista D., Huayamares K., Muñoz K.** (2012). *Metodologías Tradicionales-Ágiles* Recuperado diciembre 2017.

**Al-Balushi, T., Al Badi, R., & Ali, S.** (2012). *Influential factors of B2B E-Commerce Acceptance in SMEs' Structure and Process*.

**Guerrero Cando, Renán Mauricio; Guerrero Herrera, María Fernanda.** (2015). *Desarrollo de un sistema web de comercio electrónico B2C, para la promoción, compra on-line y gestión de stock de artículos de cuero*. (Tesis de pregrado, Escuela Politécnica Nacional - Ecuador).

**Kniberg, H.** (2007). *Scrum y XP desde las Trincheras: Cómo hacemos Scrum*. Estados Unidos: Editorial C4Media.

**Palacio, J. & Ruata, C.** (2011). *Scrum Manager Gestión de Proyectos*. Feria Informática.

**J. G. Schneider y R. Vasa,** (2006) «*Agile practices in software development-experiences from student projects,* » de *Software Engineering Conference*.

**CHIN, Gary** (2004) *Agile Project Management: How to succeed in the face of changing project requirements*. Primera Edición. New York: AMACOM.

**JALALI, Samireh y WOHLIN** (2010) "*Agile Practices in Global Software engineering a systematic map*". International Conference on Global Software Engineering (ICDSE).

**YAGGAHAVITA, Hasith,** (2011) *Challenges in Applying Scrum Methodology on Culturally Distributed Teams*. Universidad Sheffield Hallam, Facultad de Artes, Computación, Ingeniería y Ciencias (ACES).

**Gerencia de Sistemas y Proyectos.** (2012). *Gestión de Proyectos PMI y el Desarrollo Ágil: ¿Que tienen en común?*

**Gimson Loraine.** (2012). *Metodologías ágiles y desarrollo basado en conocimiento*.

**Góngora G.** (2010). *Tecnología de la información como herramienta para aumentar la productividad de una empresa*. Recuperado Noviembre 2017.

**Lone S. Carlos.** (2011). *Aplicación de los Principios de Proyectos del PMI a Metodologías Ágiles de Desarrollo de Software*.

**Luna L.** (2012). *Metodología Ágil vs Metodología Tradicional*. Recuperado Octubre 2017.

**Bahit Eugenia** (2012). *Scrum Extreme Programming Para Programadores*.

**Herranz Raúl** (2014). *Scrum Manager En Busca De La Excelencia Del Código*.

**INEI** (2015) *Empresas Software en el Perú*, Informe Anual.

**Deemer P., Benefield G., Larman C., & Vodde B** (2009). "*The Scrum Primer*".

# ANEXOS

## 1. Matriz de Consistencia

Matriz de Consistencia									
TITULO: Metodologías ágiles para la optimización de la dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C - Lima - 2017.									
PROBLEMA	OBJETIVO	HIPÓTESIS		VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ESCALA DE INDICADORES
		GENERAL	GENERAL						
¿De qué manera las Metodologías ágiles influye en la optimización de la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017?	Establecer una metodología ágil que optimice la dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017.	Las metodologías ágiles influirán en la optimización de la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017	GENERAL	VI: Metodologías ágiles	Es el conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que rige una investigación o tareas que requieran habilidades, conocimientos o cuidados específicos.	Se hizo uso de diversas técnicas e instrumentos con el objetivo de que la muestra sea confiable y el resultado lo más preciso posible, para la variable independiente se trabajaron con indicadores propios de la metodología ágil como tablas de medición.	D1. Scrum  D2. XP	Tabla Burndown  Velocidad	Iteraciones en el Proyecto  Líneas de Código Programadas
¿De qué manera las metodologías ágiles influye en la optimización en los costos de desarrollo de software?	Establecer una metodología ágil que reduzca los costos de desarrollo de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017	Las metodologías ágiles influirán en la optimización en los costos de desarrollo de software	SECUNDARIOS	Dirección de Proyectos de Software	La dirección de proyectos, es la aplicación del conocimiento, de las habilidades, y de las técnicas para ejecutar los proyectos en forma eficiente y efectiva. Es una competencia estratégica para las organizaciones, y les permite alcanzar los resultados de los proyectos a las metas del negocio, y así competir mejor en su mercado.	Se hizo uso de diversas técnicas e instrumentos con el objetivo de que la muestra sea confiable y el resultado lo más preciso posible, para la variable dependiente se realizó una matriz de observación tomando como muestra 5 proyectos, y determinando sus indicadores de desempeño	D1. Costo	CPI (Índice de desempeño del costo)	% de CPI
¿De qué manera las metodologías ágiles influye en la optimización en el tiempo de entrega de los proyectos?	Establecer una metodología ágil que optimice el tiempo de entrega de los proyectos en la empresa Bits & Systems Solutions S.A.C, Lima, 2017	Las metodologías ágiles influirán en la optimización en el tiempo de entrega de los proyectos.	SECUNDARIOS				D3. TIEMPO	SPI(Índice del desempeño del programa)	% DE SPI

## 2. Matriz de Operacionalización

VARIABLES	DIMENSIONES	INDICADORES	NUMERO DE ITEMS	ESCALA DE MEDICION	NIVELES Y RANGOS
VI: Metodología de Trabajo	Scrum	Back Log de Sprint		Iteraciones en el Proyecto	0-100%
	Extreme Programming XP	Short Release		Líneas de Código Programadas	0-100%
VD: Dirección de Proyectos de Software	Costo	CPI (Índice de desempeño del costo).	<b>CPI= EV/AC</b>	% de CPI	0-1
		SPI (Índice del desempeño del programa)	<b>SPI= EV/PV</b>	% DE SPI	0-1

### 3. Validación de Instrumentos de Recolección de Datos

ANEXO N° 01

#### CARTA DE PRESENTACIÓN

Señor(a)(ita):

---

**Presente**

**Asunto: VALIDACIÓN DE INSTRUMENTOS A TRAVÉS DE JUICIO DE EXPERTO.**

Me es muy grato comunicarme con usted para expresarle mi saludo y así mismo hacer de su conocimiento que, siendo estudiante de la carrera ingeniería de Sistemas, promoción 2016, aula A, requiero validar los instrumentos con los cuales debo recoger la información necesaria para poder desarrollar la investigación para optar el Título Profesional de Ingeniero de Sistemas.

El título o nombre del proyecto de investigación es: ***Metodologías ágiles para la optimización en la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C – Lima – 2017***, y siendo imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos, recorro y apelo a su connotada experiencia a efecto que se sirva aprobar el instrumento aludido.

El expediente de validación, que le hacemos llegar contiene:

- Carta de presentación.
- Definiciones conceptuales de las variables, dimensiones indicadores.
- Certificado de validez de contenido de los instrumentos.
- Operacionalización de las variables.

Expresándole mi sentimiento de respeto y consideración me despido de usted, no sin antes agradecerle por la atención que dispense a la presente.

Atentamente.

---

**Firma**  
**Carlos Enrique Soto Condori**  
**D.N.I: 72758288**

## **DEFINICIÓN CONCEPTUAL DE LAS VARIABLES, DIMENSIONES E INDICADORES**

### **VARIABLE 1: Metodologías ágiles.**

(Eugenia Bahit 2012, 13) Es una metodología de gestión de proyectos adaptativa, que permite llevar a cabo, proyectos de desarrollo de software, adaptándose a los cambios y evolucionando en forma conjunta con el software.

#### **Dimensiones de la variable 1:**

##### **Dimensión 1: Scrum.**

(Eugenia Bahit 2012, 30) Es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprint. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprint son de duración fija – terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos.

#### **Indicador:**

##### **Back Log de Sprint.**

(Eugenia Bahit 2012, 45) Es una lista reducida de ítems del Back log de Producto, que han sido negociados entre el Dueño de Producto y el Scrum Team durante la planificación del Sprint. Esta lista, se genera al comienzo de cada Sprint y representa aquellas características que el equipo se compromete a desarrollar durante la iteración actual. Los ítems incluidos en el Back log de Sprint se dividen en tareas las cuales generalmente, no demanden una duración superior a un día de trabajo del miembro del equipo que se haya asignado dicha tarea.

##### **Dimensión 2: Extreme Programming XP.**

(Eugenia Bahit 2012, 70) Es una metodología que tiene su origen en 1996, de la mano de Kent Beck, Ward Cunningham y Ron Jeffries. A diferencia de Scrum, XP propone solo un conjunto de prácticas técnicas, que aplicadas de manera simultánea, pretenden enfatizar los efectos positivos de en un proyecto de desarrollo de Software.

## **Indicador:**

### **Short Release.**

Se busca hacer entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración. Esto conlleva a que el cliente pueda tener una mejor experiencia con el Software, ya que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras con mayor facilidad de implementación.

## **VARIABLE 2: Dirección de Proyectos de Software.**

(Maigua y López 2012, 15) La dirección de proyectos es la aplicación del enfoque de sistemas para la administración de tareas tecnológicas complejas o de proyectos cuyos objetivos se establecen explícitamente en términos de tiempo costes y parámetros de realización.

### **Dimensiones de la variable 2:**

#### **Dimensión 1: Costo.**

(Maigua y López 2012, 74) Es un proceso que consiste en el desarrollo de una aproximación de los recursos monetarios que se necesitarán para completar las actividades que demandará el proyecto. Es una predicción hecha sobre la base de la información que está disponible en un determinado momento.

### **Indicadores**

#### **CPI (Índice de desempeño del costo).**

(Maigua y López 2012, 79) Es un índice que expresa la "eficiencia" en los costos reales del proyecto, comparando el Valor Ganado (costo presupuestado para el trabajo realizado), versus el Costo Real. Si el Valor Ganado es igual al Costo Real, diríamos que el trabajo ha costado lo previsto, y el CPI sería igual a 1. Si el Valor Ganado fuese menor al Costo Real, querría decir que el trabajo realizado (Valor Ganado) ha costado más que lo previsto, en cuyo caso el CPI sería menor a 1.

- Un CPI menor a 1 indica un desempeño peor al previsto,
- Un CPI mayor a 1 indica un desempeño mejor al previsto.

#### **Dimensión 2: Tiempo.**

(Maigua y López 2012, 53) Es el proceso de Desarrollar el Plan para la Dirección del Proyecto que produce un plan de gestión del cronograma que selecciona una metodología, una herramienta de planificación, y establece el formato y los criterios para desarrollar y controlar el cronograma del proyecto. Una metodología de planificación define las reglas y enfoques para el proceso de elaboración del cronograma. Entre las metodologías más conocidas, se encuentran el método de la ruta crítica y el de la cadena crítica.

## Indicadores

### SPI (Índice del desempeño del programa)

(Maigua y López 2012, 59) Es un índice que compara el EV (Valor Ganado), es decir lo avanzado, contra el PV (Valor Planeado) lo que se tenía pensado avanzar a un momento dado.

- Si el SPI es igual a 1, quiere decir que el entregable se está avanzando al ritmo previsto durante el presupuesto.
- Si el SPI es mayor a 1, quiere decir que el entregable se está avanzando a un ritmo mayor al previsto en el presupuesto.
- Si el SPI es menor a 1, quiere decir que se está avanzando a un ritmo peor que el previsto.

## ANEXO N° 03

### CERTIFICADO DE VALIDEZ DE CONTENIDO DE LOS INSTRUMENTOS

N°	Dimensiones / ítems	Pertinencia <sup>1</sup>		Relevancia <sup>2</sup>		Claridad <sup>3</sup>		Sugerencias
		Si	No	Si	No	Si	No	
	<b>I. Costo</b>							
1	<b>CPI (Índice de desempeño del costo).</b> EV: Valor Ganado AC: Costo Actual  Fórmula: $CPI = EV/AC$							
	<b>II. Tiempo</b>							
2	<b>SPI(Índice del desempeño del programa)</b> EV: Valor Ganado PV: Valor Planificado  Fórmula $SPI = EV/PV$							

**ANEXO N° 04**

**INSTRUMENTOS DE LAS VARIABLES**

**CPI (Índice de desempeño del costo).**

<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Costo Actual (AC)</b>	<b>CPI (EV / AC)</b>	<b>RESULTADO</b>
A1-P1				
A2-P1				
A3-P1				
A4-P1				
A1-P2				
A2-P2				
A3-P2				
A4-P2				
<b>Total</b>				

**SPI(Índice del desempeño del programa)**

<b>Actividades</b>	<b>Valor Ganado (EV)</b>	<b>Valor Planificado (PV)</b>	<b>SPI (EV / PV)</b>	<b>% COMPLETADO</b>
A1-P1				
A2-P1				
A3-P1				
A4-P1				
A1-P2				
A2-P2				
A3-P2				
A4-P2				
<b>Total</b>				

Observaciones (precisar si hay suficiencia): \_\_\_\_\_

Opinión de aplicabilidad: Aplicable [ ] Aplicable después de corregir [ ] No aplicable

[ ]

Apellidos y nombres del juez validador. Dr/ Mg:

.....

DNI:.....

Especialidad del validador:.....

<sup>1</sup>**Pertinencia:** El ítem corresponde al concepto teórico formulado.

<sup>2</sup>**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

<sup>3</sup>**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

**Nota:** Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

#### 4. Matriz de Datos

Matriz de Consistencia								
TITULO: Metodologías ágiles para la optimización en la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C - Lima - 2017.								
PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ESCALA DE INDICADORES
PRINCIPAL	GENERAL	GENERAL						
¿De qué manera las Metodologías ágiles influye en la optimización de la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017?	Las metodologías ágiles influye en la optimización de la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017	Las metodologías ágiles influirá en la optimización de la gestión de dirección de proyectos de software en la empresa Bits & Systems Solutions S.A.C, Lima, 2017	VI: Metodologías ágiles	Es el conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que riga una investigación o tareas que requieran habilidades, conocimientos o cuidados específicos.	Se hizo uso de diversas técnicas e instrumentos con el objetivo de que la muestra sea confiable y el resultado lo más preciso posible, para la variable Independiente se trabajaran con indicadores propios de la metodología ágil como tablas de medición.	D1. Scrum	Tabla Burndown	Iteraciones en el Proyecto
						D2. XP	Velocidad	Líneas de Código Programadas
SECUNDARIOS	SECUNDARIOS	SECUNDARIOS	VARIABLES	DEFINICION CONCEPTUAL	DEFINICION OPERACIONAL	DIMENSIONES	INDICADORES	ESCALA DE INDICADORES
¿De qué manera las metodologías ágiles influye en la optimización en los costos de desarrollo de software?	Las metodologías ágiles influye en la optimización en los costos de desarrollo de software	Las metodologías ágiles influirá en la optimización en los costos de desarrollo de software	VD: Dirección de Proyectos de Software	La dirección de proyectos, es la aplicación del conocimiento, de las habilidades, y de las técnicas para ejecutar los proyectos en forma eficiente y efectiva. Es una competencia estratégica para las organizaciones, y les permite atar los resultados de los proyectos a las metas del negocio, y así competir mejor en su mercado.	Se hizo uso de diversas técnicas e instrumentos con el objetivo de que la muestra sea confiable y el resultado lo más preciso posible, para la variable dependiente se realizara una matriz de observación tomando como muestra 5 proyectos, y determinando sus indicadores de desempeño	D1. Costo	CPI (Indice de desempeño o del costo)	% de CPI
						D2. Alcance	KPI (Indicador clave de desempeño)	% de KPI
¿De qué manera las metodologías ágiles influye en la optimización en el tiempo de entrega de los proyectos?	Las metodologías ágiles influye en la optimización en el tiempo de entrega de los proyectos.	Las metodologías ágiles influirá en la optimización en el tiempo de entrega de los proyectos.				D3. TIEMPO	SPI (Indice del desempeño o del programa)	% DE SPI

## **5. Propuesta de Valor**

El presente trabajo busca ser una guía para que la gestión del desarrollo de software en la Empresa Bits & Systems Solutions S.A.C., sea tratado como un proyecto y no como un simple desarrollo que además de reducir los costos y tiempos, se vaya de la mano con la calidad de los productos de software. Se busca combinar las mejores metodologías ágiles para la dirección de proyectos de Software descritos en la metodología ágil SCRUM y XP, que se fusionan, para cumplir con todo el ciclo para la entrega de un buen producto de software, dando lineamientos como por ejemplo: criterios estandarizados para el correcto desempeño, plantillas de Trabajo, comunicación antes, durante y después del proyecto, estimación del tiempo, la identificación y la solución de problemas y defectos dentro del desarrollo del proyecto, además de gestionar y validar cualquier cambio en los requerimientos y como priorizar estos.

Otro punto importante es la base de conocimientos para el registro y recuperación de información de acuerdo a las necesidades para que den soporte a la toma de decisiones. Los puntos recolectados en la empresa que se adaptan a la metodología de proyectos ágiles son los siguientes:

- La mayoría de los requisitos establecidos para los proyectos de desarrollo de software cambian de manera muy frecuente durante el ciclo de vida del proyecto.
- Utilizan componentes de programación genéricos y estandarizados.
- Trabajan con equipos pequeños máximo 6 personas.
- Es más importante tener un software funcionando frente a la documentación.

El gerente de software de la empresa Bits & Systems Solutions S.A.C expreso que usan con mayor frecuencia documentación realizada por ellos mismos o muchas veces no se dan el tiempo para poder gestionar estos documentos o reuniones como buenas prácticas, sin embargo, expresan de manera positiva el haber escuchado o conocido la utilidad que representa trabajar con las metodologías ágiles, por lo cual podemos expresar que estos aspectos facilitaran el uso y comprensión de la guía propuesta en este proyecto de tesis.

- El tiempo estimado para el desarrollo de los requerimientos esta entre Insuficiente, Regular y Suficiente con mayor votación en Regular.
- No existe claridad a la hora de elaborar los requerimientos.
- No cuentan con suficientes herramientas tecnológicas de soporte para el desarrollo y gestión de proyectos.
- Llevan una comunicación informal en todo el ciclo de vida del proyecto.

### **Estrategia de Desarrollo**

Una vez revisados los métodos ágiles más utilizados para la gestión de proyectos y en base a las necesidades encontradas en la empresa Bits & Systems Solutions S.A.C., se va a desarrollar la siguiente estrategia:

1. Aplicar varios de las herramientas aportadas por las Metodologías Ágiles de acuerdo con las necesidades de los proyectos de desarrollo de software.
2. Aplicar las directrices adecuadas de las metodologías SCRUM y XP de tal manera que se complementen en la guía propuesta.
3. Desarrollar la metodología en base a los puntos anteriores donde se determinen los procesos y entregables que se ajusten a la realidad de la empresa Bits & Systems Solutions S.A.C.
4. Establecer reuniones acordes a las metodologías ágiles, adicional a esto establecer sesiones en cada término de los Sprint con el fin de realizar una retroalimentación y comunicar las lecciones aprendidas, para aplicar estos conocimientos en los proyectos a futuro.



**Figura 8.** Beneficios de la Metodología

Fuente: Elaboración Propia.

## Herramientas Ágiles

Es importante aclarar que la guía propuesta puede aplicarse sin utilizar las herramientas de gestión de proyectos que existen en el mercado, pero sin duda estas ayudan a simplificar en gran medida las actividades de gestión, planificación, control y seguimiento de los proyectos.

Se recomienda algunas de las herramientas ágiles que existen en el mercado y que sirven de apoyo para la gestión y desarrollo de proyectos de software, las mismas que permiten aumentar la productividad de las empresas y la calidad de los productos.

Debido a que los lenguajes de programación orientada a objetos que utiliza la empresa Bits & Systems Solutions S.A.C. para el desarrollo de sus proyectos de software es Visual Fox Pro y la mayoría de proyectos a futuro van en este sentido se recomienda herramientas para estos lenguajes.

## Herramientas Para Soporte de Software

**No Magic**, es una herramienta de modelado de arquitectura diseñado para programadores, ingenieros de control de calidad, facilita el análisis y diseño de sistemas y bases de datos orientada a objetos (OO). A diferencia de otros

modelos UML y los entornos de arquitectura, MagicDraw hace que sea fácil implementar un entorno de Ciclo de Vida de Desarrollo de Software (SDLC).

**Django**, Framework Python que promueve el desarrollo rápido y el diseño limpio.

**Pylons**, Framework web para Python que enfatiza la flexibilidad y el desarrollo rápido.

**Eclipse y Netbeans**, Frameworks promueve el desarrollo de aplicaciones en java y php.

### **Herramientas Para Control de Versiones**

Concurrent Versions System (CVS), es el más popular sistema de control de versiones. Puede almacenar el historial de los archivos fuente y documentos.

**Subversión (SVN)**, es un software de sistema de control de versiones, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo y no un número de revisión de cada archivo por separado.

**Git**, es una herramienta distribuida de control de revisiones desarrollada para administrar el árbol del kernel de Linux, rapidez en la gestión de ramas.

### **Herramientas Para Control de Código**

Checkstyle, permite dar estilo de codificación como, por ejemplo: el nivel de indentación, utilizar espacios o tabuladores, añadir comentarios, cómo organizar las llaves de los bloques de código, el tamaño máximo de la línea, etc. La uniformidad de estilo a la hora de codificar facilita el grado de cohesión del equipo de desarrollo, la mantenibilidad de la base de código y en definitiva la productividad. Si todo el equipo trabaja con un mismo estilo, puede entender la estructura del programa más fácilmente, es una herramienta que genera informes del nivel de seguimiento de estas convenciones. Se puede rebajar el nivel de exigencia decidiendo para qué convenciones se quiere generar una alarma y cuáles se puede ignorar o incluso para personalizar convenciones.

**Simian**, es un detector de copia de bloques de código. La replicación de código, en general, es menos justificable e introduce un gran coste en mantenibilidad y calidad.

**PMD y FindBugs**, son herramientas de análisis en busca de estructuras potencialmente peligrosas tales como: posibles bugs, código “muerto” (variables no accedidas, bloques de ejecución inalcanzables).

### **Herramientas de Gestión de Proyectos**

Jira, es un software de gestión de proyectos para el seguimiento de errores, administración de tareas. Permite conectarse con otras herramientas para asociar las tareas con el código fuente de esta manera se puede supervisar el progreso del trabajo.

**Green Hopper**, es un complemento de Jira que integra de forma fácil los proyectos de Jira, Planifica proyectos de forma ágil usando kamban y Scrum.

**VersionOne**, permite gestionar proyectos ágiles basados en Scrum, XP, Kamban. API's para integración con tecnologías .NET y Open Source.

### **Normas Generales Propuestas**

1. Los proyectos de desarrollo de software deben estar sustentados en un plan de gestión, los mismos que deben estar aprobados por los responsables del proyecto.
2. Antes de empezar un proyecto de desarrollo de software este deberá contar con un documento habilitante o contrato donde se especifiquen los compromisos adquiridos entre la empresa desarrolladora y el cliente.
3. Todo cambio, riesgo o retraso no contemplado en el plan, deberá ser notificado al líder de proyecto o Scrum Master y a los Interesados del proyecto.
4. Todo cambio deberá ser analizado y aprobado de acuerdo a los términos establecidos en el contrato.
5. Todos los cambios deberán ser analizado por todo el equipo del proyecto.
6. Antes de cerrar una etapa, iteración, reléase del proyecto se debe haber completado los entregables establecidos definidos en la etapa.
7. Todo integrante del equipo del proyecto debe informar en cualquier momento al responsable del proyecto (Product Owner) la presencia de riesgos, así como cambios que afecten a la calidad, tiempo, costos del

proyecto para que el responsable comunique a los interesados y se pueda tomar acciones al respecto.

8. Cada etapa de cierre de la iteración incluirá la aceptación formal y pruebas que certifiquen la calidad y el cumplimiento de los objetivos de entregar un producto funcional con valor.
9. Se deberá de analizar y retroalimentar las lecciones aprendidas en cada cierre de un proyecto con todos los integrantes del equipo.

### Dirigir el Desarrollo del Software

Las solicitudes de cambio por parte de los clientes están permitidas al inicio de cada iteración de tal manera que son incluidas en la planificación del Sprint. Los cambios son considerados como algo natural para que las necesidades del cliente sean satisfechas. El proceso de cambios en las metodologías ágiles es sencillo.

**Tabla 12.** Historia del Usuario

Historia de Usuario	
<b>Número: 1</b>	<b>Usuario: Vendedora</b>
<b>Nombre Historia: Introducción de pedido (cliente preferente)</b>	
<b>Prioridad en negocio: Alta</b>	<b>Riesgo en desarrollo: Baja</b>
<b>Puntos estimados: 4</b>	<b>Iteración asignada: 1</b>
<b>Programador responsable: Jordhy Egusquiza</b>	
<b>Descripción:</b> Crear un módulo para poder gestionar los pedidos de los clientes con más de 2 años en la empresa, otorgándoles mensualmente descuentos según una tabla mantenedora (Ofertas).	
<b>Observaciones:</b> La vendedora será exclusiva para estos tipos de clientes preferentes (perfil único).	

**Nota.** Fuente: Elaboración Propia del Autor.

### Monitorear y Controlar el Proyecto - Sprint

Realizar el seguimiento y control del avance del Sprint en el Proyecto, revisar e informar el progreso del Sprint a fin de cumplir con los objetivos de desempeño

planteados, de esta manera los interesados pueden conocer el estado actual del proyecto.

Para la presente tesis con las metodologías ágiles planteadas se hará uso del gráfico burndown donde se representa diariamente la cantidad de trabajo restante del Sprint para saber si estamos retrasados o adelantados de la planificación inicial. También utilizan burndown a nivel de versión para conocer cuántos puntos de historia quedan pendientes en la pila después de cada Sprint.

**Gráfico Burn-Down**, Muestra la situación actual del sprint de forma gráfica y dinámica, este se actualiza diariamente, el eje X representa los días del Sprint sin tomar en cuenta fines de semana. El eje Y representa los puntos de historia del Sprint.

El Scrum Master es el responsable de asegurarse de que el equipo de trabajo se active cuando se presentan los siguientes inconvenientes.

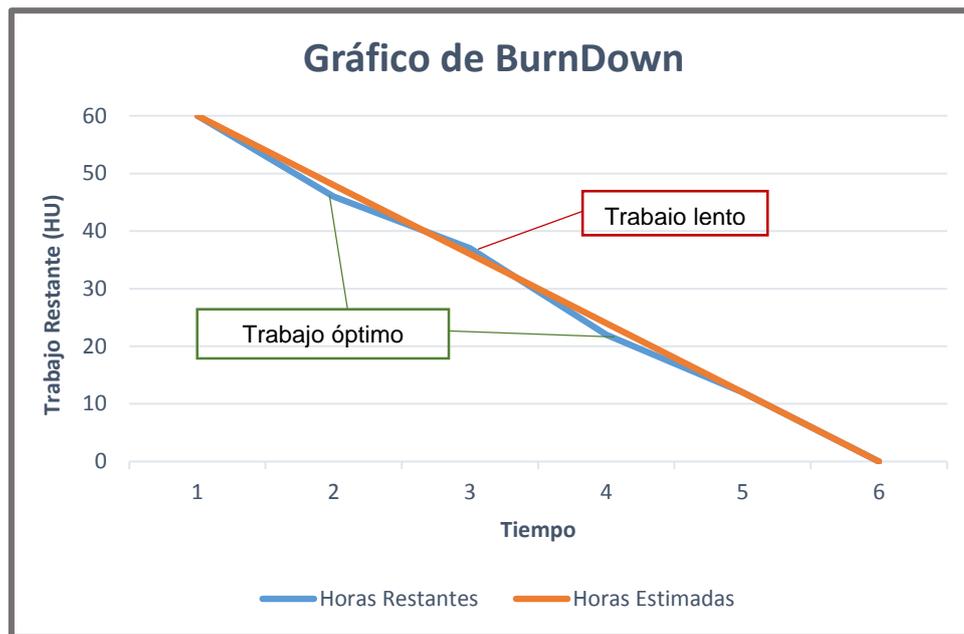
- Elaboramos una tabla colocando los días de cada de la ITERACION que estamos trabajando, correlacionándolas con las Historias de Usuario que se trabajaran en el Sprint, se realiza con el fin de que se pueda supervisar el Trabajo realizado de forma diaria, plasmándolo el avance en un gráfico.

**Tabla 13.** Tabla Burn-Down

	Estimado (Horas)	SPRINT - (Historias de Usuario)					Total de Horas (Reales)
		Dia 5	Dia 4	Dia 3	Dia 2	Dia 1	
HU 1	8	3	1	2	1	1	8
HU 2	4	1	1	2	0	1	5
HU 3	6	2	0	2	0	2	6
HU 4	5	1	2	2	0	1	6
HU 5	5	1	0	1	1	0	3
HU 6	6	2	1	1	3	1	8
HU 7	8	3	3	1	0	1	8
HU 8	6	1	1	0	0	4	6
HU 9	8	0	0	4	2	0	6
HU 10	4	0	0	0	3	1	4
<b>Horas Restantes</b>	60	46	37	22	12	0	
<b>Horas Estimadas Restantes</b>	60	48	36	24	12	0	

**Nota.** Fuente: Elaboración Propia del Autor.

- Cuando el gráfico presentado registra puntos de historia sobre la línea base del Sprint indica que existen demasiadas historias y que el equipo no avanza en el tiempo previsto para desarrollarlas.
- Cuando el gráfico presentado registra puntos de historia bajo la línea base del Sprint indica que las historias de usuario se están desarrollando más rápido de lo previsto y podrían añadir más puntos de historia.



**Figura 9.**Gráfico Burn-Down

Fuente: Elaboración Propia.

Tablero Scrum, Este tablero que actúa como radiador visual de información para realizar el seguimiento del estado del progreso de las historias de usuario del Sprint, de tal manera que el equipo o cualquier interesado conozcan del avance del Sprint, el estado de este y cuáles son los puntos específicos de cada iteración.

Cada fila del tablero Scrum es la historia de un usuario. Durante la reunión de la carga de trabajo, el equipo selecciona los elementos de la carga de trabajo del producto que deben completar.

Cada una de estas tareas se representa por una tarjeta colocada sobre el tablero. El equipo actualiza el tablero Scrum continuamente. Antes o durante el Scrum diario, las estimaciones cambian y las tarjetas se mueven alrededor del tablero.

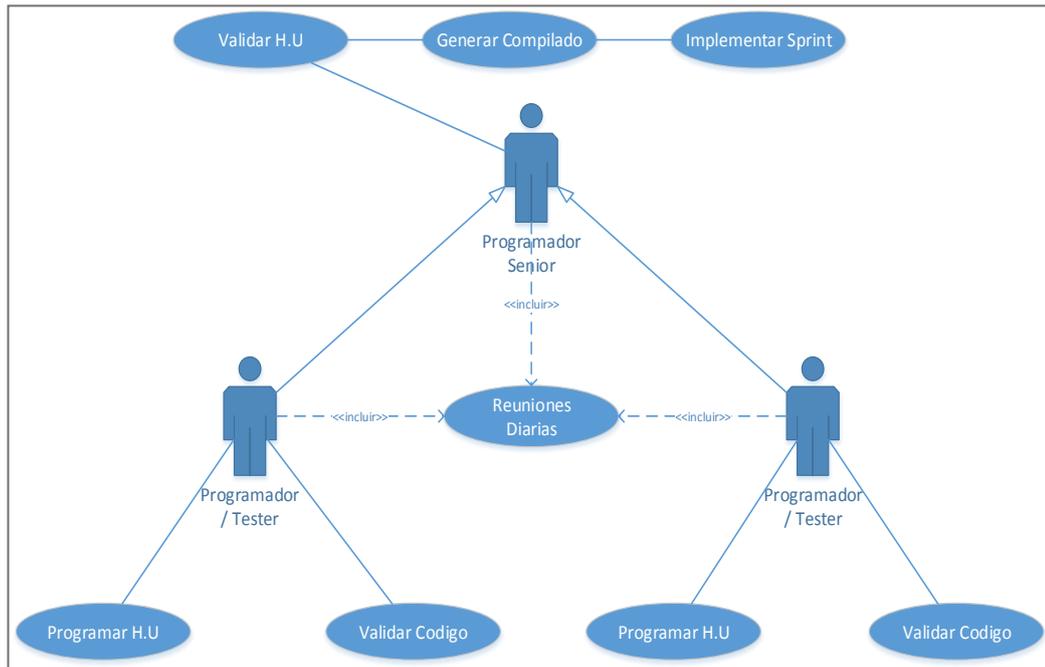
- Historia.
- Por hacer.
- En progreso.
- Para verificar.
- Hecho.

	Historia 2 / 2	Por Hacer 4 / 4	En Progreso 10 / 10		Hecho
			Acción	Verificando	
Historias de Usuario 1	+ añadir tarea Introducción de pedido (cliente preferente)	+ añadir tarea Crear Reporte de Clientes Frecuentes por Semana  Crear Reporte Línea de Producto más vendido por Semana	+ añadir tarea Procedimientos Almacenados para Consultar Datos  Elaborar Cubos y Datamarts para los Reportes	+ añadir tarea Validar Data Warehouse y Data Marts  Caja Negra y Blanca de los Procedimientos Almacenados	+ añadir tarea Reporte de Clientes Frecuentes por Semana  Reporte de Productos Más Vendidos por Semana
	Historias de Usuario 2	+ añadir tarea Planificación y Presupuesto del Producto  Crear una interfaz donde presupuestar los Costos del producto	+ añadir tarea Crear interfaz para la Planificación del Producto  Creación de Procedimientos Almacenados para los Presupuestos	+ añadir tarea Crear tablas y estructura de la BD  Creación de plataforma gráfica	+ añadir tarea Validar Integridad de la BD  Validar funcionalidad, integridad y concurrencia de la Plataforma  Caja Negra y Caja Blanca de los Procedimientos Almacenados

**Figura 10.** Tablero Scrum

Fuente: Elaboración Propia.

**Supervisión Programación XP;** La supervisión del desarrollo del Software a nivel de código se realizará bajo el enfoque XP, ya que este se adapta muy bien al esquema de trabajo que se tiene en la empresa, los programadores desarrollan el código para las H.U del Sprint, luego de eso realizan las pruebas unitarias para evitar cualquier tipo de error o fallas, además existe un filtro final por parte del Programador Senior que es el que supervisa el avance del código y la calidad que este tenga al final del Sprint ya que es este el que valida el resultado final y se hace responsable del trabajo realizado por los programadores en todo el sprint, el método de trabajo son reuniones diarias y constante comunicación directa.



**Figura 11.**Supervisión Programación XP

Fuente: Elaboración Propia.

**Retrospectiva del Sprint;** La retrospectiva del sprint se realiza básicamente para analizar los problemas que se han presentado durante el sprint y poder tomar acciones correctivas. Se muestra el resultado obtenido en la retrospectiva del sprint.

Problemas	Causas 1 / 1	Acciones
+ añadir tarea	+ añadir tarea	+ añadir tarea
Al implementar las historias de usuario, el alcance de algunas iba creciendo en medio del desarrollo.	Antes de iniciar la implementación de las HU, no se establecieron las tareas necesarias para su desarrollo.	A toda historia de usuario se le debe detallar con las tareas que se deben realizar para su cumplimiento.

**Figura 12.**Retrospectiva Sprint

Fuente: Elaboración Propia

## Monitorear y Controlar el Proyecto de Desarrollo de Software

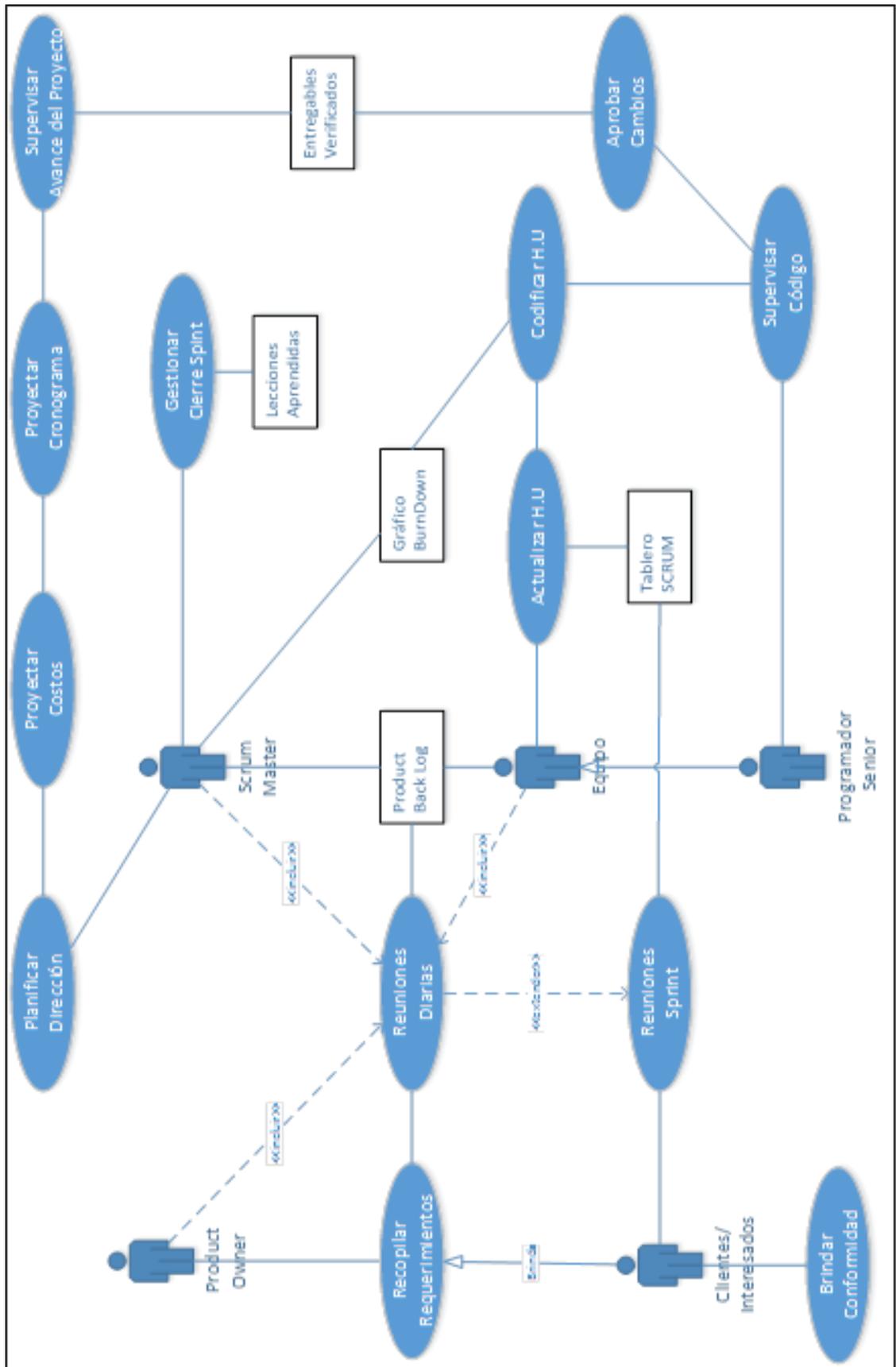
*Tabla 14. Dirigir el Proyecto de Desarrollo de Software*

Atributo	
<b>Participantes</b>	Product Owner Scrum Master Equipo Interesados.
<b>Responsables</b>	Product Owner Scrum Master
<b>Herramientas y Técnicas</b>	Gráfico burndown. Tablero de Scrum. Juicio de Expertos para interpretar la información. Sistema o Herramientas para Dirección de proyectos. Reuniones.
<b>Entradas</b>	Product Back log. Sprint Back log. Planes para la dirección de proyectos. Proyectar el cronograma para conocer las desviaciones de las fechas planificadas y las fechas reales. Proyectar los costos respecto de los gastos planificados y los reales. Cambios aprobados con validaciones. Información del desempeño de avance en puntos de la historia de usuario. Lecciones aprendidas
<b>Proceso</b>	<ol style="list-style-type: none"> <li>1. El equipo actualiza a diario la información de cada historia de usuario cómo va el estado de la historia, el porcentaje de avance y los puntos de historia terminados.</li> <li>2. Se registran los cambios validados como nuevas historias de usuario o corrección de errores con su estado en el Product Back log y se analiza y registra como afecta a la línea base de los Costos y Tiempos.</li> <li>3. Realizar un seguimiento del cronograma se usa el BurnDown y se puede utilizar un gráfico en el Project.</li> <li>4. Llevar el registro y control de los costos autorizados y los costos incurridos hasta la fecha que van de la mano del avance de las historias y los entregables completados (CPI).</li> <li>5. La calidad del software es controlada por los desarrolladores en los test unitarios y la supervisión del programador Senior (XP), luego se ejecutan las pruebas definidas y también cuando los usuarios funcionales realizan la certificación de las historias basadas en los criterios de aceptación.</li> </ol>

	<ol style="list-style-type: none"> <li>6. El Scrum Master realiza un seguimiento y control del avance del Sprint mediante el gráfico burndown y consolida la información del avance de cada Sprint y de la velocidad del equipo.</li> <li>7. El equipo de trabajo utiliza el tablero para ir registrando las novedades y avances del Sprint esto sirve como medio de seguimiento y comunicación para que la información llegue al equipo y a los interesados del proyecto.</li> <li>8. Las reuniones del Sprint permiten al equipo y a los interesados realizar el seguimiento y control referente a desempeño del proyecto, incidentes presentados, establecer necesidades futuras y aclarar dudas.</li> <li>9. Las reuniones del Sprint se tratan de forma diaria ya que se establecen los riesgos, esto permitirá controlar e identificar nuevos riesgos potenciales que afecten al proyecto o Sprint, donde se tratan las acciones a tomar para hacer frente a los riesgos.</li> </ol>
<b>Salidas</b>	<ul style="list-style-type: none"> <li>• Tablero Scrum.</li> <li>• Gráfico burndown.</li> <li>• Informe del desempeño del equipo, como el registro de la velocidad de cada Sprint del proyecto según.</li> <li>• Registro del avance del Proyecto.</li> <li>• Actualización de los Sprint, historias de Usuarios, Costos y Tiempo.</li> <li>• Entregables verificados.</li> <li>• Retroalimentación de las Lecciones aprendidas.</li> </ul>

**Nota.** Fuente: Elaboración Propia del Autor.

La metodología planteada en esta Tesis está preparada para aplicarse en cualquier proyecto de Software, el paso a paso presentado en la tabla anterior es una guía que orienta a los responsables del proyecto, sin embargo la gestión que se realice dentro del mismo es completo manejo personal del profesional a cargo, el proceso general de la metodología se presenta en la Figura 13 elaborada por el autor, esta refleja todo el trabajo realizado por el equipo hasta el término del Sprint y el cierre del Proyecto.



**Figura 13.**Diagrama Dirección del Proyecto

Fuente: Elaboración Propia