



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

**Selección de un Método Ágil para el desarrollo
de Sistemas Big Data**

presentada por

Lic. Antonio Juan Capistran Abundez

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis

Dr. Juan Carlos Rojas Pérez

Cuernavaca, Morelos, México. Julio de 2020.



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., **25/junio/2020**

OFICIO No. DCC/099/2020
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFCIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Antonio Juan Capistrán Abundez**, con número de control M18CE001, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"Selección de un Método Ágil para el Desarrollo de Sistemas Big Data"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Juan Carlos Rojas Pérez
Doctor en Ciencias en Ciencias de la
Computación
6099372
Director de tesis

Dra. Olivia Graciela Frago Díaz
Doctora en Ciencias en Ciencias de la
Computación
7420199
Co-directora de tesis

Dr. René Santaolaya Salgado
Doctor en Ciencias de la Computación
4454821
Revisor 1

M.C. Mario Guillén Rodríguez
Maestro en Ciencias con Especialidad en
Sistemas Computacionales
7573768
Revisor 2

C.c.p. Depto. Servicios Escolares
Expediente / Estudiante
JGGS/lmz





"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos **01/julio/2020**

OFICIO No. SAC/ 205/2020

Asunto: Autorización de impresión de tesis

ANTONIO JUAN CAPISTRÁN ABUNDEZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "*Selección de un Método Ágil para el Desarrollo de Sistemas Big Data*", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

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

ATENTAMENTE

Excelencia en Educación Tecnológica
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



**CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA**

C.c.p. M.E. Guadalupe Garrido Rivera, Jefa del Departamento de Servicios Escolares
Expediente
GVGR/CHC



Dedicatoria

“Las verdades que revela la inteligencia permanecen estériles. Sólo el corazón es capaz de fecundar los sueños” (Anatole France, 1844 - 1924)

A la vida, por la enseñanza del día, por la fe y esperanza de la humanidad.

Para mi madre y mi padre por su apoyo incondicional durante la realización de este trabajo. Por proporcionarme amor y educación a lo largo de mi vida.

A mi familia con quienes he compartido gratas experiencias.

A todos los que cada día se levantan para edificar el futuro de este país.

A los que ya no están, y que seguimos recordando con todo el corazón.

A todos ellos, con mucho cariño.

Para mis amigos de generación **Julia, Orlando, Vicente y Valentino** que me acompañaron durante este proceso de formación académica y profesional.

Para ustedes con todo mi cariño

Agradecimientos

Al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por otorgarme el sustento económico para realizar este trabajo de investigación.

Al **Tecnológico Nacional de México (TecNM) / Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)**, a su personal docente y administrativo por brindarme la oportunidad de realizar mis estudios de maestría en esta destacada institución.

A mi director de tesis, **Dr. Juan Carlos Rojas Pérez**, por su tiempo, apoyo, motivación, paciencia y conocimientos para la realización de este trabajo. Gracias por su confianza y por aportar en gran medida mi formación profesional.

A mi comité revisor conformado por: **Dra. Olivia Graciela Fragoso Díaz, Dr. René Santaolaya Salgado, M.C. Mario Guillen Rodríguez**, por su tiempo y atención brindados en el desarrollo de este trabajo para engrandecer el contenido a través de sus valiosas aportaciones.

Al **Cuerpo Académico de Ingeniería de Software (CAIS)** por sus enseñanzas transmitidas en cada una de las clases y asesorías que me brindaron.

A **mi familia** por todo su apoyo, confianza y amor brindado para seguir adelante y concluir con éxito esta etapa de mi vida.

A todos los amigos y compañeros que conocí en el CENIDET, a **Dulce Maria y Socrates** por su apoyo y motivación para estudiar la maestría, a los de mi generación por llevar esta etapa de formación conmigo, así como compartir gratos y alegres momentos: **Julia, Orlando, Vicente y Valentino**.

A todos por ser parte de este importante logro, ¡gracias totales!

Resumen

Nuestros tiempos destacan por la rápida evolución de la informática, áreas emergentes como el aprendizaje automático a través de la Inteligencia Artificial, el Internet de las Cosas con múltiples dispositivos conectados a la red y plataformas digitales en diversas áreas de conocimiento genera una ascendente cantidad de usuarios conectados a dispositivos y sensores electrónicos, los cuales, generan decenas de datos medibles en terabytes (TB), petabytes (PB) y Exabytes (EB), ocasionando que empresas de distinto giro económico modifiquen la forma de tomar decisiones en sus procesos de negocios (John Gantz, 2013).

El origen de grandes cantidades de datos va desde transacciones bancarias, pagos con tarjeta, señales GPS o búsquedas en Google hasta datos generados de forma voluntaria en teléfonos inteligentes, al publicar información en blogs, Youtube o mensajes en redes sociales como Twitter y Facebook. Una de las áreas generadoras de datos en constante crecimiento, son las redes sociales y el contenido generado por los usuarios (User Generated Content UGC) en Internet a través del cual se puede conocer las sensaciones de usuarios y clientes de diversas áreas económicas (Marine-Roig & Anton Clavé, 2015).

Es imperante que los datos generados por empresas y usuarios conectados a internet sean analizados en busca de información pertinente para visualizar y prever posibles comportamientos de los usuarios que direccionen las tendencias del mercado. Es así como nacen los sistemas Big Data, los cuales, debido a sus características de volumen, velocidad y variedad de los datos sobrepasan los recursos de software y hardware actualmente disponibles. Por lo tanto, los proyectos de desarrollo de software para aplicaciones de análisis de Big Data (BDAA) requieren enfoques innovadores para ayudar a administrarlos y usarlos de manera efectiva (Al-Jaroodi, Hollein, & Mohamed, 2017).

El rápido aumento en la cantidad de datos no estructurados y estructuras de datos complejas, deja obsoletos los métodos de desarrollo de software tradicionales ya que su funcionamiento es secuencial, por lo tanto, no son efectivos para resolver la compleja combinación de 3v's en Big Data. En el presente documento se revisan metodologías ágiles que permitan presentar una guía general asequible para desarrollar sistemas Big Data. El enfoque ágil con características de iteraciones cortas y adaptación al cambio, facilitan que empresas de diversas áreas económicas, disminuyan tiempos de comercialización de productos, automatización de procesos y entregas de productos funcionales que cumplan satisfactoriamente los requisitos del cliente(Sharma, 2012).

Palabras clave: Big Data, Metodologías Ágiles

Abstract

Our times stand out for the rapid evolution of computer science, emerging areas such as machine learning through Artificial Intelligence, the Internet of Things with multiple devices connected to the network and digital platforms in various areas of knowledge generates an ascending number of users connected to electronic devices and sensors, which generate tens of measurable data in terabytes (TB), petabytes (PB) and Exabytes (EB), causing companies of different economic activity to modify the way of making decisions in their business processes (John Gantz, 2013).

The origin of large amounts of data ranges from bank transactions, card payments, GPS signals or Google searches to data generated voluntarily on smartphones, by posting information on blogs, YouTube or messages on social networks such as Twitter and Facebook. One of the areas that generate data in constant growth are social networks and user-generated content (User Generated Content UGC) on the Internet through which you can know the feelings of users and customers from various economic areas (Marine-Roig & Anton Clavé, 2015).

It is imperative that the data generated by companies and users connected to the Internet be analyzed for relevant information, to visualize and anticipate possible user behaviors, which address market trends. This is how Big Data systems are born, which, due to their characteristics of volume, speed and variety of data, exceed the currently available software and hardware resources. Therefore, software development projects for Big Data analysis applications (BDAA) require new and innovative approaches to help manage and use them effectively (Al-Jaroodi & Mohamed, 2017).

The rapid increase in the amount of unstructured data and complex data structures makes traditional working methods or software development obsolete since operation is sequential, therefore, they are not effective in solving the complex combination of 3v's in Big Data. This document reviews agile methodologies that allow presenting an affordable general guide for developing Big Data systems. The agile approach with characteristics of short iterations and adaptation to change, make it easier for companies from various economic areas to reduce product marketing times, process automation and deliveries of functional products that satisfactorily meet customer requirements(Sharma, 2012).

Keywords: Big Data, Agile Methodologies

Índice de Contenido

Capítulo 1 Introducción	1
1.1 <i>Presentación</i>	2
1.2 <i>Planteamiento del problema</i>	3
1.3 <i>Justificación</i>	4
1.4 <i>Objetivos</i>	4
1.4.1 <i>Objetivo general</i>	4
1.4.2 <i>Objetivo específico</i>	4
1.5 <i>Alcances y limitaciones</i>	4
1.5.1 <i>Alcances</i>	4
1.5.2 <i>Limitaciones</i>	4
1.6 <i>Beneficios</i>	4
1.7 <i>Complejidad de la solución</i>	5
1.8 <i>Organización del Documento</i>	5
Capítulo 2 Marco Teórico	6
2.1 <i>Big Data</i>	7
2.1.1 <i>Características de Big Data</i>	7
• <i>Volumen</i>	7
• <i>Velocidad</i>	8
• <i>Variedad</i>	9
2.2 <i>Metodologías Ágiles</i>	9
Capítulo 3 Estudio de trabajos relacionados	10
3.1 <i>Big Data</i>	11
Big Data Ecosystem – Review on Architectural Evolution (Singh, Behera, & Mantri, 2019)	11
Big Data: A Systematic Review (Antonio Fernando Cruz Santos & Ota´vio Manoel Pereira Siqueira, 2018)	11
Hadoop: Solution to Unstructured Data Handling (Madaan, Sharma, Pahwa, Das, & Sharma, 2018)	11
Big Data Challenges and Solutions (Srinivasu, Koushik, & Santhosh, 2017)	11
3.2 <i>Enfoques Tradicionales de Desarrollo de Software</i>	15
A Study of Software Development Life Cycle Process Models (Ali, 2017)	15
Proyectos de desarrollo de software (Tapias, 2014)	15
3.3 <i>Enfoque ágil</i>	17
Agile Methodologies: Comparative Study and Future Direction (Ahmad, Soomro, & Broh, 2014)	17
Empirical Study of Agile Software Development Methodologies: A Comparative Analysis (Matharu et al., 2015)	17
Agile Software Development Methodologies: Survey of Surveys (Al-Zewairi et al., 2017)	18
Agile Methodologies in Education: A Review (Salza, Musmarra, & Ferrucci, 2019)	18
Agile Methodologies: Comparative Study and Future Direction (Ahmad et al., 2014)	21

Empirical Study of Agile Software Development Methodologies: A Comparative Analysis (Matharu et al., 2015).....	21
Agile Software Development Methodologies: Survey of Surveys (Al-Zewairi et al., 2017).....	21
Agile Methodologies in Education: A Review (Salza et al., 2019).....	21
3.4 Enfoque tradicional versus Enfoque ágil	22
Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software (Montero, Cevallos, & Cuesta, 2018).....	23
Challenges of Traditional and Agile Software Processes (Sánchez, 2018).....	23
A Comparison between Agile and Traditional Software Development Methodologies (Awad, 2019). ...	24
Analysis of Software Development Methodologies (SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, 2019). 24	
3.5 Metodologías Ágiles.....	26
3.5.1 Beneficios	27
3.5.2 Lean.....	30
3.5.3 Kanban.....	30
3.5.4 Extreme Programming.....	31
3.5.5 Scrum.....	32
Capítulo 4 Metodología de solución.....	34
4.1 Metodología de solución.....	35
4.3 Caso de estudio.....	35
4.4 Mapeo del caso de estudio con Scrum	36
4.4.1 Marco Scrum.....	36
4.4.2 Mapeo Big Data-Scrum.....	37
4.4.3 Implementación.....	37
4.4.4 Procesos Scrum	38
4.4.4.1 Proceso 1: Inicio	38
4.4.4.2 Proceso 2: Planeación y estimación.....	58
4.4.4.3 Proceso 3: Implementación.....	70
4.4.4.4 Proceso 4: Revisión y retrospectiva.....	74
4.4.4.5 Proceso 5: Lanzamiento.....	76
Capítulo 5. Conclusiones y Trabajo futuro.....	77
5.1 Conclusiones.....	78
5.2 Trabajos futuros.....	79
Referencias	79
Anexo A: Metodología Lean.....	82
Lean	83
Beneficios (Prodintec, 2010).....	84
Principios (Prodintec, 2010)	84
Anexo B: Metodología Kanban.....	87
Kanban.....	88
Características (Matharu et al., 2015).....	88
Reglas (Pérez, 2012).....	88
Roles.....	89

Anexo C: Metodología Extreme Programming	90
<i>Metodología Extreme Programming (Wells, 2009)</i>	<i>91</i>
Fases (Pérez, 2012)	91
Reglas (Wells, 2009).....	92
Valores (Wells, 2009)	95
Roles (Pérez, 2012)	96
Anexo D: Metodología Scrum	98
<i>Scrum (SCRUMstudy, 2017).....</i>	<i>99</i>
Ventajas de Scrum	99
Scrum versus Gestión tradicional de proyectos.....	99
Principios de Scrum.....	100
Aspectos de Scrum.....	109
Procesos de Scrum.....	125
Inicio.....	126
Planificación y estimación	127
Implementación.....	129
Revisión y retrospectiva	130
Lanzamiento	130
Procesos adicionales.....	131
Scrum para grandes proyectos	132
Scrum para la empresa.....	132
Anexo E: Análisis de contenido generado por el usuario	134
<i>Fases de sistemas Big Data.</i>	<i>135</i>
Etapa 1. Selección de alojamiento web.....	136
Etapa 2. Recopilación de datos.....	137
Etapa 3. Pre procesamiento de datos.	137
Etapa 4. Análisis de contenido.....	139
Anexo F: Mapeo Big Data-Metodologías Ágiles.....	141
<i>Mapeo Big Data y Metodologías Ágiles.....</i>	<i>142</i>
Anexo G: Documentos	145

Índice de Figuras

FIGURA 1 BIG DATA (CUBANA, 2017)	7
FIGURA 2 VOLUMEN EN BIG DATA (LADRERO, 2018)	8
FIGURA 3 VELOCIDAD EN BIG DATA (LADRERO, 2018)	8
FIGURA 4 VARIEDAD EN BIG DATA (LADRERO, 2018)	9
FIGURE 5 CICLO ÁGIL (LEON, 2017)	9
FIGURE 6 ASPECTOS DE DESARROLLO DE SOFTWARE ÁGIL (ABRAHAMSSON ET AL., 2002)	19
FIGURA 7 CICLOS DE ENFOQUES TRADICIONAL Y ÁGIL	23
FIGURE 8 BENEFICIOS ÁGILES (MATHARU ET AL., 2015)	27
FIGURE 9 LEAN, MEJORAR REDUCIENDO (GALGANO, 2008)	30
FIGURA 10 EJEMPLO DE TABLERO KANBAN (PÉREZ, 2012)	31
FIGURA 11 FLUJO XP (WELLS, 2009)	31
FIGURA 12 CICLO SCRUM (MATHARU ET AL., 2015)	32
FIGURA 13 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA DE SOLUCIÓN	35
FIGURE 14 FLUJO DE PROYECTO SCRUM (SCRUMSTUDY, 2017)	36
FIGURA 15 CRONOGRAMA SPRINT 0	37
FIGURA 16 LEAN: MEJORAR REDUCIENDO (GALGANO, 2008)	83
FIGURE 17 FILOSOFÍA LEAN (PRODINTEC, 2010)	84
FIGURA 18 MUDA O DESPERDICIO (SOLUTIONS, 2019)	85
FIGURA 19 MURA O IRREGULARIDAD (SOLUTIONS, 2019)	85
FIGURA 20 MURI O SOBRECARGA (SOLUTIONS, 2019)	86
FIGURA 21 FLUJO CONTINUO (PRODINTEC, 2010)	86
FIGURA 22 TABLERO KANBAN (PÉREZ, 2012)	89
FIGURA 23 FLUJO DE XP (WELLS, 2009)	91
FIGURA 24 METODOLOGÍA SCRUM (MATHARU ET AL., 2015)	99
FIGURE 25 PRINCIPIOS DE SCRUM (SCRUMSTUDY, 2017)	101
FIGURA 26 TRANSPARENCIA EN SCRUM (SCRUMSTUDY, 2017)	101
FIGURA 27 INSPECCIÓN EN SCRUM (SCRUMSTUDY, 2017)	102
FIGURA 28 ADAPTACIÓN EN SCRUM (SCRUMSTUDY, 2017)	103
FIGURE 29 OBJETIVOS DE EQUIPO AUTO ORGANIZADO (SCRUMSTUDY, 2017)	104
FIGURA 30 COLABORACIÓN EN PROYECTOS SCRUM (SCRUMSTUDY, 2017)	105
FIGURA 31 PRIORIZACIÓN BASADA EN EL VALOR (SCRUMSTUDY, 2017)	106
FIGURA 32 BLOQUES DE TIEMPO PARA REUNIONES DE SCRUM (SCRUMSTUDY, 2017)	107
FIGURA 33 ITERACIONES EN MODELO TRADICIONAL (SCRUMSTUDY, 2017)	108
FIGURA 34 ITERACIONES EN MODELO SCRUM (SCRUMSTUDY, 2017)	109
FIGURA 35 ASPECTOS DE SCRUM (SCRUMSTUDY, 2017)	109
FIGURA 36 INTERACCIÓN DE ROLES CENTRALES DE SCRUM (SCRUMSTUDY, 2017)	114
FIGURA 37 BACKLOG PRIORIZADO DEL PRODUCTO (SCRUMSTUDY, 2017)	119
FIGURA 38 CRITERIOS DE ACEPTACIÓN (SCRUMSTUDY, 2017)	119
FIGURA 39 CICLO PDCA EN SCRUM (SCRUMSTUDY, 2017)	121
FIGURA 40 PROCESO DE APROBACIÓN DE CAMBIOS (SCRUMSTUDY, 2017)	122
FIGURA 41 ACTUALIZACIÓN DE BACK LOG (SCRUMSTUDY, 2017)	122
FIGURE 42 PROCESO DE PRIORIZACIÓN DE RIESGOS (SCRUMSTUDY, 2017)	125
FIGURA 43 FLUJO DE DATOS DE LA FASE DE INICIO (SCRUMSTUDY, 2017)	127
FIGURA 44 FLUJO DE DATOS DE LA FASE DE PLANIFICACIÓN Y ESTIMACIÓN (SCRUMSTUDY, 2017)	128
FIGURA 45 FLUJO DE DATOS DE LA FASE DE IMPLEMENTACIÓN (SCRUMSTUDY, 2017)	129
FIGURA 46 FLUJO DE DATOS DE LA FASE DE REVISIÓN Y RETROSPECTIVA (SCRUMSTUDY, 2017)	130
FIGURA 47 FLUJO DE DATOS DE LA FASE DE LANZAMIENTO (SCRUMSTUDY, 2017)	131
FIGURA 48 FASES GENERALES DE BIG DATA (FERNÁNDEZ-MANZANO, NEIRA, & CLARES-GAVILÁN, 2016)	135
FIGURA 49 ETAPAS DE PROCESAMIENTO UGC (MARINE-ROIG & ANTON CLAVÉ, 2015)	136
FIGURA 50 EMPRESAS EN MÉXICO APLICANDO SCRUM (SCRUMSTUDY, 2017)	144

Índice de Tablas

TABLA 1 CARACTERÍSTICAS BIG DATA (TECHAMERICA & FOUNDATION, 2012)	12
TABLA 2 ARTÍCULOS REFERENTES A BIG DATA.....	13
TABLA 3 VENTAJAS Y DESVENTAJAS DEL MODELO EN CASCADA	16
TABLA 4 LISTA DE ESTADO DEL ARTE DE ARTÍCULOS REFERENTES AL ENFOQUE ÁGIL.	21
TABLA 5 ESTUDIO COMPARATIVO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE AGILES Y TRADICIONALES (MATHARU ET AL., 2015).	24
TABLA 6 METODOLOGÍAS COMPATIBLES SOBRE TAMAÑOS DE PROYECTOS (SOOBIA SAEED, NZ JHANJHI, MEHMOOD NAQVI, 2019).	25
TABLA 7 MAPEO SCRUM-BIG DATA.....	37
TABLA 8 SCRUM VS GESTIÓN TRADICIONAL (SCRUMSTUDY, 2017)	100
TABLA 9 RESPONSABILIDADES DEL PRODUCT OWNER EN LOS PROCESOS DE SCRUM (SCRUMSTUDY, 2017).....	110
TABLA 10 RESPONSABILIDADES DEL SCRUM MASTER EN LOS PROCESOS DE SCRUM (SCRUMSTUDY, 2017)	111
TABLA 11 RESPONSABILIDADES DEL EQUIPO SCRUM EN LOS PROCESOS DE SCRUM (SCRUMSTUDY, 2017).....	112
TABLA 12 ROLES CENTRALES DE SCRUM (SCRUMSTUDY, 2017)	114
TABLA 13 VALOR BASADO EN PROYECTOS DE SCRUM (SCRUMSTUDY, 2017)	117
TABLA 14 RESPONSABILIDADES EN LA JUSTIFICACIÓN DE NEGOCIO (SCRUMSTUDY, 2017)	117
TABLA 15 CRITERIOS DE TERMINADO EN CASCADA (SCRUMSTUDY, 2017)	120
TABLA 16 PROCESO DE SCRUM INICIO (SCRUMSTUDY, 2017)	126
TABLA 17 PROCESO DE SCRUM PLANIFICACIÓN Y ESTIMACIÓN (SCRUMSTUDY, 2017).....	127
TABLA 18 PROCESO DE SCRUM IMPLEMENTACIÓN (SCRUMSTUDY, 2017)	129
TABLA 19 PROCESO DE SCRUM REVISIÓN Y RETROSPECTIVA (SCRUMSTUDY, 2017)	130
TABLA 20 PROCESO DE SCRUM LANZAMIENTO (SCRUMSTUDY, 2017)	130
TABLA 21 SCRUM PARA GRANDES PROYECTOS (SCRUMSTUDY, 2017)	132
TABLA 22 SCRUM PARA LA EMPRESA (SCRUMSTUDY, 2017)	132
TABLA 23 PUNTO DE VISTA: USO (IACOVELLI & SOUVEYET, 2008)	142
TABLA 24 PUNTO DE VISTA: AGILIDAD (IACOVELLI & SOUVEYET, 2008)	142
TABLA 25 PUNTO DE VISTA: APLICABILIDAD (IACOVELLI & SOUVEYET, 2008).....	143
TABLA 26 PUNTO DE VISTA: PRODUCTOS (IACOVELLI & SOUVEYET, 2008).....	143

Índice de Gráficas

GRÁFICA 1 RETOS DE BIG DATA	14
GRÁFICA 2 ETAPAS DEL DESARROLLO DE SOFTWARE	16
GRAFICA 3 ENFOQUE ÁGIL	22
GRÁFICA 4 ASPECTOS DE BENEFICIOS (VERSION ONE, 2019).	28
GRÁFICA 5 ÁREAS DE IMPLEMENTACIÓN (VERSION ONE, 2019)	28
GRÁFICA 6 RAZONES COMERCIALES PARA ADOPTAR AGILE (VERSION ONE, 2019).....	29
GRAFICA 7 METODOLOGÍAS ÁGILES MÁS UTILIZADAS (VERSION ONE, 2019).....	29
GRAFICA 8 CARACTERÍSTICAS ÁGILES	144

Capítulo 1 Introducción

En este capítulo se describe el planteamiento del problema que atiende esta investigación, la justificación, el objetivo general y sus objetivos específicos, los alcances y las limitaciones, los beneficios y la organización del documento de tesis.

1.1 Presentación

La Informática, a través del tiempo, ha logrado filtrarse en la mayoría de las actividades realizadas por el ser humano, tanto en el ámbito económico como social, principalmente por la creciente cantidad de dispositivos conectados a Internet. Dichas actividades pueden ser medibles de forma digital, generar enormes cantidades de datos factibles ser analizados para obtener información pertinente, que permita a las empresas prestadoras de servicios o vendedoras de productos, mejorar la calidad de sus procesos de negocio para predecir tendencias de mercado logrando anticiparse a gustos y necesidades de sus clientes.

El origen del termino Big Data, es consecuencia de la enorme cantidad de datos generados diariamente por nosotros como usuarios de diversos dispositivos electrónicos conectados a Internet, los cuales, a través de cookies, registran nuestras actividades en servicios digitales. Los datos generados diariamente, han provocado que en la actualidad el análisis de datos sea una actividad imperativa de diversas organizaciones. Sin embargo, registrar y analizar datos en busca de información pertinente, se ha convertido en una actividad problemática debido a la compleja combinación de volumen, velocidad y variedad que caracteriza a los sistemas Big Data (TechAmerica & Foundation, 2012).

Disminuir la complejidad de análisis en sistemas Big Data, ha sido complicado para las empresas que durante años utilizaron métodos tradicionales de desarrollo de software. La capacidad para realizar iteraciones cortas en tiempo, realizar cambios de requisitos en diversos tipos y tamaños de proyectos, permite a las metodologías ágiles ser una alternativa de apoyo actual. Las características ágiles, incrementan su popularidad en múltiples áreas de negocio vigentes. La agilidad tiene como constante la mejora continua, incrementando la calidad suficiente de productos y servicios para satisfacción de los clientes (Matharu, Mishra, Singh, & Upadhyay, 2015).

En este trabajo de investigación, se realiza un mapeo entre las características generales de los sistemas Big Data y las Metodologías Ágiles, para encontrar cual enfoque puede adaptarse como guía general para desarrollar sistemas Big Data. Se revisan las formas de implementación de los cuatro métodos ágiles más populares en la actualidad, así como los beneficios que han logrado diversas empresas una vez que decidieron migrar sus procesos de negocio hacia un método ágil. Con esta investigación se pretende dejar una guía general de cómo implementar un método ágil en un sistema Big Data.

1.2 Planteamiento del problema

La rápida generación de información digital, ha llevado a la sociedad a incluir el concepto Big Data, tanto en el lenguaje común de las personas como en el vocabulario de personas especializadas. Empresas económicas de productos o servicios que dependen de información obtenida de análisis de datos específicos, son quienes más se relacionan con sistemas Big Data para tomar decisiones pertinentes acordes a las tendencias del mercado.

Sin embargo, los sistemas Big Data son complejos debido a la combinación de volumen, velocidad y variedad de datos. Tal combinación, dificulta los procesos de análisis para obtener información, principalmente por que los datos que se depuran, además de ser grandes cantidades, se generan casi en tiempo real y con variedad de formato. Los enfoques tradicionales para análisis y desarrollo de software quedan obsoletos ante la rapidez de cambios en el desarrollo de sistemas Big Data.

Los métodos tradicionales tales como los modelos en cascada o secuenciales trabajan con formatos rígidos para realizar cambios en alguna de las etapas propuestas del desarrollo de software, estos métodos no son capaces de detectar problemas cuando se inicia el proyecto, por lo tanto, no son modelos adecuados para desarrollar y enfrentar las características de los sistemas Big Data (SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, 2019).

El problema consiste en que no se cuenta con una metodología para sistemas Big Data que atienda a las características de desarrollo de este tipo de sistemas. Pérdida de tiempo efectivo, atraso en fechas de entrega y trabajo repetitivo, son consecuencia de una equivocada planeación y levantamiento de requisitos. En tiempos presentes, para desarrollar sistemas Big Data, las empresas requieren enfoques ágiles híbridos acordes a las circunstancias de sus procesos de negocio. Las metodologías ágiles más utilizadas por diversas corporaciones son: Scrum y Extreme Programming (Version One, 2019).

1.3 Justificación

Los sistemas Big Data actuales caracterizados con volumen, velocidad y variedad de datos, desbordan la capacidad tecnológica de hardware y software vigente. Atender y procesar datos reconocidos como Big Data casi en tiempo real, es prioritario en empresas que requieren de información efectiva para la toma de decisiones. Se requiere de herramientas o entornos de trabajo que funcionen como guía de apoyo para desarrollar sistemas Big Data. Para solventar las desventajas de los enfoques tradicionales, se propone a las metodologías ágiles. Específicamente, la propuesta de este trabajo es Scrum como marco de trabajo para desarrollar sistemas Big Data.

1.4 Objetivos

1.4.1 Objetivo general

- Seleccionar de entre un conjunto de metodologías ágiles, una metodología o combinación de metodologías que se puedan adaptar a las características cambiantes de los sistemas Big Data.

1.4.2 Objetivo específico

- Mapear la metodología ágil seleccionada al desarrollo de un sistema Big Data.

1.5 Alcances y limitaciones

1.5.1 Alcances

- Investigar al menos 4 métodos ágiles para el desarrollo de sistemas Big data.
- Proponer una guía con enfoque ágil para desarrollar sistemas Big Data.

1.5.2 Limitaciones

- La guía con enfoque ágil no abarca el desarrollo de todos los tipos de sistemas Big Data.

1.6 Beneficios

Disponer de una guía que permita al usuario seleccionar un marco de trabajo o metodología ágil adecuada que pueda aplicarse en el desarrollo de sistemas Big Data.

1.7 Complejidad de la solución

La complejidad radica en la búsqueda y análisis de literatura relacionada a Big Data y Metodologías Ágiles la cual nos permita proponer una guía con enfoque ágil adecuada para atender la complejidad característica de los sistemas Big Data.

1.8 Organización del Documento

La organización del presente documento y la distribución de sus capítulos, así como una breve descripción de los mismos, son presentados a continuación:

Capítulo 1 Introducción

Se describe el planteamiento del problema que atiende esta investigación, la justificación, el objetivo general y sus objetivos específicos, los alcances y las limitaciones, los beneficios y la organización del documento de tesis.

Capítulo 2 Marco Teórico

Se define los conceptos de Big Data y Metodologías Ágiles como aspectos teóricos pertinentes para comprender el contexto de esta investigación.

Capítulo 3 Estudio de trabajos relacionados

Se describe la literatura que da soporte a este trabajo. Se presentan los trabajos relacionados con mayor semejanza al contexto de los desafíos inmersos en los sistemas Big Data, así como la evolución de las Metodologías Ágiles en diversas organizaciones.

Capítulo 4 Metodología de solución

Se describe porque se elige Scrum como la metodología ágil implementada para proyectos Big Data. Se realiza el análisis de un caso de estudio y se aplica Scrum al caso de estudio.

Capítulo 5 Conclusiones y Trabajo futuro

Se presentan las conclusiones de acuerdo al mapeo realizado entre las características de los sistemas Big Data y las Metodologías Ágiles además de un análisis de posibles trabajos futuros.

Capítulo 2 Marco Teórico

En este capítulo se definen los conceptos relacionados a Big Data y Metodologías Ágiles como aspectos teóricos pertinentes para comprender el contexto de esta investigación.

2.1 Big Data

Doug Laney define los sistemas Big Data como datos con alto volumen, velocidad y variedad. El volumen procede de redes sociales, sensores asociados con Internet de las cosas, transacciones comerciales y datos históricos. La velocidad refleja que los datos se generan en tiempo real. La variedad reconoce a diversas fuentes generadoras de datos con estructura tradicional, datos semiestructurados y datos no estructurados (Arndt, 2018), los cuales requieren técnicas y tecnologías avanzadas para permitir la captura, almacenamiento, distribución, gestión y análisis de la información (TechAmerica & Foundation, 2012). En la figura 1 se observa de forma metafórica el conocimiento actual que se tiene de Big Data.



Figura 1 Big Data (Cubana, 2017)

2.1.1 Características de Big Data

- **Volumen**

Se refiere esencialmente a las grandes cantidades de datos que se generan continuamente desde actividades realizadas en teléfonos inteligentes, comercio electrónico, sitios web y redes sociales (Mukherjee & Shaw, 2016). En la figura 2 se observa la escala de datos creados diariamente.

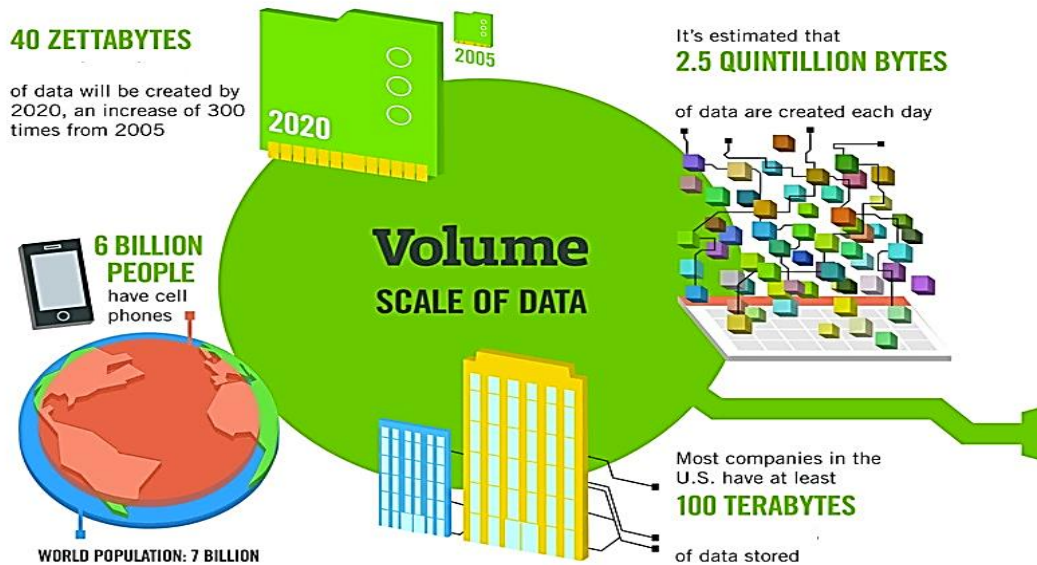


Figura 2 Volumen en Big Data (Ladrero, 2018)

- **Velocidad**

Es la rapidez a la que se generan cantidades colosales de datos. Facebook, por ejemplo: genera 2.700 millones de acciones similares por día y 300 millones de fotos, aproximadamente 2,5 millones de piezas de contenido cada día, mientras que Google Now procesa más de 1,2 billones de búsquedas por año en todo el mundo (Mukherjee & Shaw, 2016). En la figura 3 se observa la rapidez a la que se generan múltiples datos.

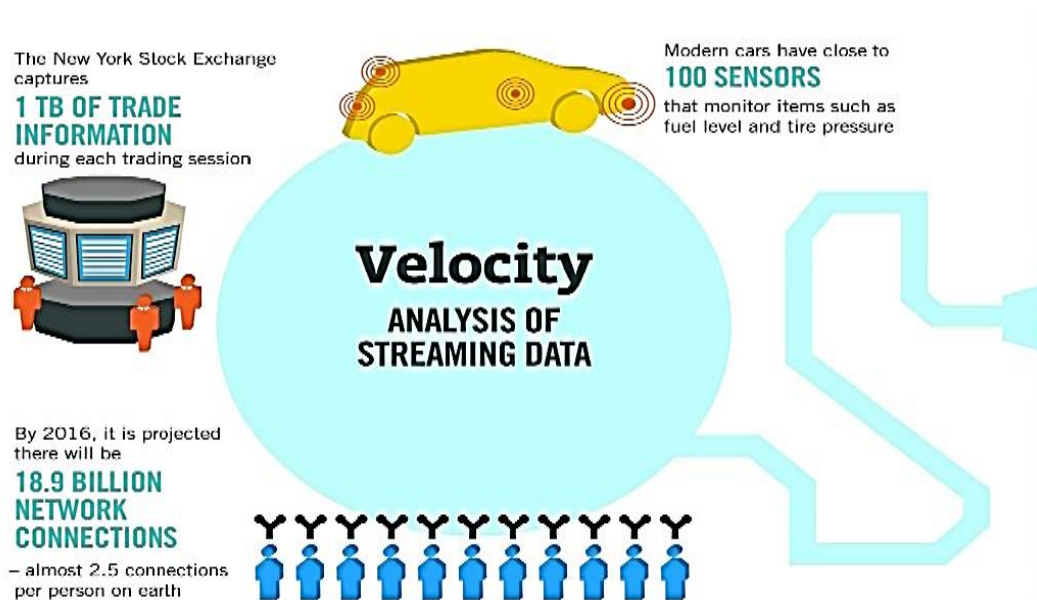


Figura 3 Velocidad en Big Data (Ladrero, 2018)

- **Variedad**

Datos generados de cualquier tipo de estructura, semiestructura o sin estructura. La forma convencional de datos son datos estructurados. Por ejemplo, texto. Los datos no estructurados pueden generarse a partir de sitios de redes sociales, sensores y satélites (Mukherjee & Shaw, 2016). En la figura 4 se observan diversas fuentes generadoras de datos tales como redes sociales, dispositivos y sensores electrónicos.

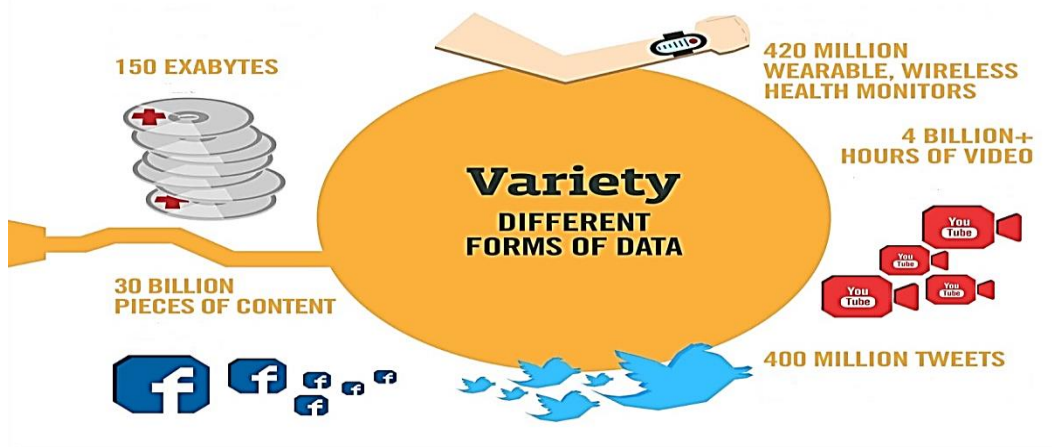


Figura 4 Variedad en Big Data (Ladrero, 2018)

2.2 Metodologías Ágiles

Son metodologías que intentan enfocarse en el objetivo principal del desarrollo efectivo de software, es decir, la creación de software en funcionamiento sin defectos (Franková, Drahošová, & Balco, 2016). Tiene como base la mejora iterativa como representación de un ciclo de vida de desarrollo de software autónomo a pequeña escala. A diferencia del enfoque tradicional, los métodos ágiles suponen simplicidad en todas las prácticas (Al-Zewairi, Biltawi, Etaawi, & Shaout, 2017). En la figura 5 se observa el ciclo de desarrollo en los enfoques ágiles.



Figure 5 Ciclo Ágil (Leon, 2017)

Capítulo 3 Estudio de trabajos relacionados

En este capítulo se describe el estudio literario en la que se fundamenta esta investigación. Se analiza lo que destacan los autores respecto a la observación de enfoques referentes a sistemas Big Data. Se revisa el análisis de diversos autores respecto a los enfoques tradicionales para desarrollar software y las características de enfoques ágiles. También se realiza una comparación entre características del enfoque tradicional y ágil. Finalmente, se describen metodologías ágiles investigadas para esta investigación.

3.1 Big Data

[Big Data Ecosystem – Review on Architectural Evolution \(Singh, Behera, & Mantri, 2019\).](#)

En este artículo, los autores revisan los desafíos de Big Data, identificando los principales problemas en las etapas de almacenamiento y procesamiento. Se destacaron varias herramientas, sistemas de archivos, administradores de recursos, programadores, motores de búsqueda, marcos SQL y NoSQL, además de marcos de monitoreo, para proporcionar al investigador la información necesaria para comprender Big Data.

Big data se describe como conjunto de datos complejos y masivos difíciles de administrar con herramientas de software tradicionales. Para que un proyecto Big Data tenga un rendimiento eficiente en el uso de los recursos, debe contar con las siguientes características: tolerancia a fallos, baja latencia, escalabilidad, extensibilidad, consultas a medida, mínimo mantenimiento y facilidad para ser depurado.

[Big Data: A Systematic Review \(Antonio Fernando Cruz Santos & Ota'vio Manoel Pereira Siqueira, 2018\).](#)

En este estudio, se efectúa una revisión sistemática de 466 publicaciones desde 2005 hasta marzo de 2016. Los autores presentan una visión amplia de Big Data referida como un conjunto de datos que no pueden ser percibidos, adquiridos, gestionados y procesados por las herramientas tradicionales de Tecnología de la Información (TI), dentro de un lapso de tiempo razonable.

[Hadoop: Solution to Unstructured Data Handling \(Madaan, Sharma, Pahwa, Das, & Sharma, 2018\).](#)

En este documento, los autores describen brevemente las características de Big Data en cuanto a volumen, velocidad y variedad. Se enfocan en el problema de volumen, debido a que los datos generados en el presente, ya no se miden en gigabytes, sino en petabytes y zettabytes, además de que alrededor de 80% de los datos generados, no están estructurados. También, se describe a la herramienta Hadoop y sus componentes, como ideales para resolver problemas de volumen de datos, específicamente si se trata de situaciones de agrupamiento u orientación. Se describen las características de MapReduce y YARN además de los beneficios de implementar Hadoop.

[Big Data Challenges and Solutions \(Srinivasu, Koushik, & Santhosh, 2017\).](#)

En este documento, los autores proporcionan una descripción general de efectos, influencia y desafíos de Big Data. Realiza un análisis de las tecnologías actuales para procesar sistemas Big Data. En el documento se concluye que, la enorme cantidad de datos que obtenemos de múltiples fuentes, pueden aprovecharse para tomar decisiones que permitan a las empresas ahorrar dinero o multiplicar sus ganancias.

Big Data es un concepto contemporáneo en la literatura científica y tecnológica, ya que muchas áreas de investigación se apoyan en análisis estadísticos obtenidos de la generación de enormes cantidades de datos. Big Data, internet de las cosas y realidad aumentada, destacan como principales tecnologías de la llamada industria 4.0 (Santos et al., 2017). La influencia de los datos se refleja en áreas tales como: salud, sector público administración, datos de ubicación personal y ventas en general. Además, es indudable que se agreguen áreas referentes a redes sociales o datos profesionales. El análisis de datos, permite detectar fraudes, tendencias sociales o manejo de riesgos (Zicari, n.d.).

Big Data describe grandes volúmenes de datos de alta velocidad, complejos y variables que requieren técnicas y tecnologías avanzadas para permitir la captura, almacenamiento, distribución, administración y análisis de la información. El aspecto de volumen aumenta por la multiplicación de fuentes de datos. La velocidad incrementa por las mejoras en la potencia informática que genera los datos y la variedad se diversifica desde datos genómicos, hasta datos generados por dispositivos móviles tanto estructurados como no estructurados (TechAmerica & Foundation, 2012). En la tabla 1 se observan la descripción de las características de Big Data.

Tabla 1 Características Big Data (TechAmerica & Foundation, 2012)

Característica	Descripción	Atributos	Manejador
<u>Volumen</u>	Gran cantidad de datos generados que se almacenan, analizan y gestionan para tomar decisiones basadas en el análisis completo de los datos.	Según el Estudio del universo digital de IDC, el mundo está en proceso de generar 1.8 Zettabytes de información, de forma continuo, proyectándose a 35 Zettabytes en 2020.(John Gantz, 2013)	Aumento de las fuentes de datos, sensores de mayor resolución.
<u>Velocidad</u>	Rapidez con la que se producen y modifican los datos y velocidad con la que se deben recibir, analizar y procesar los datos.	<ul style="list-style-type: none"> ▪ Accesibilidad: cuándo, dónde y cómo quiere el usuario la información. ▪ Aplicable: información relevante y valiosa para una 	<ul style="list-style-type: none"> ▪ Incremento en las fuentes de datos. ▪ Potencia computacional mejorada de los dispositivos generadores de datos.

		empresa en tiempo real.	
<u>Variedad</u>	El aumento de la información proveniente de nuevas fuentes crea presiones de integración, administración y arquitectura en TI.	<ul style="list-style-type: none"> • Estructurado • No estructurado • Semiestructurado 	<ul style="list-style-type: none"> • Móvil • Redes sociales • Videos • Chat • Genómica • Sensores

Para adaptarse y asimilar la importancia de Big Data existen muchos desafíos, que las empresas deben plantearse para aumentar su valor. Proyectos en ciencia, gobierno, sector privado o desarrollo tecnológico, se enfrentan a problemas de privacidad, acceso, almacenamiento, procesamiento y análisis de información. También, asumen retos de escalabilidad, calidad de datos o habilidades y técnicas requeridas para manejar grandes datos. Hadoop y Map Reduce son herramientas preferenciales por las empresas para disminuir los múltiples retos que les presenta Big Data (Katal, Wazid, & Goudar, 2013).

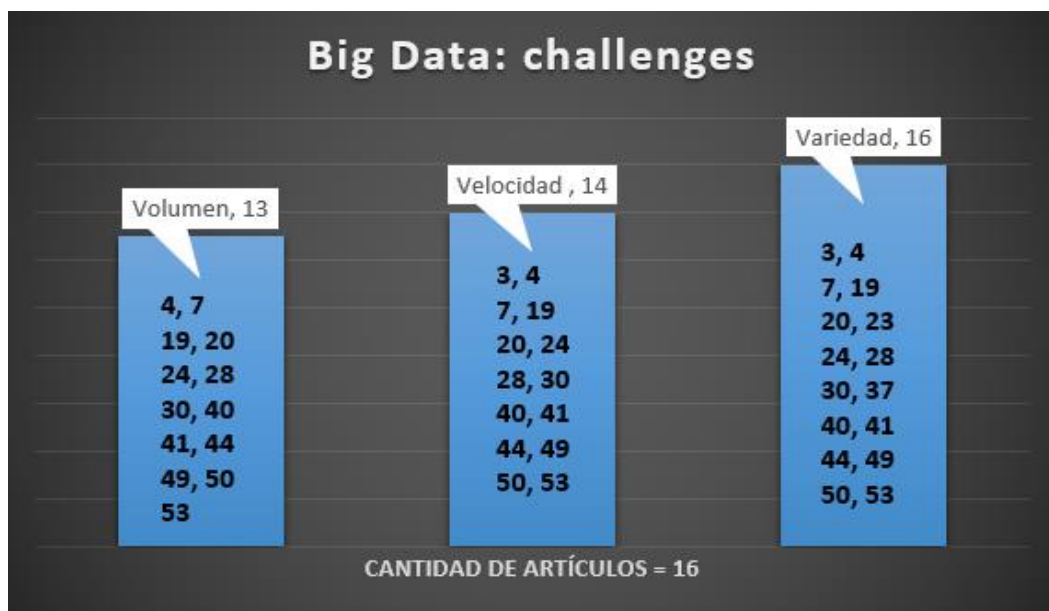
A pesar de las mejoras informáticas en hardware y software, los desafíos de Big Data en etapas de almacenamiento, gestión y procesamiento de datos son causa de la permanente generación y crecimiento de datos. En la tabla 2 se observan los títulos de trabajos relacionados que analizan los retos y características de Big Data.

Tabla 2 Artículos referentes a Big Data

ID	Artículo	Volumen	Velocidad	Variedad
40	Big Data Ecosystem – Review on Architectural Evolution (Singh et al., 2019).	✓	✓	✓
7	Big Data: A Systematic Review (Antonio Fernando Cruz Santos & Ota'vio Manoel Pereira Siqueira, 2018).	✓	✓	✓
23	Hadoop: Solution to Unstructured Data Handling (Madaan et al., 2018).			✓
37	A Big Data system supporting Bosch Braga Industry 4.0 strategy (Santos et al., 2017)			✓
3	Applying Software Engineering Processes for Big Data Analytics Applications Development (Al-Jaroodi et al., 2017).		✓	✓
44	Big Data Challenges and Solutions (Srinivasu et al., 2017).	✓	✓	✓
41	Critical analysis of Big Data challenges and analytical methods (Sivarajah, Kamal, Irani, & Weerakkody, 2017)	✓	✓	✓

28	Big Data – Concepts, Applications, Challenges and Future Scope (Mukherjee & Shaw, 2016)	✓	✓	✓
50	Big Data: A Review (Tjobod, 2015)	✓	✓	✓
30	Data-intensive applications, challenges, techniques and technologies: A survey on Big Data (Philip Chen & Zhang, 2014).	✓	✓	✓
19	Big Data: Issues and Challenges Moving Forward (Kaisler, Armour, Espinosa, & Money, 2013)	✓	✓	✓
20	Big Data: Issues, Challenges, Tools and Good Practices (Katal et al., 2013)	✓	✓	✓
49	Demystifying Big Data (TechAmerica & Foundation, 2012)	✓	✓	✓
53	Big Data: Challenges and Opportunities (Zicari, n.d.)	✓	✓	✓
24	Big Picture of Big Data Software Engineering (Madhavji, Miranskyy, & Kontogiannis, 2015).	✓	✓	✓
4	Characteristics and Requirements of Big Data Analytics Applications (Al-Jaroodi & Mohamed, 2017).	✓	✓	✓

Con base en la lectura de los artículos de la tabla 2, se observa que la variedad es el factor frecuente causante de incertidumbre en los proyectos Big Data ya que, debido a la capacidad actual del hardware, la problemática de volumen y velocidad se ha ido reduciendo. En la gráfica 1 se observa los identificadores de artículos en los cuales se hace referencia al volumen, velocidad y variedad como desafíos en sistemas Big Data.



Gráfica 1 Retos de Big Data

En los artículos consultados referentes a Big Data, los autores caracterizan el volumen, la velocidad y la variedad como los desafíos de almacenamiento y procesamiento más influyentes en los sistemas Big Data. Este trabajo, caracteriza la naturaleza cambiante de proyectos con Macrodatos, provocada por la dificultad de gestionar grandes volúmenes de datos con velocidad de transmisión casi en tiempo real y variedad de fuentes de información. Por lo tanto, se infiere que para obtener el mejor aprovechamiento de análisis estadístico y tomar decisiones que permitan ahorrar recursos o multiplicar ganancias a las empresas, los principios del Manifiesto Ágil se adecuan para gestionar sistemas Big Data.

3.2 Enfoques Tradicionales de Desarrollo de Software.

[A Study of Software Development Life Cycle Process Models \(Ali, 2017\).](#)

En este documento, el autor describe diferentes modelos de Software Development Life Cycle (SDLC). Se explican las ventajas y desventajas de algunos de los modelos importantes de SDLC como Waterfall Model, Iterative Model, Spiral Model, V-Model, Big Bang Model, Agile Model, Rapid Application Development Model y Software Prototype. También se describe cual modelo de SDLC se ajusta mejor a cada tipo de aplicación de software.

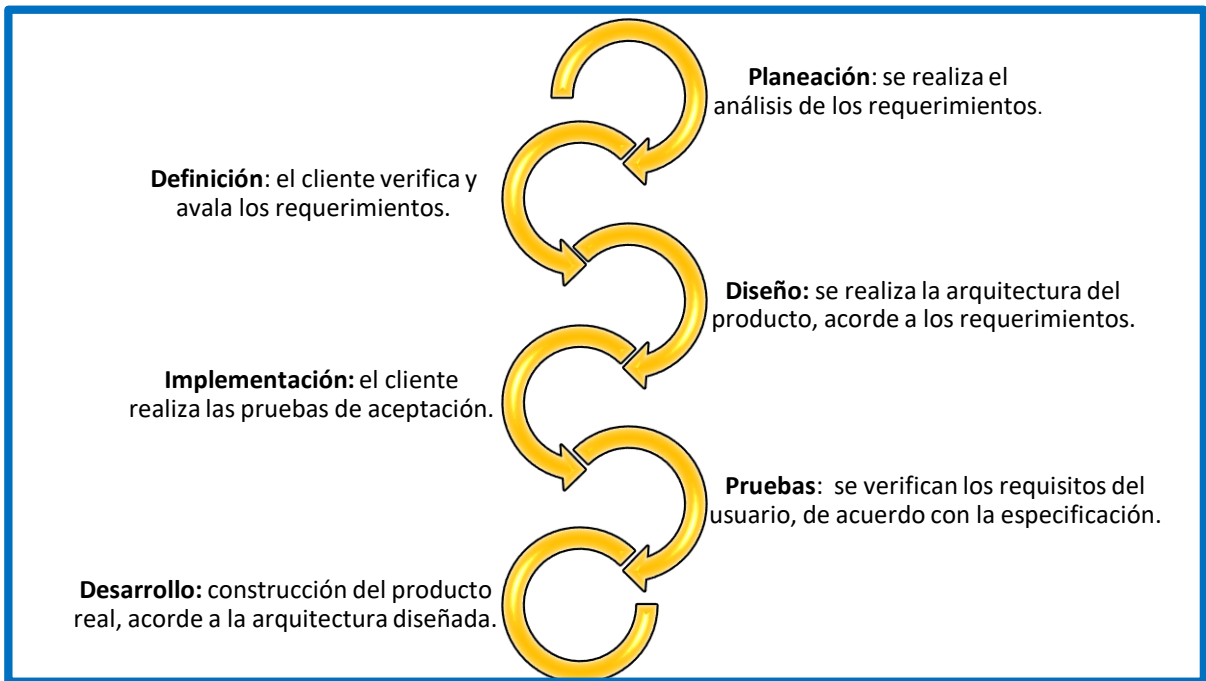
Con base en el conocimiento de que los enfoques tradicionales son adecuados para proyectos secuenciales con requisitos bien definidos y que los métodos ágiles son ideales para proyectos con requisitos de mercado que cambian frecuentemente, se concluye que el análisis de este artículo, es útil para gerentes de proyectos, desarrolladores, docentes, estudiantes y cualquier persona interesada en desarrollo de software, para decidir el tipo de metodología que implementan en sus proyectos.

[Proyectos de desarrollo de software \(Tapias, 2014\).](#)

En esta investigación, el autor describe como la transición de una sociedad de la información a una sociedad del conocimiento, hace pertinente la creación de software cuyas características, satisfagan las demandas de las empresas, respecto a mayores volúmenes de información, automatización de procesos, modelado o realidad virtual. Sin embargo, a pesar de establecer metas para mejorar la calidad del software, una tercera parte de los proyectos falla y más de la mitad supera el presupuesto inicial.

En este trabajo, se desglosan los elementos que conforman al software, así como los problemas respecto a equipo de desarrollo y tecnología utilizada en los proyectos. También se analiza el ciclo de vida de un proyecto de desarrollo de software en distintos modelos. Finalmente, se destacan las ventajas y desventajas de los modelos en cascada, incremental, evolutivo, espiral, concurrente y modelado de requerimientos.

Con base en (Ali, 2017), en la gráfica 2 se observan, las etapas y descripción general del ciclo de vida de desarrollo de software.



Gráfica 2 Etapas del desarrollo de software

Con base en (Ali, 2017) y (Tapias, 2014), en la tabla 3 se observan, las ventajas y desventajas que presenta el modelo en cascada en el desarrollo de software.

Tabla 3 Ventajas y desventajas del Modelo en cascada

Modelo de desarrollo en cascada	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Apropriado para problemas bien entendidos. • Produce resultados predecibles. • Fácil de entender y usar. • Funcional para proyectos pequeños. • Documentación de procesos. • Documentación de resultados. • Fases claramente definidas. 	<ul style="list-style-type: none"> ○ Requisitos fijados al inicio del proyecto. ○ Rigidez para introducir cambios. ○ Poca gestión de cambio de requisitos. ○ Modelo insuficiente para grandes proyectos. ○ Dificultad para medir progreso en las fases. ○ Amplitud de riesgos.

<ul style="list-style-type: none"> • Organización sencilla de tareas. 	<ul style="list-style-type: none"> ○ Hasta el final, nada terminado. ○ Puede quedar obsoleto con rapidez.
--	---

En la literatura revisada en relación al enfoque tradicional de desarrollo de software, los autores analizan como el ciclo de desarrollo de software en distintos modelos, es adecuado para proyectos definidos de forma secuencial. Para este trabajo, son pertinentes las etapas del ciclo de vida de desarrollo de software, así como las ventajas y desventajas que presentan los modelos de desarrollo en cascada.

3.3 Enfoque ágil

[Agile Methodologies: Comparative Study and Future Direction \(Ahmad, Soomro, & Broh, 2014\).](#)

En este estudio los autores exploran tres metodologías populares. Extreme programming (XP), cuya base es el desarrollo iterativo e incremental para entregar software de calidad, acorde a requisitos del cliente. Scrum, implementa iteraciones adecuadas para entornos de proyectos complejos con requisitos cambiantes. Rational Unified Process (RUP) es un enfoque arquitectónico incremental e iterativo, centrado en los principios de la Ingeniería de Software.

La propuesta de los autores es un híbrido XSR (XP, Scrum y RUP) el cual pretende combinar las características de gestión del proyecto de software en Scrum, codificación y fortalezas estándar en las características XP, así como satisfacción del cliente y centrarse en el objetivo de la organización de RUP lo que dará como resultado productos de software de alta calidad con capacidad para cumplir los objetivos comerciales.

[Empirical Study of Agile Software Development Methodologies: A Comparative Analysis \(Matharu et al., 2015\).](#)

En este documento los autores realizan un estudio empírico sobre la elección de Scrum, Extreme Programming y Kanban como las metodologías más populares para el desarrollo de software. El estudio realizado permite a los autores concluir, que desarrollar software basado en metodologías ágiles ofrece una solución efectiva a los desafíos de requisitos dinámicos con los que se enfrenta actualmente la industria del software. La característica de iteración frecuente entre desarrolladores y clientes, permite a los enfoques ágiles ofrecer valor agregado y buen retorno de inversión, demostrando así su utilidad y efectividad. Se destaca Scrum como la metodología con la mayor tendencia de adopción.

[Agile Software Development Methodologies: Survey of Surveys \(Al-Zewairi et al., 2017\)](#)

En este trabajo, los autores presentan un estudio bibliográfico de diferentes metodologías, utilizando una metodología de investigación llamada “Comparar y Revisar” para analizar desde el año 2000 hasta el año 2015 las diferentes metodologías ágiles. El enfoque ágil incluye las siguientes metodologías: Desarrollo de software adaptativo (ASD), Proceso unificado ágil (AUP), Métodos de cristal, Metodología de desarrollo de sistemas dinámicos (DSDM), Programación eXtreme (XP), Desarrollo impulsado por características (FDD), Kanban, Lean Software Development, y Scrum.

Se encontró que la metodología ágil se basa en iteraciones para representar un ciclo de vida del desarrollo del software y acepta cambios en los requisitos de cualquier fase del ciclo de desarrollo. A diferencia del modelo espiral, la metodología ágil ofrece simplicidad en todas las prácticas. Los autores concluyen que la metodología ágil XP es la más investigada, popular y mejor documentada.

[Agile Methodologies in Education: A Review \(Salza, Musmarra, & Ferrucci, 2019\).](#)

En este trabajo los autores exploran las principales metodologías ágiles (XP, Kanban, Scrum) y su adaptación al entorno educativo. Los resultados de investigación, sugieren que el enfoque ágil, además de simular cursos de Ingeniería de Software, se puede aplicar para impartir otras materias. En esta revisión, los autores muestran que existe un interés en aplicar metodologías de aprendizaje ágil para permitir a los estudiantes trabajar de forma específica y efectiva. También, se resalta el esfuerzo de los investigadores para que las metodologías ágiles se formalicen en el contexto educativo tal como *eduScrum*.

Lo que hace que un método de desarrollo sea ágil, es el desarrollo incremental con pequeños lanzamientos de software en ciclos rápidos, es cooperativo con los clientes y desarrolladores, quienes trabajan constantemente en conjunto para tener comunicación cercana, la metodología ágil es fácil de aprender y modificar además de estar bien documentada siendo su principal característica la capacidad para realizar cambios de último momento (Abrahamsson, Salo, & Warsta, 2002). El enfoque ágil mejora los tiempos de desarrollo cumpliendo con el objetivo de satisfacer al cliente, por lo cual se concluye que en términos de productividad la agilidad es la mejor metodología (Sharma, 2012).

La usabilidad es pertinente como indicador de calidad, ya que para el usuario lo más importante es interactuar con el software. Se adoptaron metodologías ágiles en el desarrollo de software para tratar problemas persistentes, tales como costosas modificaciones del software y demoras excesivas en el proceso de desarrollo en general.

Los resultados muestran que las metodologías ágiles en las que se han aplicado más los métodos de usabilidad son Scrum y Extreme Programming (Salvador, Nakasone, & Pow-Sang, 2014). Durante el ciclo de vida del desarrollo de software, el mantenimiento es la etapa más costosa. Para solventar este inconveniente, las metodologías ágiles son adecuadas para proyectos variables, debido a que, aceleran la entrega de productos de

calidad y lanzan prototipos cortos para que sean revisados por los usuarios a través del equilibrio entre costo, requisitos y calidad de productos (Tarwani & Chug, 2016).

Respecto a proyectos Big Data, existe una gran necesidad de contar con una metodología específica que se enfoque en análisis, personas, tecnología y procesos (Saltz & Shamshurin, 2016). Los métodos ágiles son adecuados para desarrollar sistemas Big Data, ya que, involucran a la comunidad empresarial durante el desarrollo, además de entrega temprana de productos útiles. Cuenta con desarrollo evolutivo e iterativo, permitiendo cambio de herramientas, enfoques o datos. Es flexible, incluso a cambios tardíos en beneficio del objetivo comercial (Dharmapal & Sikamani, 2016).

Las ventajas que presenta el enfoque ágil en aspectos tales como tamaño o dinamismo permite concluir que para enfrentar los constantes cambios de requisitos en sistemas Big Data, la agilidad es recomendable en proyectos pequeños (Franková et al., 2016). La alternativa para el desarrollo de sistemas complejos son las metodologías Scrum y Extreme Programming debido a que cuentan con la mayor preferencia en la industria del software por permitir cambios de requisitos de parte del cliente además de ayudar a los desarrolladores a lograr productos rápidos y funcionales (Arndt, 2018).

Las metodologías ágiles, se presentan a partir de la creación del Manifiesto de Desarrollo de Software Ágil publicado por profesionales y consultores de software en 2001. A través de la experiencia y el apoyo a terceros se descubren mejores formas de desarrollar software, por lo tanto, aunque los valores que se observan en la figura 6 son convergentes, se aprendió a priorizar los elementos de la izquierda (Abrahamsson et al., 2002).

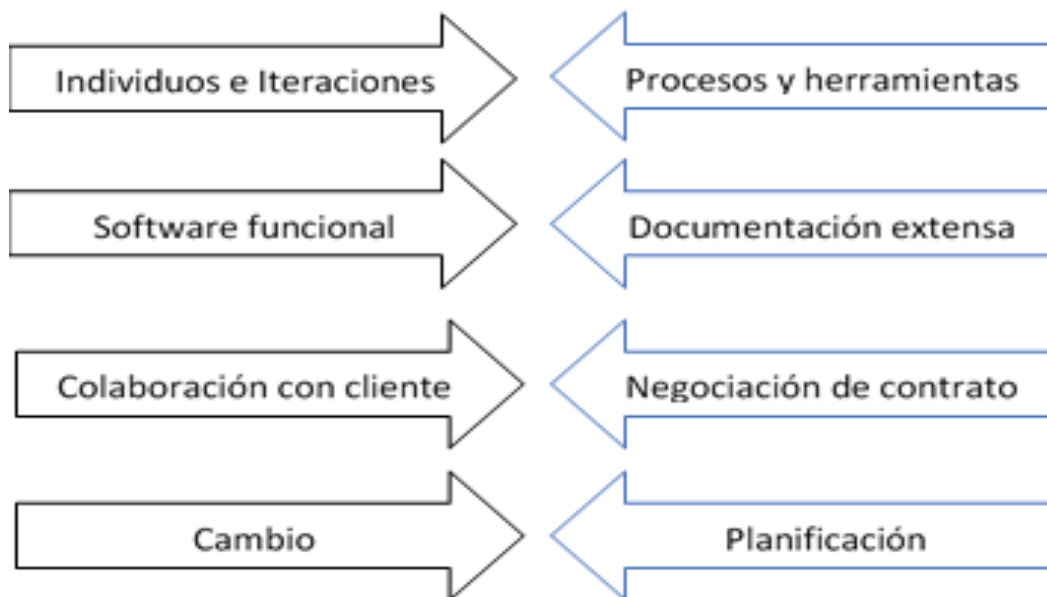


Figure 6 Aspectos de desarrollo de software ágil (Abrahamsson et al., 2002).

Como referencia, el manifiesto de desarrollo de software ágil surge en febrero de 2001 en una reunión entre representantes de Scrum, Extreme Programming, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Feature-Driven Development (FDD), Pragmatic Programming, y otros simpatizantes de una alternativa a procesos de desarrollo de software pesados y documentación tediosa (Beck et al., 2001).

A continuación de forma general, se mencionan los doce principios del manifiesto ágil:

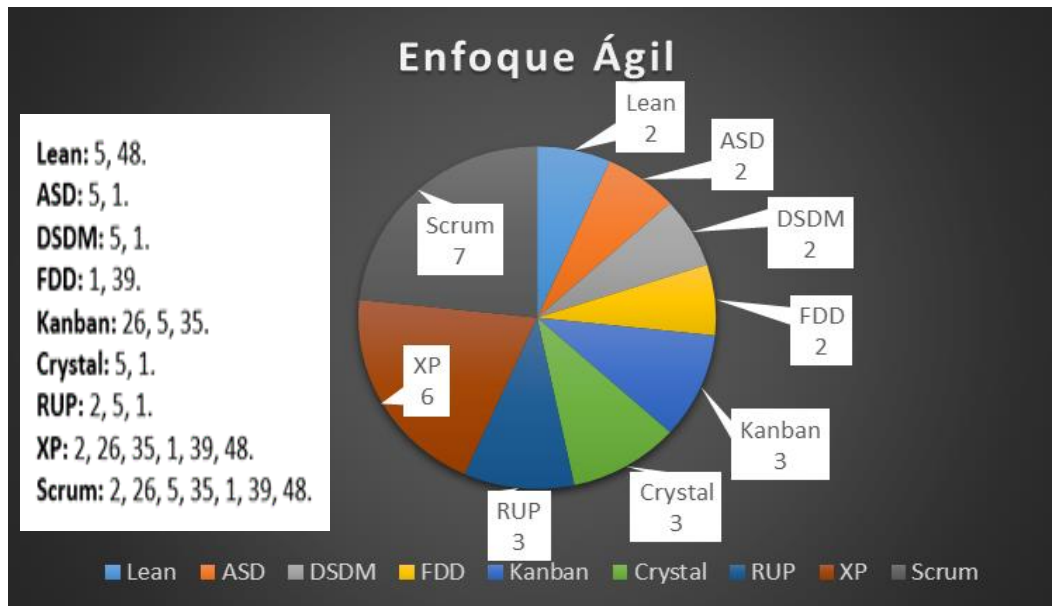
- ✓ Se prioriza satisfacer al cliente con entregas tempranas y continuas de software.
- ✓ Se aceptan cambios de requisitos en cualquier etapa del desarrollo.
- ✓ Se entrega software funcional frecuentemente, entre dos semanas y dos meses.
- ✓ Responsables de negocio y los desarrolladores trabajan juntos durante todo el proyecto.
- ✓ Desarrollar proyectos en torno a individuos motivados otorgándoles el apoyo necesario y confiarles la ejecución del trabajo.
- ✓ El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- ✓ El software funcionando es la medida principal de progreso.
- ✓ Promotores, desarrolladores y usuarios promueve el desarrollo sostenible llevando un ritmo constante de forma indefinida.
- ✓ Mejorar la agilidad, implica atención continua al diseño y la excelencia técnica.
- ✓ La simplicidad es esencial para favorecer la colaboración con del cliente.
- ✓ Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- ✓ Intervalos regulares para que el equipo reflexione sobre cómo ser más efectivo para ajustar y perfeccionar su comportamiento.

En la literatura referente al enfoque ágil, los autores presentan las principales características y ventajas que las metodologías ágiles tienen respecto a enfoques tradicionales o en cascada. De la agilidad, se resalta su capacidad de adaptación a proyectos de desarrollo de software con diferentes tamaños, requisitos cambiantes y objetivos no siempre definidos. Sin embargo, algunas desventajas ágiles son: dependencia de las personas y falta de reusabilidad a consecuencia de la poca documentación. En la tabla 4 se observan los títulos de trabajos en los que se analizan los beneficios logrados por la implementación del enfoque ágil.

Tabla 4 Lista de estado del arte de artículos referentes al enfoque ágil.

Id	Artículos
2	Agile Methodologies: Comparative Study and Future Direction (Ahmad et al., 2014).
26	Empirical Study of Agile Software Development Methodologies: A Comparative Analysis (Matharu et al., 2015).
5	Agile Software Development Methodologies: Survey of Surveys (Al-Zewairi et al., 2017)
35	Agile Methodologies in Education: A Review (Salza et al., 2019).
1	Agile Software Development Review and Analysis (Abrahamsson et al., 2002)
39	Agile Processes and Methodologies: A Conceptual Study (Sharma, 2012)
34	A systematic review of usability techniques in agile methodologies (Salvador et al., 2014)
48	Agile methodologies in software maintenance: A systematic review (Tarwani & Chug, 2016)
33	Big data team process methodologies: A literature review and the identification of key factors for a project's success (Saltz & Shamshurin, 2016)
12	Big data analytics using agile model (Dharmapal & Sikamani, 2016)
14	Agile Project Management Approach and its Use in Big Data Management (Franková et al., 2016)
8	Big Data and software engineering: prospects for mutual enrichment (Arndt, 2018)

Además de analizar la usabilidad y la aplicación de ingeniería de software como características de calidad en desarrollo de software, también se revisa la necesidad de aplicar metodologías en sistemas Big Data que ayuden a resolver su característica problemática de incertidumbre debido a sus cambios constantes. En la gráfica 3 se observan los artículos de la tabla 4 que revisan las características de las metodologías ágiles. Scrum destaca como la metodología ágil más estudiada en la literatura.



Grafica 3 Enfoque ágil

En los trabajos revisados los autores destacan que el enfoque ágil es ideal para aplicarse en proyectos que, debido a la complejidad para definir requisitos, requieran cambios en cualquiera de sus fases de desarrollo. Para este trabajo de investigación, es conveniente conocer que el enfoque ágil permite desarrollo iterativo e incremental los cuales mejoran los tiempos de pequeños lanzamientos de productos terminados. Sin embargo, la agilidad al estar orientada a personas, implica desventaja debido a que complica la comunicación en equipos de desarrollo con más de 20 integrantes (Awad, 2019). Otras desventajas ágiles son: mayor esfuerzo de pruebas, escalabilidad complicada, e incremento de deuda técnica (Sánchez, 2018).

3.4 Enfoque tradicional versus Enfoque ágil

En esta sección se describen trabajos en lo que se contrastan las características de los enfoques tradicionales y ágiles destacando parámetros generales de ambos enfoques (tablas 5-6). De forma general, en la figura 7 se observan las etapas funcionales de ambos enfoques.



Figura 7 Ciclos de enfoques tradicional y ágil

Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software (Montero, Cevallos, & Cuesta, 2018).

En este documento de investigación los autores hacen una revisión de publicaciones de metodologías ágiles y tradicionales. Del enfoque tradicional destacan que sus procesos son prescriptivos, secuenciales, carentes de comunicación con el cliente y rígidos para los cambios. Los modelos en cascada tienen cinco etapas bien definidas: análisis de requisitos, diseño, codificación, pruebas y mantenimiento.

Por otra parte, el enfoque ágil cuya principal particularidad es la flexibilidad, ya que los proyectos desarrollados en el entorno ágil, son divididos en proyectos más pequeños con entregas funcionales, retroalimentación constante y posibilitar cambios de requerimientos por parte del cliente como una característica especial. En la investigación se concluye que no se puede tomar una metodología universal, por lo tanto, es recomendable analizar las características del proyecto para determinar que metodología es conveniente implementar.

Challenges of Traditional and Agile Software Processes (Sánchez, 2018).

En este documento la autora proporciona una visión general de la evolución de los métodos de desarrollo de software, así como una revisión de los enfoques clásico y ágil que existen en la actualidad. Aunque el modelo en cascada se estableció durante un tiempo como ideal para el desarrollo de software, se reconoce como una actividad compleja que con el paso del tiempo generó problemas por documentación excesiva o cambios costosos.

Los enfoques ágiles surgen como alternativa para simplificar el proceso de software y promover ciclos de tiempos cortos. Sin embargo, la evidencia sugiere que no hay un enfoque único para el desarrollo de software, por lo tanto, se concluye como ideal

aprovechar lo mejor de la disciplina y agilidad, dependiendo de factores específicos relacionados al contexto de la organización, proyecto o equipo de desarrollo.

[A Comparison between Agile and Traditional Software Development Methodologies \(Awad, 2019\).](#)

En este trabajo el autor describe las características de algunas metodologías tradicionales y ágiles utilizadas para desarrollar software. Se realiza un análisis de fortalezas y debilidades de ambos enfoques, se describe la importancia del manifiesto ágil, el cual prioriza las iteraciones, colaboración con el cliente y respuesta al cambio frente a la documentación excesiva y procesos planificados. El autor refuerza su investigación con cuestionarios a profesionales de la industria de software para evaluar que metodología tiene mejor tasa de éxito. El autor concluye que los enfoques ágiles son benéficos principalmente para proyectos de software de pequeña y mediana escala.

[Analysis of Software Development Methodologies \(SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, 2019\).](#)

En este documento los autores hacen un análisis de 21 metodologías para desarrollar software entre las cuales destacan por su popularidad: Cascada, Iterativa, Espiral, Programación Extrema, Scrum, Proceso Racional Unificado, Lean y Desarrollo de Software Adaptivo (ASD). El análisis de los autores tiene como objetivo elegir la mejor metodología en función de factores tales como: recursos disponibles, tamaño y tipo del proyecto.

Los autores concluyen que las metodologías clásicas son adecuadas para proyectos que no cambian requisitos y permiten una complejidad detallada del proyecto. Estas metodologías son fáciles de entender e implementar debido a que tienen una documentación completa. Por otra parte, los métodos ligeros son adecuados para proyectos cuyos requisitos no están definidos y pueden modificarse tanto por factores internos como externos.

En la tabla 5 se observa el contraste de características generales destacadas entre Metodologías ágiles y Metodologías tradicionales.

Tabla 5 Estudio comparativo de metodologías de desarrollo de software ágiles y tradicionales (Matharu et al., 2015).

Parámetro	Metodologías Ágiles	Metodologías Tradicionales
❖ Aprendizaje	✓ Durante el desarrollo	▪ Posterior al desarrollo
❖ Cambio	✓ Adaptable	▪ Compatible

❖ Documentación	✓ Baja	▪ Alta
❖ Enfoque	✓ Adaptativo	▪ Predictivo
❖ Gestión	✓ Colaborativa	▪ Controlada
❖ Orientación	✓ A las personas	▪ A los procesos
❖ Planeación	✓ Corto plazo	▪ Largo plazo
❖ Tamaño de proyecto	✓ Mediano/Pequeño	▪ Grande

De acuerdo a (SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, 2019), en la tabla 6, se observan los enfoques ágiles compatibles en tamaños o escalas de proyectos.

Tabla 6 Metodologías compatibles sobre tamaños de proyectos (SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, 2019).

Metodología	Ideal para tamaños:
➤ Cascada	✓ Pequeños
➤ Incremental	✓ Medianos, Grandes
➤ Espiral	✓ Medianos, Grandes
➤ Sistemas dinámicos (DSDM)	✓ Medianos, Grandes
➤ Desarrollo adaptativo (ASD)	✓ Pequeños, Medianos, Grandes
➤ Cristal	✓ Pequeños, Medianos, Grandes

➤ Basada en funciones (FDD)	✓ Pequeños, Medianos, Grandes
➤ Proceso racional unificado (RUP)	✓ Pequeños, Medianos, Grandes
➤ Lean	✓ Pequeños, Medianos, Grandes
➤ Extreme Programming	✓ Pequeños, Medianos, Grandes
➤ Scrum	✓ Pequeños, Medianos, Grandes

En los artículos de cotejo entre los enfoques tradicionales y ágiles, los autores acentúan que en proyectos con etapas y requisitos rígidos lo recomendable es aplicar un enfoque tradicional. Por otra parte, si los proyectos son cambiantes es ideal aplicar la flexibilidad del enfoque ágil ya que permite dividir el proyecto en pequeñas partes. Para esta investigación, es conveniente conocer la función de los parámetros en ambos enfoques, así como el tamaño de proyectos en los que mejor se desempeña cada enfoque.

3.5 Metodologías Ágiles

Es determinante saber que no existe una metodología común para desarrollar cualquier proyecto de software. Por una parte, los enfoques tradicionales tienen cierta resistencia a los cambios mientras que los enfoques ágiles requieren la disponibilidad del cliente para lograr retroalimentación continua (Montero et al., 2018). Aspectos tales como orientado a personas, balance entre flexibilidad y planeación además de enfoque descentralizado y auto organización de equipos, son las principales diferencias en ágil y tradicional (Awad, 2019).

Esta sección presenta de forma general la utilidad de implementar enfoques ágiles. También de forma breve se hace referencia a cuatro metodologías ágiles seleccionadas para investigar de acuerdo a lo siguiente: Lean como origen del sistema de producción Toyota en occidente. Kanban como sucesor de Lean, además junto con Extreme Programming y Scrum son las metodologías ágiles con más referencias bibliográficas.

3.5.1 Beneficios

La utilidad que ofrecen las metodologías ágiles, es poder elegir un marco de trabajo o bien, mezclar varias prácticas y personalizar el enfoque de acuerdo al contexto de los proyectos dependiendo de los perfiles del equipo y del tipo de producto a desarrollar. De forma general, en la figura 8 se observan los beneficios que de acuerdo a (Matharu et al., 2015) se obtienen al implementar enfoques ágiles en distintos proyectos.

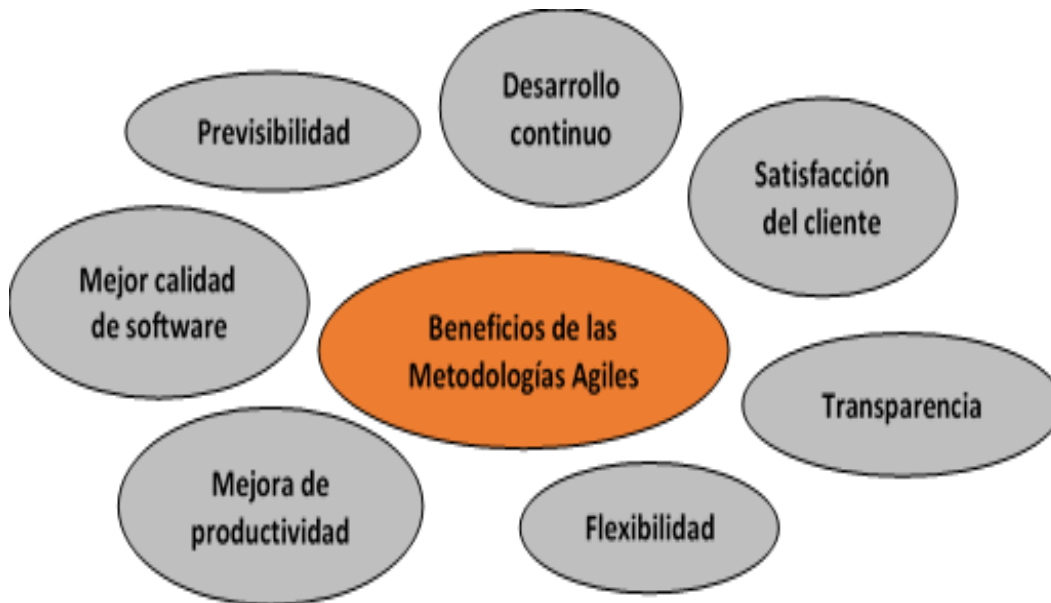
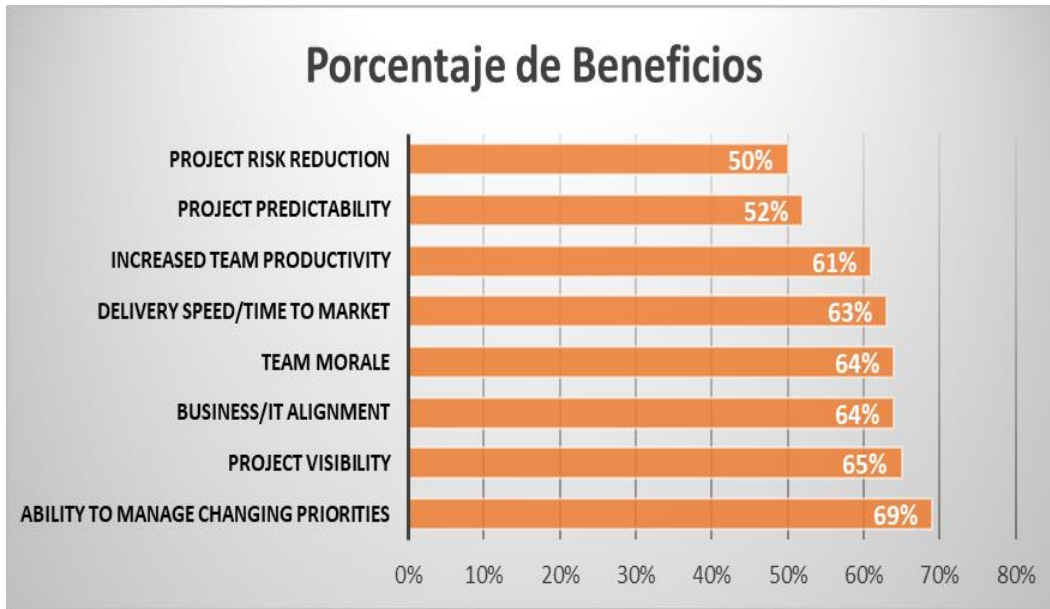


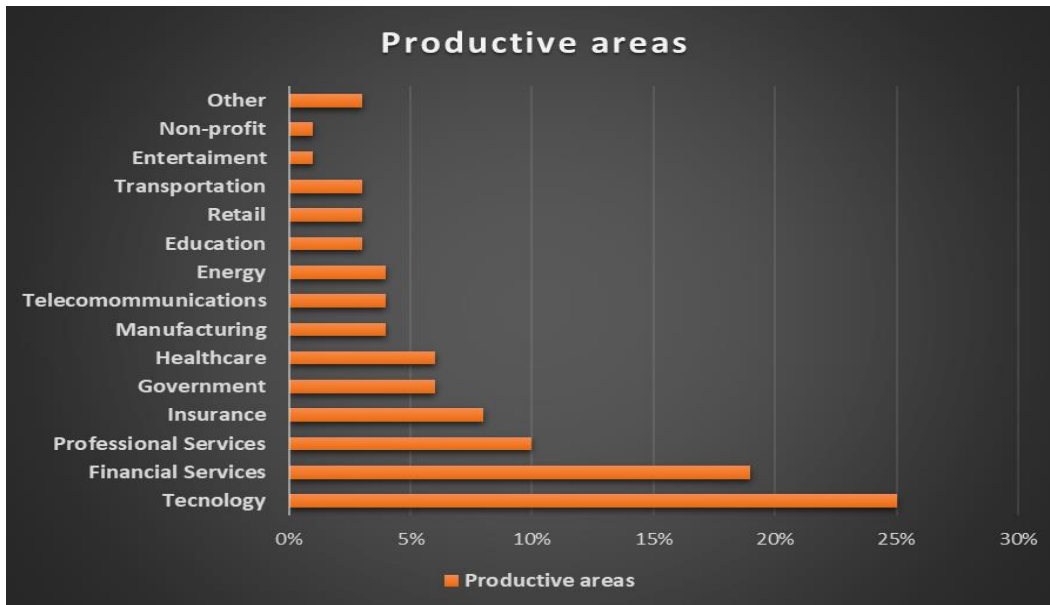
Figure 8 Beneficios ágiles (Matharu et al., 2015)

De acuerdo a (Version One, 2019), en la gráfica 4 se observan los aspectos en los cuales las metodologías ágiles, impactan de forma positiva en empresas donde se aplica tal enfoque.



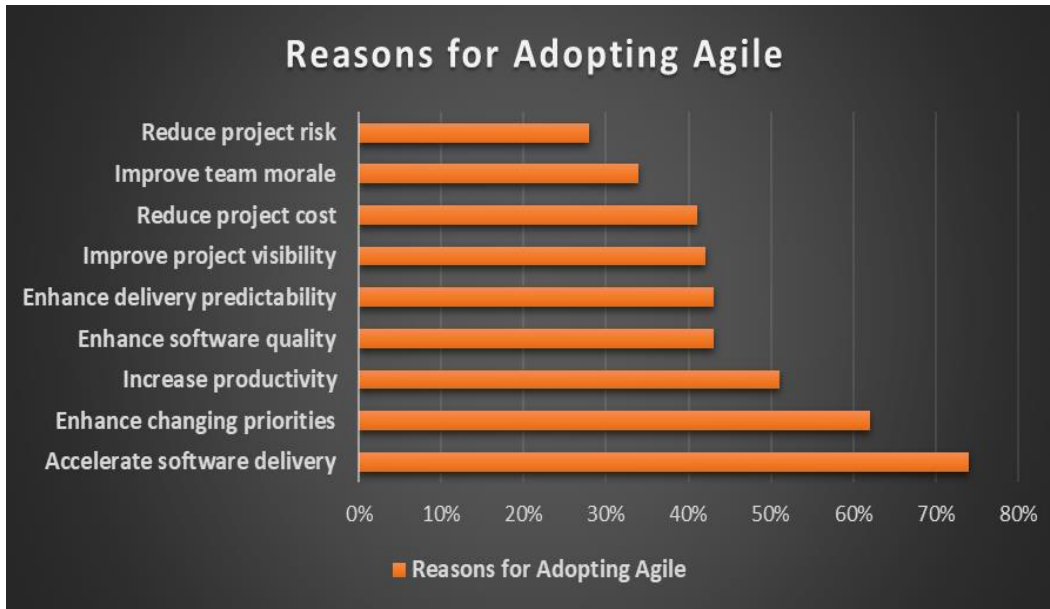
Gráfica 4 Aspectos de beneficios (Version One, 2019).

De acuerdo a (Version One, 2019), en la gráfica 5 se observa las diversas áreas productivas donde se implementa el enfoque ágil entre los cuales, destaca el sector tecnológico.



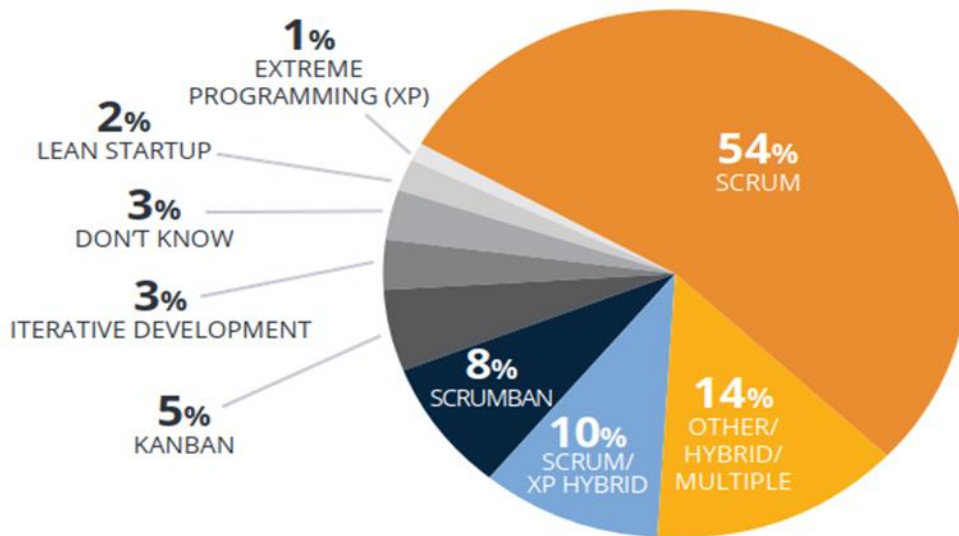
Gráfica 5 Áreas de implementación (Version One, 2019)

De acuerdo a (Version One, 2019), en la gráfica 6 se observan las razones por las que diversos sectores económicos implementan el enfoque ágil en sus procesos de negocio.



Gráfica 6 Razones comerciales para adoptar Agile (Version One, 2019)

De acuerdo a la gráfica 7 se observa que, la metodología Scrum es la más utilizada en organizaciones que implementan el enfoque ágil en sus procesos de negocio.



Gráfica 7 Metodologías ágiles más utilizadas (Version One, 2019)

A continuación, se presentan de forma general, características de las metodologías ágiles más utilizadas:

3.5.2 Lean

El término significa manufactura esbelta o ágil y es el nombre con que se conoce en occidente al sistema de producción de la empresa Toyota. Su objetivo es la satisfacción del cliente, a través de entregar productos y servicios de calidad que el cliente necesita, cuando lo necesita y en la cantidad requerida al precio correcto, utilizando la mínima cantidad de materiales, equipamiento, trabajo y tiempo. En la figura 9 se observa el objetivo Lean de producir más con menos recursos (Galgano, 2008). Información detallada en [Anexo A](#).



Figure 9 Lean, mejorar reduciendo (Galgano, 2008).

3.5.3 Kanban

Es una herramienta del método de abastecimiento Justo a tiempo, el cual pertenece a la filosofía de producción Lean (Ríos, 2015). La palabra es de origen japonés y se compone de los términos:

Kan= “visual”

Ban = “insignia”

Kanban = Insignia visual

Su objetivo es gestionar como se van completando las tareas, y en la actualidad ha ganado popularidad ya que proporciona un medio para visualizar y limitar el progreso del trabajo durante el proceso de desarrollo de software debido a que es una metodología adaptativa que tiene la menor resistencia al cambio (Matharu et al., 2015). En la figura 10 se observa el ejemplo de un tablero Kanban para visualizar el progreso del trabajo. Información detallada en [Anexo B](#).



Figura 10 Ejemplo de tablero Kanban (Pérez, 2012).

3.5.4 Extreme Programming.

Se centra principalmente en la fase de desarrollo más que en el aspecto administrativo de los proyectos de software. Inicialmente se desarrolla un plan de lanzamiento. El usuario escribe historias de usuarios para describir lo que quiere y es parte del equipo de desarrolladores. Esto asegura que todos los requisitos se agreguen de acuerdo y en presencia del usuario. El equipo divide la tarea en iteraciones y al final la prueba de aceptación se realiza para satisfacer al usuario (Tarwani & Chug, 2016). El aspecto más sorprendente de la Programación Extrema son las reglas simples, las cuales se muestran en el diagrama de flujo de la figura 11. Información detallada en [Anexo C](#).

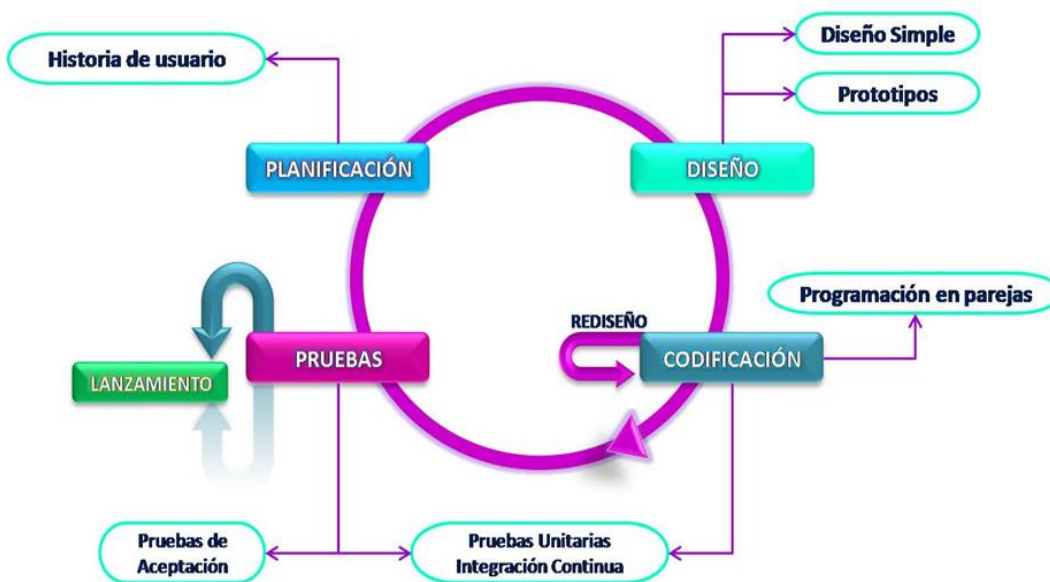


Figura 11 Flujo XP (Wells, 2009).

3.5.5 Scrum.

Desarrollado en la década de 1990 por Ken Schwaber, Mike Beedle y Jeff Sutherland. Es un método ágil incremental e iterativo enfocado no solo en el desarrollo sino también en aspectos administrativos. En Scrum el trabajo se divide en ciclos de trabajo llamados sprints. En cada sprint los requisitos se priorizan y se denominan historias de usuario. Esto se hace para desarrollar el requisito de valor más alto primero para el usuario. Desde entonces, Scrum ha aumentado en popularidad y actualmente es el método de desarrollo de proyectos favorito de múltiples organizaciones a nivel mundial (Tarwani & Chug, 2016). En la figura 12 se observa el ciclo general de Scrum. Información detallada en [Anexo D](#).

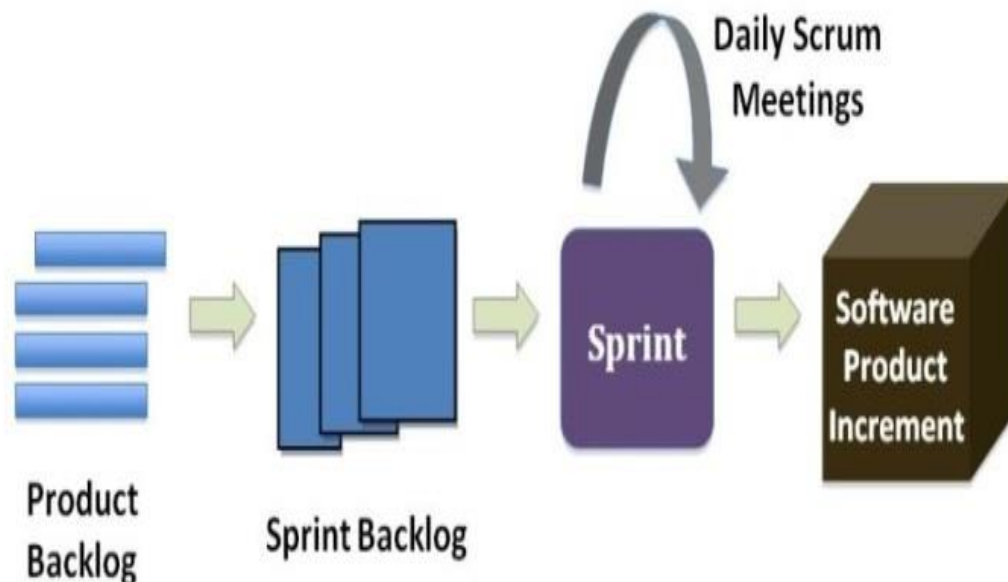


Figura 12 Ciclo Scrum (Matharu et al., 2015).

En la literatura referente a las metodologías ágiles, los autores destacan la flexibilidad que tienen para combinar sus características, así como las razones de adaptar el enfoque ágil para obtener los beneficios que aporta su aplicación en organizaciones de diversas áreas económicas. Para esta investigación, es relevante la filosofía de Lean respecto a producir más con menos recursos, la utilización de tableros Kanban para gestionar tareas, la técnica para elaborar historias de usuario de Extreme Programming y la priorización de requisitos a través de sprints incrementales de Scrum.

En esta investigación, para el caso de estudio, se elige aplicar la metodología ágil Scrum. Algunas razones a destacar para respaldar la elección es que, como framework, es adaptable e iterativo. También, es compatible con el desarrollo de cualquier tipo productos, servicios, industrias o proyectos. Destaca el uso de equipos auto organizados que dividen su trabajo en ciclos cortos llamados sprints (SCRUMstudy, 2017).

Otro aspecto relevante que se tomó en cuenta para aplicar Scrum al caso de estudio, es la evaluación de los cuatro puntos de vista para clasificar métodos: “uso”, “agilidad”, “aplicabilidad” y “productos”(Iacovelli & Souveyet, 2008), el cual se puede observar detalladamente en el [Anexo F](#). Para el caso de estudio se aplicó Scrum combinado con el método de estimación Planning Poker de Extreme Programming y el tablero Kanban para gestión de tareas.

Capítulo 4 Metodología de solución

En este capítulo, se describe porque se elige Scrum como la metodología ágil adecuada para atender los desafíos de los sistemas Big Data. Por medio de un caso de estudio, se analizan las fases de procesamiento de datos para mapear e implementar los procesos de Scrum.

4.1 Metodología de solución

Para este trabajo de investigación la metodología de solución implementada fue investigar y caracterizar las cuatro metodologías ágiles más utilizadas. Identificar las fases del desarrollo de sistemas de software donde se aplica la agilidad para mejorar resultados y a partir de esta investigación determinar cuál metodología ágil o cual combinación de metodologías ágiles son adaptables al desarrollo de sistemas Big Data.

En la figura 13 se observa de forma general la metodología de solución implementada para encontrar una metodología ágil recomendable como guía para desarrollar proyectos Big Data.



Figura 13 Descripción general de la metodología de solución

4.3 Caso de estudio.

El objetivo del caso de estudio es analizar el comportamiento de la imagen transmitida en línea de las redes sociales de Barcelona de 117,487 blogs de viajes relevantes escritos en inglés por turistas que han visitado la ciudad durante el periodo 2005-2015. El método utilizado en el caso de estudio para recopilar y analizar los datos del contenido generado por usuarios, consta de cuatro fases posteriores a la elección de la ciudad destino (Marine-Roig & Anton Clavé, 2015):

- 1) Selección de alojamiento web,
- 2) Recopilación de datos,
- 3) Pre procesamiento de contenido y
- 4) Análisis de contenido.

Este método tiene como base el paradigma de procesamiento por lotes en el cual primero se almacenan los datos y luego se analizan. Además de revisar el caso de estudio, en el documento se propone como solucionar cada una de las etapas que de forma general tienen los proyectos Big Data aplicando los procesos de la metodología ágil Scrum.

4.4 Mapeo del caso de estudio con Scrum

A continuación se describe el proceso de mapeo que se realizó entre la metodología Scrum, el tablero de gestión de tareas Kanban y las historias de usuario de Extreme Programming con el caso de estudio “Turismo inteligente de la ciudad de Barcelona” (Marine-Roig & Anton Clavé, 2015) de un sistema Big Data. La descripción completa del caso de estudio se encuentra en [Anexo E](#).

4.4.1 Marco Scrum

Para realizar el mapeo entre el sistema Big Data antes mencionado y la metodología ágil Scrum, se sigue el flujo de comportamiento que se observa en la figura 14.

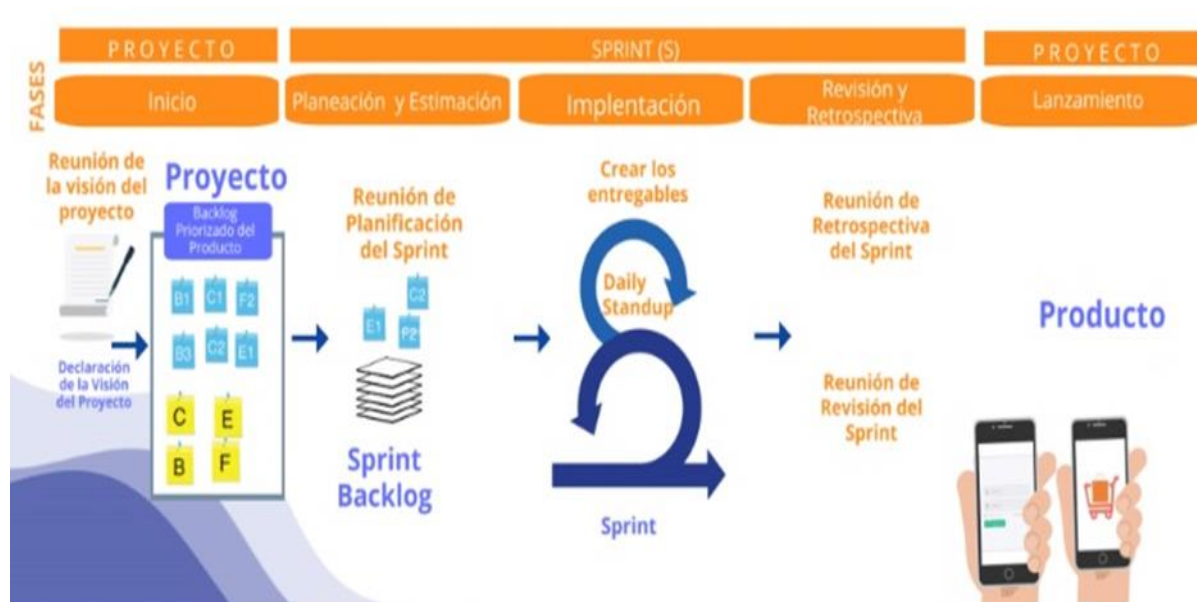


Figura 14 Flujo de proyecto Scrum (SCRUMstudy, 2017)

4.4.2 Mapeo Big Data-Scrum

De forma específica el mapeo entre los procesos Scrum y las etapas Big Data del caso de estudio, se pueden observar en la tabla 7.

Tabla 7 Mapeo Scrum-Big Data

Procesos Scrum	Etapas Big Data
Inicio	Requisitos metodológicos y tecnológicos
Estimación	1) Selección de alojamiento web 2) Recopilación de datos 3) Pre procesamiento de contenido 4) Análisis de contenido
Implementación	
Retrospectiva	
Lanzamiento	Producto terminado

4.4.3 Implementación.

Se considera como iteración o sprint cero la fase de inicio del proyecto en la cual se prepara el equipo Scrum durante diez días laborables (dos semanas), con las condiciones tecnológicas y metodológicas para iniciar los siguientes sprints tal como se observa en la figura 15.

Cronograma Sprint 0						
2019 Octubre						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 Product Owner, Vision del proyecto	19	20
21 Scrum Master, Stakeholder	22 Equipo de desarrollo	23 Epicas 1, 2	24 Epicas 3, 4	25 Product Backlog Priorizado	26	27
28 Cronograma Sprint 1	29 Cronograma Sprint 2	30 Cronograma Sprint 3	31 Cronograma Sprint 4			

Figura 15 Cronograma Sprint 0

4.4.4 Procesos Scrum

En el proceso **Inicio** de Scrum se implementan seis fases que permiten definir la visión y actividades del proyecto Big Data, requisitos metodológicos y tecnológicos, definición de roles, desarrollar las épicas y priorizarlas a través de un Backlog y cronograma. De forma general se planea el inicio de los sprints.

4.4.4.1 Proceso 1: Inicio

4.4.4.1.1 Crear la visión del proyecto.

En esta fase, se realiza una reunión para establecer los objetivos a lograr a través de la definición del caso de negocio y la visión del proyecto, también se identifica al Product Owner quien se encarga de establecer herramientas tecnológicas y métodos de desarrollo.

Entrada	Herramientas	Salidas
1.- Caso de negocio del proyecto	1.- Reunión de la visión del proyecto	1.- Product Owner identificado 2.- Reunión de la visión del proyecto

Actividades

A. Reunión de un día para identificar al Product Owner cuyo rol será encargarse de comunicar la visión del producto al equipo de desarrollo y representar el interés del cliente por medio de los requisitos y la priorización. Realiza las siguientes funciones:

- Enlista requisitos para el software.
- Define las historias de usuario.
- Prioriza los ítems del producto.
- Valida el primer Sprint.

Requisitos tecnológicos:

- Laptop Dell Procesador Inter Core i3 1.70 Ghz
- Disco Duro 2 Terabytes
- Memoria RAM 8 Gigabytes
- Windows 10 de 64 Bits
- Navegador web Google Chrome Versión 78.0.3904.108 (Build oficial) (64 bits)
- Copiadora web, específicamente Offline Explorer Enterprise.
- Microsoft Expression Web 4 (Versión gratuita).
- Analizador de contenido del sitio web, Site Content Analyzer (SCA).

Requisitos metodológicos:

- Metodología ágil Scrum
- Tablero Kanban
- Método de estimación (Planning Poker)

B. Reunión de un día para crear la visión del proyecto con base en el análisis de objetivos del caso de negocio.

Caso de negocio: análisis de Big Data para dar soporte a destinos inteligentes.

Visión del proyecto: analizar el comportamiento de la imagen transmitida en línea en redes sociales de la ciudad de Barcelona de las opiniones en 117 487 blogs de viaje sobre la experiencia de turistas que visitaron Barcelona durante el periodo de años 2005-2015. Aplicación Java basado en la Biblioteca de detección de idiomas.

4.4.4.1.2 Identificar al Scrum Master y al stakeholder

En esta fase, se elige a la persona cumpla con las características necesarias para el rol de Scrum Master, también que o quien será el Stakeholder ya que de este rol depende las entradas o recursos para crear el producto del proyecto.

Entrada	Herramientas	Salidas
1.- Product Owner 2.- Reunión de la visión del proyecto	1.- Criterios de selección	1.- Scrum Master identificado 2.- Stakeholder identificado

Actividades
<p>A. Reunión de un día para elegir a la persona que tendrá el rol de Scrum Master, cuyas características sean la siguientes:</p> <ul style="list-style-type: none"> • Habilidad y experiencia para resolver problemas que enfrente el Equipo Scrum. • Disponibilidad para organizar reuniones de lanzamiento, Daily Standup y Sprints. • Compromiso para asegurar que el Equipo Scrum cuente con un ambiente laboral conductivo para garantizar la entrega exitosa de proyectos Scrum. • Estilo de liderazgo servicial para ser enlace entre el Product Owner y el equipo de desarrollo. <p>B. En la misma reunión se identifica a la persona o personas cuyo rol será Stakeholder como termino colectivo que incluye clientes, usuarios y patrocinadores. Tienen las siguientes características:</p> <ul style="list-style-type: none"> • Proporcionar entradas para la creación del producto del proyecto. • Dueño del negocio con contacto directo con el Product Owner <p><u>Stakeholders:</u> Contenido generado por usuarios. (User Content Generated)</p>

4.4.4.1.3 Formar el equipo SCRUM

En esta fase, a través de una reunión se elige a las personas que cuenten con los aspectos adecuados para formar el Equipo de desarrollo.

Entrada	Herramientas	Salidas
1.- Product Owner 2.- Scrum Master 3.- Reunión de la visión del proyecto	1.- Selección del equipo Scrum	1.- Equipo Scrum identificado

Actividades
<p>1. Reunión de un día para elegir a las personas que forman parte del Equipo Scrum, cuyas características sean la siguientes:</p> <ul style="list-style-type: none"> • Generalistas / especialistas con conocimiento en diversos campos y expertos en al menos uno. • Equipos de 4 a 8 personas con auto organización, motivación, colaboración y misma localización.

4.4.4.1.4 Desarrollar épicas

En esta fase, se realiza una reunión para definir de forma breve y concreta las épicas que tiene el proyecto Big Data de Turismo inteligente.

Entrada	Herramientas	Salidas
<p>1.- Equipo principal de Scrum</p> <p>2.- Declaración de la visión del proyecto</p>	<p>1.- Reuniones de grupo de usuarios</p>	<p>1.- Épicas</p>

Actividades
<p>A. Se realiza una reunión de dos días entre el equipo principal de Scrum (Product Owner, Scrum Master, Development Team), para definir las épicas y prototipos.</p> <p>Épicas:</p> <ul style="list-style-type: none"> • Búsqueda de páginas web de turismo. • Selección y descarga de datos de turismo. • Procesamiento de datos de turismo. • Análisis de contenido web de turismo. <p>Prototipos:</p> <ul style="list-style-type: none"> • Organizaciones de turismo. • Organizaciones de marketing. • Ciudades inteligentes.

4.4.4.1.5 Crear el Backlog Priorizado del Producto

En esta fase con base en las épicas se genera la lista de producto priorizado para establecer el orden de actividades a realizar de acuerdo al orden de importancia y establecer fechas estimadas de finalización.

Entrada	Herramientas	Salidas
<p>1.- Equipo principal de Scrum</p> <p>2.- Épicas</p> <p>3.- Prototipos</p>	<p>1.- Métodos de priorización de historias de usuario</p>	<p>1.- Backlog priorizado del producto</p>

Reunión de un día para crear el Product Backlog priorizado con base a las épicas redactadas.

Product Backlog priorizado						
ID	Épica	Historias de Usuario	Prioridad	Estado	Estimacion	Sprint
E1	Búsqueda de páginas web de turismo.	HU01. Estructurar páginas web. HU02. Seleccionar páginas web.	A	No iniciado	20 Días	Uno
E2	Selección y descarga de datos de turismo.	HU01. Configurar paginas web. HU02. Descarga de páginas web.	B	No iniciado	20 Días	Dos
E3	Procesamiento de datos de turismo.	HU01. Extraer datos de paginas web. HU02. Organizar datos páginas web.	C	No iniciado	20 Días	Tres
E4	Análisis de contenido web de turismo.	HU01. Configurar datos de paginas web. HU02. Categorización de palabra clave.	D	No iniciado	20 Días	Cuatro

Criterios de terminado

- E1: Páginas web seleccionadas con estructura de promoción turística.
- E2: Páginas web configuradas para ser descargadas.
- E3: Páginas web organizadas con datos referentes a turismo.
- E4: Páginas web analizadas de acuerdo a categorización de datos referentes a turismo.

4.4.4.1.6 Realizar la planificación del lanzamiento

En esta fase final del proceso de inicio de Scrum se realiza el cronograma general del total de historias de usuario, así como cronogramas específicos por cada uno de los sprints a realizar.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Stakeholders 3.- Declaración de la visión del proyecto 4.- Backlog priorizado del producto 5.- Criterios de terminado	1.- Sesiones de planificación del lanzamiento. 2.- Métodos de priorización del lanzamiento	1.- Cronograma de planificación del lanzamiento

Épica 1. Búsqueda de páginas web de turismo.

Reunión de cuatro días, en el primer día con base en las épicas descritas previamente, se establece el cronograma de planificación de tareas del lanzamiento del primer sprint.

- **Historia de usuario 1:** Estructurar páginas web.
- **Historia de usuario 2:** Seleccionar páginas web.

Cronograma Sprint 1						
2019 Noviembre						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
				1 E1 -HU1(T1): Seleccionar páginas web de turismo como fuentes de información.	2	3
4 E1 -HU1(T1): Seleccionar páginas web de turismo como fuentes de información.	5 E1 -HU1(T1): Seleccionar páginas web de turismo como fuentes de información.	6 E1 -HU1(T1): Seleccionar páginas web de turismo como fuentes de información.	7 E1 -HU1(T1): Seleccionar páginas web de turismo como fuentes de información.	8 E1 -HU1(T2): Descripción Consultar cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.	9	10
11 E1 -HU1(T2): Descripción Consultar cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.	12 E1 -HU1(T2): Descripción Consultar cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.	13 E1 -HU1(T2): Descripción Consultar cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.	14 E1 -HU1(T2): Descripción Consultar cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.	15 E1 -HU2(T3): Seleccionar los sitios web ubicados en la primera posición con más de 100 entradas.	16	17
18 E1 -HU2(T3): Seleccionar los sitios web ubicados en la primera posición con más de 100 entradas.	19 E1 -HU2(T3): Seleccionar los sitios web ubicados en la primera posición con más de 100 entradas.	20 E1 -HU2(T3): Seleccionar los sitios web ubicados en la primera posición con más de 100 entradas.	21 E1 -HU2(T3): Seleccionar los sitios web ubicados en la primera posición con más de 100 entradas.	22 E1 -HU2(T4): Clasificar los sitios web preseleccionados.	23	24
25 E1 -HU2(T4): Clasificar los sitios web preseleccionados.	26 E1 -HU2(T4): Clasificar los sitios web preseleccionados.	27 E1 -HU2(T4): Clasificar los sitios web preseleccionados.	28 E1 -HU2(T4): Clasificar los sitios web preseleccionados.	29 E2 -HU3(T5): Identificar los hipervinculos de las páginas web relevantes.	30	

Épica 2. Búsqueda de páginas web de turismo.

Reunión de cuatro días, en el segundo día con base en las épicas descritas se establece el cronograma de planificación de tareas del lanzamiento del segundo sprint.

- **Historia de usuario 3:** Configurar páginas web.
- **Historia de usuario 4:** Descargar páginas web.

Cronograma Sprint 2						
2019 Diciembre						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
						1
2 E2 -HU3(T5): Identificar los hipervínculos de las páginas web relevantes.	3 E2 -HU3(T5): Identificar los hipervínculos de las páginas web relevantes.	4 E2 -HU3(T5): Identificar los hipervínculos de las páginas web relevantes.	5 E2 -HU3(T5): Identificar los hipervínculos de las páginas web relevantes.	6 E2 -HU3(T6): Descargar archivos con filtro de nivel y tipo de archivo.	7	8
9 E2 -HU3(T6): Descargar archivos con filtro de nivel y tipo de archivo.	10 E2 -HU3(T6): Descargar archivos con filtro de nivel y tipo de archivo.	11 E2 -HU3(T6): Descargar archivos con filtro de nivel y tipo de archivo.	12 E2 -HU3(T6): Descargar archivos con filtro de nivel y tipo de archivo.	13 E2 -HU4(T7): Descargar archivos con filtro URL y de contenido.	14	15
16 E2 -HU4(T7): Descargar archivos con filtro URL y de contenido.	17 E2 -HU4(T7): Descargar archivos con filtro URL y de contenido.	18 E2 -HU4(T7): Descargar archivos con filtro URL y de contenido.	19 E2 -HU4(T7): Descargar archivos con filtro URL y de contenido.	20 E2 -HU4(T8): Descargar las páginas web utilizando la copiadora web Offline Explorer Enterprise	21	22
23 E2 -HU4(T8): Descargar las páginas web utilizando la copiadora web Offline Explorer Enterprise	24 E2 -HU4(T8): Descargar las páginas web utilizando la copiadora web Offline Explorer Enterprise	25 DESCANSO	26 E2 -HU4(T8): Descargar las páginas web utilizando la copiadora web Offline Explorer Enterprise	25 E2 -HU4(T8): Descargar las páginas web utilizando la copiadora web Offline Explorer Enterprise	28	29
30 E3 -HU5(T9): Extraer de los contenidos de la página web, segmentar por nacionalidad e idioma.	31 E3 -HU5(T9): Extraer de los contenidos de la página web, segmentar por nacionalidad e idioma.					

Épica 3. Búsqueda de páginas web de turismo.

Reunión de cuatro días, en el tercer día con base en las épicas descritas se establece el cronograma de planificación de tareas del lanzamiento del tercer sprint.

- **Historia de usuario 5:** Extraer datos de páginas web.
- **Historia de usuario 6:** Organizar datos de páginas web.

Cronograma Sprint 3						
2020 Enero						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
		1 DESCANSO	2 E3 -HU5(T9): Extraer de los contenidos de la página web, segmentar por nacionalidad e idioma.	3 E3 -HU5(T9): Extraer de los contenidos de la página web, segmentar por nacionalidad e idioma.	4	5
6 E3 -HU5(T9): Extraer de los contenidos de la página web, segmentar por nacionalidad e idioma.	7 E3 -HU5(T10): Organizar los datos con la estructura de los archivos.	8 E3 -HU5(T10): Organizar los datos con la estructura de los archivos.	9 E3 -HU5(T10): Organizar los datos con la estructura de los archivos.	10 E3 -HU5(T10): Organizar los datos con la estructura de los archivos.	11	12
13 E3 -HU5(T10): Organizar los datos con la estructura de los archivos.	14 E3 -HU6(T11): Preservar contenido web generado por el usuario con el software Microsoft Expression Web 4.	15 E3 -HU6(T11): Preservar contenido web generado por el usuario con el software Microsoft Expression Web 4.	16 E3 -HU6(T11): Preservar contenido web generado por el usuario con el software Microsoft Expression Web 4.	17 E3 -HU6(T11): Preservar contenido web generado por el usuario con el software Microsoft Expression Web 4.	18	19
20 E3 -HU6(T11): Preservar contenido web generado por el usuario con el software Microsoft Expression Web 4.	21 E3 -HU6(T12): Depurar nombres que afectan destinos y atracciones.	22 E3 -HU6(T12): Depurar nombres que afectan destinos y atracciones.	23 E3 -HU6(T12): Depurar nombres que afectan destinos y atracciones.	24 E3 -HU6(T12): Depurar nombres que afectan destinos y atracciones.	25	26
27 E3 -HU6(T12): Depurar nombres que afectan destinos y atracciones.	28 E4 -HU7(T13): Analizar el contenido de forma cuantitativa con con delimitador lista negra.	29 E4 -HU7(T13): Analizar el contenido de forma cuantitativa con con delimitador lista negra.	30 E4 -HU7(T13): Analizar el contenido de forma cuantitativa con con delimitador lista negra.	31 E4 -HU7(T13): Analizar el contenido de forma cuantitativa con con delimitador lista negra.		

Épica 4. Búsqueda de páginas web de turismo.

Reunión de cuatro días, en el cuarto día con base en las épicas descritas se establece el cronograma de planificación de tareas del lanzamiento del cuarto sprint.

- **Historia de usuario 7:** Configurar datos de páginas web.
- **Historia de usuario 8:** Categorización de palabras clave de páginas web.

Cronograma Sprint 4						
2020 Febrero						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
					1	2
3 E4 -HU7(T13): Analizar el contenido de forma cuantitativa con delimitador lista negra.	4 E4 -HU7(T14): Analizar el contenido de forma cuantitativa con delimitador lista de palabras compuestas.	5 E4 -HU7(T14): Analizar el contenido de forma cuantitativa con delimitador lista de palabras compuestas.	6 E4 -HU7(T14): Analizar el contenido de forma cuantitativa con delimitador lista de palabras compuestas.	7 E4 -HU7(T14): Analizar el contenido de forma cuantitativa con delimitador lista de palabras compuestas.	8	9
10 E4 -HU7(T14): Analizar el contenido de forma cuantitativa con delimitador lista de palabras compuestas.	11 E4 -HU8(T15): Analizar el contenido de forma cuantitativa con delimitador peso de la palabra clave.	12 E4 -HU8(T15): Analizar el contenido de forma cuantitativa con delimitador peso de la palabra clave.	13 E4 -HU8(T15): Analizar el contenido de forma cuantitativa con delimitador peso de la palabra clave.	14 E4 -HU8(T15): Analizar el contenido de forma cuantitativa con delimitador peso de la palabra clave.	15	16
17 E4 -HU8(T15): Analizar el contenido de forma cuantitativa con delimitador peso de la palabra clave.	18 E4 -HU8(T16): Agrupar las palabras clave en categorías relacionadas con factores de atracción turística.	19 E4 -HU8(T16): Agrupar las palabras clave en categorías relacionadas con factores de atracción turística.	20 E4 -HU8(T16): Agrupar las palabras clave en categorías relacionadas con factores de atracción turística.	21 E4 -HU8(T16): Agrupar las palabras clave en categorías relacionadas con factores de atracción turística.	22	23
24 E4 -HU8(T16): Agrupar las palabras clave en categorías relacionadas con factores de atracción turística.	25	26	27	28	29	

En el proceso **Planeación y estimación** de Scrum se crean las historias de usuario con sus respectivas tareas, se estima el tiempo en se van a realizar y se compromete a las personas encargadas de realizarlas. Finalmente se crea el Sprint Backlog y se lleva un registro de la evolución de las historias de usuario a través de la gráfica Burn-up Chart.

4.4.4.2 Proceso 2: Planeación y estimación

4.4.4.2.1 Crear historias de usuario

En esta fase inicial de la planeación, se redacta la descripción y validación de todas las historias de usuario a desarrollar durante el proyecto de análisis Big Data para turismo de la ciudad de Barcelona.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Backlog priorizado del producto 3.- Criterios de terminado 4.- Prototipos	1.- Experiencia en la redacción de historias de usuario	1.- Historias de usuario 2.- Criterios de aceptación de historias de usuario

Historias de usuario:

Historia de usuario	
Id	HU1.
Nombre	Estructurar páginas web.
Prioridad	Alta.
Riesgo	Alta.
Descripción	Como miembro del equipo del equipo de desarrollo, quiero seleccionar fuentes de información UGC (User Generated Content) en área de turismo para tener los datos recopilados en forma sistemática.
Validación	Páginas web de turismo estructuradas.

Historia de usuario	
Id	HU2.
Nombre	Seleccionar páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero seleccionar las páginas web de turismo ubicadas en la primera posición que cumplan los criterios de estructura.
Validación	Páginas web de turismo seleccionadas.

Historia de usuario	
Id	HU3.
Nombre	Configurar páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero aplicar filtros a las páginas web de turismo para no sobrecargar el tráfico de alojamiento web.
Validación	Páginas web de turismo con filtros de descarga.

Historia de usuario	
Id	HU4.
Nombre	Descargar páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero descargar las páginas web de turismo relacionadas con el caso de estudio para iniciar el procesamiento y análisis de datos.
Validación	Páginas web de turismo descargadas.

Historia de usuario	
Id	HU5.
Nombre	Extraer datos de páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero extraer de las páginas web de turismo datos del usuario y el idioma en que están escritos para poder clasificar los datos.
Validación	Datos de usuario e idioma en que están escritos.

Historia de usuario	
Id	HU6.
Nombre	Organizar datos de páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero organizar los datos de las páginas web de turismo siguiendo una estructura paralela a la del servidor web para identificar y clasificar los blogs de viajes.
Validación	Datos de páginas web de turismo organizados.

Historia de usuario	
Id	HU7.
Nombre	Configurar datos de páginas web.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero configurar los datos de las páginas web de turismo con delimitadores y zonas de análisis para categorizar por palabras clave.
Validación	Páginas web de turismo delimitadas.

Historia de usuario	
Id	HU8.
Nombre	Categorización de palabras clave.
Prioridad	Alta
Riesgo	Alta
Descripción	Como miembro del equipo de desarrollo, quiero agrupar en categorías las palabras clave de las páginas web de turismo para obtener información y conocimiento de los datos
Validación	Lista de categorías de palabras clave.

4.4.4.2 Estimar historias de usuario

En esta fase los integrantes del equipo de desarrollo de acuerdo a su experiencia, realizan la estimación de tiempo para concluir cada una de las historias de usuario con la técnica Planning Poker utilizada por primera vez en el método ágil Xtreme Programming.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Historias de usuario	1.- Métodos de estimación	1.- Historias de usuario estimadas

Historias del usuario estimadas por el equipo de desarrollo de cuatro integrantes:

Rango de valores de Planning Poker

0 = Horas 1 = día 2 = días 3 = días 5 = días
8 = días 13 = días 21 = días 30 = días 50 = Meses

Historia de Usuario	Miembro				Estimación media	Prioridad
	A	B	C	D		
HU01	10	10	8	13	10.20	Alta
HU02	10	10	8	13	10.20	Alta
HU03	10	10	13	8	10.20	Alta
HU04	10	10	13	8	10.20	Alta
HU05	10	8	13	10	10.20	Alta
HU06	10	10	10	13	10.75	Alta
HU07	10	10	10	13	10.75	Alta
HU08	10	10	10	13	10.75	Alta

4.4.4.2.3 Comprometer historias de usuario estimadas.

En esta fase, se comprometen las historias de usuario con un rol responsable, para este caso de estudio todas las historias de usuario son realizadas por el equipo de desarrollo.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Historias de usuario estimadas 3.- Duración del sprint	1.- Reuniones de planificación del sprint	1.- Historias de usuario comprometidas

Historias de usuario estimadas:

Historia de Usuario	Historia de usuario	Sprint	Responsable
HU01	Como miembro del equipo del equipo de desarrollo, quiero seleccionar fuentes de información UGC (User Generated Content) en área de turismo para tener los datos recopilados en forma sistemática.	1	Equipo Scrum
HU02	Como miembro del equipo de desarrollo, quiero seleccionar las páginas web de turismo ubicadas en la primera posición que cumplan los criterios de estructura.	1	Equipo Scrum
HU03	Como miembro del equipo de desarrollo, quiero aplicar filtros a las páginas web de turismo para no sobrecargar el tráfico de alojamiento web.	2	Equipo Scrum
HU04	Como miembro del equipo de desarrollo, quiero descargar las páginas web de turismo relacionadas con el caso de estudio para iniciar el procesamiento y análisis de datos.	2	Equipo Scrum
HU05	Como miembro del equipo de desarrollo, quiero extraer de las páginas web de turismo datos del usuario y el idioma en que están escritos para poder clasificar los datos.	3	Equipo Scrum
HU06	Como miembro del equipo de desarrollo, quiero organizar los datos de las páginas web de turismo siguiendo una estructura paralela a la del servidor web para identificar y clasificar los blogs de viajes.	3	Equipo Scrum
HU07	Como miembro del equipo de desarrollo, quiero configurar los datos de las páginas web de turismo con delimitadores y zonas de análisis para categorizar por palabras clave.	4	Equipo Scrum
HU08	Como miembro del equipo de desarrollo, quiero agrupar en categorías las palabras clave de las páginas web de turismo para obtener información y conocimiento de los datos	4	Equipo Scrum

4.4.4.2.4 Identificar tareas

En esta fase, se describen y enlistan cada una de las tareas que tienen las historias de usuario, para este caso de estudio, todas las historias de usuario son realizadas por el equipo de desarrollo.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Historias de usuario comprometidas	1.- Reuniones de planificación del sprint	1.- Lista de tareas

Lista de tareas:

Tarea	T01.
Historia de usuario	HU01.
Estado	No iniciada, En progreso, Completada.
Descripción	Seleccionar páginas web de turismo como fuentes de información con los atributos post-título, destino y fecha de viaje, país de origen del autor del diario de viaje y tema al que se refieren (cosas que hacer, factor de atracción, hotel, restaurante, etc.)

Tarea	T02.
Historia de usuario	HU01.
Estado	No iniciada.
Descripción	Consultar en los motores de búsqueda BlogSearch.Google.com y Bing.com con las cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.

Tarea	T03.
Historia de usuario	HU02.
Estado	No iniciada.
Descripción	Seleccionar los sitios web ubicados en la primera posición que contengan más de 100 entradas relacionadas con el estudio de caso.

Tarea	T04.
Historia de usuario	HU02.
Estado	No iniciada.
Descripción	Clasificar los sitios web preseleccionados utilizando una fórmula ponderada: $TBRH = 1*B(V) + 1*B(P) + 2*B(S)$. Donde 'B' es el método de pedido de Borda, 'V' es la visibilidad del sitio web (cantidad y calidad de enlaces entrantes), 'P' su popularidad (visitantes, visitas y tráfico en general) y 'S' el tamaño (número de entradas relacionadas con el estudio de caso).

Tarea	T05.
Historia de usuario	HU03.
Estado	No iniciada.
Descripción	Identificar los hipervínculos de las páginas web relevantes relacionadas con el caso de estudio (blogs de viajes, páginas de viajes, diarios de viaje y reseñas de viajes sobre Barcelona y otros destinos en los alrededores).

Tarea	T06.
Historia de usuario	HU03.
Estado	No iniciada.
Descripción	<p>Descargar la cantidad mínima posible de archivos, para no sobrecargar el tráfico de alojamiento web y ahorrar espacio en el disco duro local. Para este propósito, se aplicaron filtros:</p> <ul style="list-style-type: none"> • De nivel, para definir la profundidad de la búsqueda de documentos HTML. En otras palabras, el nivel es el número de clics del mouse en los hipervínculos que son necesarios para pasar de la página de inicio a la última página necesaria. • Tipo de archivo para permitir restringir la descarga por extensión de nombre de archivo.

Tarea	T07.
Historia de usuario	HU04.
Estado	No iniciada.
Descripción	<p>Descargar la cantidad mínima posible de archivos, para no sobrecargar el tráfico de alojamiento web y ahorrar espacio en el disco duro local. Para este propósito, se aplicaron filtros:</p> <ul style="list-style-type: none"> • URL permite realizar en cualquier parte del mismo: protocolo (http, https, ftp, etc.), servidor, dominio, subdirectorios (carpetas) y nombres de archivo. También permite actuar en cualquier segmento de la URL utilizando las palabras clave incluidas y excluidas. • Filtros de contenido permiten al usuario especificar las palabras clave para buscar en todo el texto de las páginas descargadas con la desventaja de tener que verificar si el archivo contiene las palabras clave que buscó antes de procesarlo.

Tarea	T08.
Historia de usuario	HU04.
Estado	No iniciada.
Descripción	Descargar las páginas web relacionadas con el estudio de caso utilizando la copiadora web Offline Explorer Enterprise (OEE), porque tiene la capacidad de descargar hasta 100 millones de URL por proyecto.

Tarea	T09.
Historia de usuario	HU05.
Estado	No iniciada.
Descripción	Extraer información útil de los contenidos de la página web, segmentar por nacionalidad y detectar lenguaje con programa Java basado en la Biblioteca de detección de idiomas.

Tarea	T10.
Historia de usuario	HU05.
Estado	No iniciada.
Descripción	Organizar los datos con la estructura de los archivos con el siguiente patrón de carpetas y archivos: \ host \ brand \ town \ date_lang_isFrom_pageld_topic.htm Las barras diagonales inversas del patrón representan subdirectorios o carpetas, los guiones unen las palabras compuestas, y los guiones bajos separan las diferentes palabras o números del nombre del archivo.

Tarea	T11.
Historia de usuario	HU06.
Estado	No iniciada.
Descripción	Con el software Microsoft Expression Web 4 preservar la parte del contenido de la página web generada por el usuario sin modificar su formato HTML y eliminar todo lo demás.

Tarea	T12.
Historia de usuario	HU06.
Estado	No iniciada.
Descripción	Depurar los datos, ya que el idioma catalán utiliza caracteres especiales como cedillas y vocales con acentos que son problemáticos para la codificación en sitios web y desconcertantes para los usuarios de habla inglesa. Especialmente aquellos que afectan los nombres de destinos y atracciones y los nombres propios en general, como problemas de codificación de caracteres, errores ortográficos y traducciones.

Tarea	T13.
Historia de usuario	HU07.
Estado	No iniciada.
Descripción	Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'lista negra' para incluir palabras insignificantes que ayudan a construir oraciones pero no representan ningún contenido, como algunos adverbios, artículos, preposiciones, pronombres y conjunciones.

Tarea	T14.
Historia de usuario	HU07.
Estado	No iniciada.
Descripción	Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'lista de palabras compuestas' que consiste en grupos de palabras. Por ejemplo, la 'Basílica de la Sagrada Familia' conserva la 'de' y la 'la' a pesar de estar incluida en la lista negra, y tiene preferencia sobre las palabras clave simples 'Basílica', 'Sagrada' y 'Familia'.

Tarea	T15.
Historia de usuario	HU08.
Estado	No iniciada.
Descripción	Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'peso de la palabra clave' para indicar la visibilidad de la palabra clave o frase clave. Por ejemplo, en general, se ha demostrado en estudios sobre flujo de clics y patrones de navegación que el contenido más difundido es el que se encuentra en la parte superior de una página web.

Tarea	T16.								
Historia de usuario	HU08.								
Estado	No iniciada.								
Descripción	<p>Agrupar las palabras clave en las siguientes categorías relacionadas con factores de atracción para analizar el componente cognitivo de la imagen de destino:</p> <table border="0"> <tr> <td>-- Comida y Vino</td> <td>--Patrimonio inmaterial</td> </tr> <tr> <td>-- Ocio y recreación</td> <td>--Naturaleza y turismo activo</td> </tr> <tr> <td>-- Deportes</td> <td>--Sol, mar y arena</td> </tr> <tr> <td>-- Patrimonio tangible</td> <td>--Medio ambiente urbano</td> </tr> </table>	-- Comida y Vino	--Patrimonio inmaterial	-- Ocio y recreación	--Naturaleza y turismo activo	-- Deportes	--Sol, mar y arena	-- Patrimonio tangible	--Medio ambiente urbano
-- Comida y Vino	--Patrimonio inmaterial								
-- Ocio y recreación	--Naturaleza y turismo activo								
-- Deportes	--Sol, mar y arena								
-- Patrimonio tangible	--Medio ambiente urbano								

4.4.4.2.5 Estimar tareas

En esta fase los integrantes del equipo de desarrollo de acuerdo a su experiencia, realizan la estimación de tiempo para concluir cada una de las tareas pertenecientes a cada historia de usuario con la técnica Planning Poker utilizada por primera vez en el método ágil Xtreme Programming.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Lista de tareas	1.- Reuniones de planificación del sprint 2.- Criterios de estimación 3.- Métodos de estimación	1.- Lista de tareas estimadas

Lista de tareas estimadas:

Rango de valores de Planning Poker por semana.

0 = Horas 1 = día 3 = días 5 = días
 8 = días 13 = días 21 = días 40 = Meses

Tareas	Miembro				Estimación media
	A	B	C	D	
T01	5	8	4	5	4.7
T02	5	5	8	4	4.7
T03	5	5	5	8	5.7
T04	5	5	5	8	5.7
T05	5	5	8	5	5.7
T06	5	5	8	5	5.7
T07	5	5	5	4	4.7
T08	5	5	5	4	4.7
T09	5	5	5	4	4.7
T10	5	5	5	4	4.7
T11	5	5	8	5	5.7
T12	5	5	8	5	5.7
T13	5	5	5	5	5
T14	5	5	5	5	5
T15	5	5	5	5	5
T16	5	5	5	5	5

4.4.4.2.6 Crear el Sprint Backlog.

En esta fase, se crea en Sprint Backlog en el cual se describen cada una de las tareas pertenecientes a las historias de usuario y la correspondencia con los sprints en los que se desarrollan dichas tareas.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Lista de tareas estimadas 3.- Duración de sprints	1.- Reuniones de planificación del sprint	1.- Sprint Backlog

Sprints Backlog:

Sprint 1			
Id	Historia de usuario	Estimación	Tareas
HU01	Como miembro del equipo de desarrollo, quiero seleccionar fuentes de información UGC (User Generated Content) en área de turismo para tener los datos recopilados en forma sistemática.	10.20	<p>T01 Seleccionar páginas web de turismo como fuentes de información con los atributos post-título, destino y fecha de viaje, país de origen del autor del diario de viaje y tema al que se refieren (cosas que hacer, factor de atracción, hotel, restaurante, etc.)</p> <p>T02 Consultar en los motores de búsqueda BlogSearch.Google.com y Bing.com con las cadenas de búsqueda 'blog de viajes' O 'revisión de viajes'.</p>
HU02	Como miembro del equipo de desarrollo, quiero seleccionar las páginas web de turismo ubicadas en la primera posición que cumplan los criterios de estructura.	10.20	<p>T03 Seleccionar los sitios web ubicados en la primera posición que contengan más de 100 entradas relacionadas con el estudio de caso.</p> <p>T04 Clasificar los sitios web preseleccionados utilizando una fórmula ponderada: $TBRH = 1*B(V) + 1*B(P) + 2*B(S)$. Donde 'B' es el método de pedido de Borda, 'V' es la visibilidad del sitio web (cantidad y calidad de enlaces entrantes), 'P' su popularidad (visitantes, visitas y tráfico en general) y 'S' el tamaño (número de entradas relacionadas con el estudio de caso).</p>

Sprint 2			
Id	Historia de usuario	Estimación	Tareas
HU03	Como miembro del equipo de desarrollo, quiero aplicar filtros a las páginas web de turismo para no sobrecargar el tráfico de alojamiento web.	10.20	<p>T05 Identificar los hipervínculos de las páginas web relevantes relacionadas con el estudio de caso (blogs de viajes, páginas de viajes, diarios de viaje y reseñas de viajes sobre Barcelona y otros destinos en los alrededores).</p> <p>T06 Descargar la cantidad mínima posible de archivos, para no sobrecargar el tráfico de alojamiento web y ahorrar espacio en el disco duro local. Para este propósito, se aplicaron filtros:</p> <ul style="list-style-type: none"> De nivel, para definir la profundidad de la búsqueda de documentos HTML. En otras palabras, el nivel es el número de clics del mouse en los hipervínculos que son necesarios para pasar de la página de inicio a la última página necesaria.

			<ul style="list-style-type: none"> Tipo de archivo para permitir restringir la descarga por extensión de nombre de archivo.
HU04	Como miembro del equipo de desarrollo, quiero descargar las páginas web de turismo relacionadas con el caso de estudio para iniciar el procesamiento y análisis de datos.	10.20	<p>T07 Descargar la cantidad mínima posible de archivos, para no sobrecargar el tráfico de alojamiento web y ahorrar espacio en el disco duro local. Para este propósito, se aplicaron filtros:</p> <ul style="list-style-type: none"> URL permite realizar en cualquier parte del mismo: protocolo (http, https, ftp, etc.), servidor, dominio, subdirectorios (carpetas) y nombres de archivo. También permite actuar en cualquier segmento de la URL utilizando las palabras clave incluidas y excluidas. Filtros de contenido permiten al usuario especificar las palabras clave para buscar en todo el texto de las páginas descargadas con la desventaja de tener que verificar si el archivo contiene las palabras clave que buscó antes de procesarlo. <p>T08 Descargar las páginas web relacionadas con el estudio de caso, utilizando la copiadora web Offline Explorer Enterprise (OEE), porque tiene la capacidad de descargar hasta 100 millones de URL por proyecto.</p>

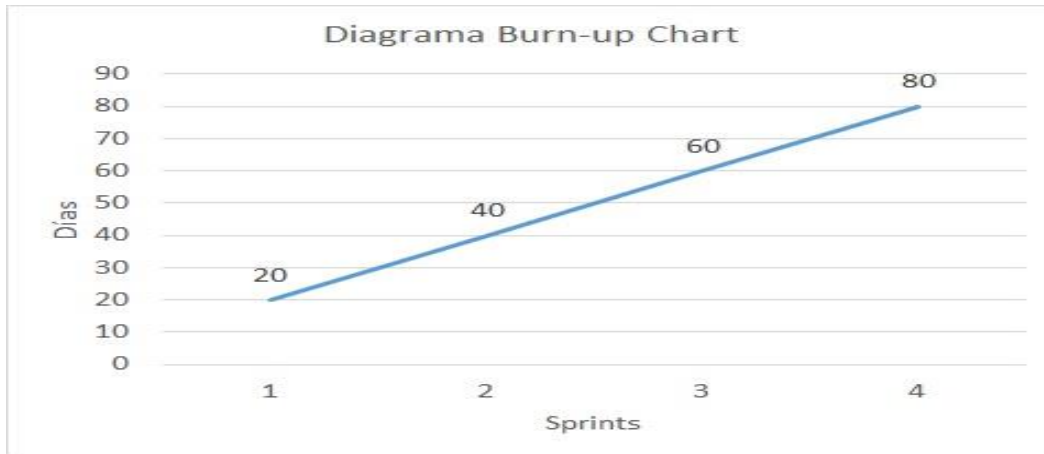
Sprint 3			
Id	Historia de usuario	Estimación	Tareas
HU05	Como miembro del equipo de desarrollo, quiero extraer de las páginas web de turismo datos del usuario y el idioma en que están escritos para poder clasificar los datos.	10.20	<p>T09 Extraer información útil de los contenidos de la página web, segmentar por nacionalidad y detectar lenguaje con programa Java basado en la Biblioteca de detección de idiomas.</p> <p>T10 Organizar los datos con la estructura de los archivos con el siguiente patrón de carpetas y archivos: \ host \ brand \ town \ date_lang_isFrom_pageld_topic.htm Las barras diagonales inversas del patrón representan subdirectorios o carpetas, los guiones unen las palabras compuestas, y los guiones bajos separan las diferentes palabras o números del nombre del archivo.</p>
HU06	Como miembro del equipo de desarrollo, quiero organizar los datos de las páginas web de turismo siguiendo una estructura paralela a la del servidor web para	10.75	<p>T11 Con el software Microsoft Expression Web 4 preservar la parte del contenido de la página web generada por el usuario sin modificar su formato HTML y eliminar todo lo demás.</p> <p>T12 Depurar los datos, ya que el idioma catalán utiliza caracteres especiales como cedillas y vocales con acentos que son problemáticos para la codificación en</p>

	identificar y clasificar los blogs de viajes.		sitios web y desconcertantes para los usuarios de habla inglesa. Especialmente aquellos que afectan los nombres de destinos y atracciones y los nombres propios en general, como problemas de codificación de caracteres, errores ortográficos y traducciones.
--	---	--	--

Sprint 4			
Id	Historia de usuario	Estimación	Tareas
HU07	Como miembro del equipo de desarrollo, quiero configurar los datos de las páginas web de turismo con delimitadores y zonas de análisis para categorizar por palabras clave.	10.75	<p>T13 Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'lista negra' para incluir palabras insignificantes que ayudan a construir oraciones, pero no representan ningún contenido, como algunos adverbios, artículos, preposiciones, pronombres y conjunciones.</p> <p>T14 Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'lista de palabras compuestas' que consiste en grupos de palabras. Por ejemplo, la 'Basílica de la Sagrada Familia' conserva la 'de' y la 'la' a pesar de estar incluida en la lista negra, y tiene preferencia sobre las palabras clave simples 'Basílica', 'Sagrada' y 'Familia'.</p>
HU08	Como miembro del equipo de desarrollo, quiero agrupar en categorías las palabras clave de las páginas web de turismo para obtener información y conocimiento de los datos	10.75	<p>T15 Analizar el contenido de forma cuantitativa a través del analizador de contenido del sitio web, Site Content Analyzer (SCA) y utilizar el delimitador 'peso de la palabra clave' para indicar la visibilidad de la palabra clave o frase clave. Por ejemplo, en general, se ha demostrado en estudios sobre flujo de clics y patrones de navegación que el contenido más difundido es el que se encuentra en la parte superior de una página web.</p> <p>T16 Agrupar las palabras clave en las siguientes categorías relacionadas con factores de atracción para analizar el componente cognitivo de la imagen de destino:</p> <ul style="list-style-type: none"> ▪ Comida y Vino ▪ Deportes ▪ Medio ambiente urbano ▪ Naturaleza y turismo activo ▪ Ocio y recreación ▪ Patrimonio inmaterial ▪ Patrimonio tangible ▪ Sol, mar y arena

Sprint Burndown Chart: muestra la cantidad de días en que van finalizando los sprints.

- 1º Sprint termina a los 20 días de iniciado el proyecto.
- 2º Sprint termina a los 40 días de iniciado el proyecto.
- 3º Sprint termina a los 60 días de iniciado el proyecto.
- 4º Sprint termina a los 80 días de iniciado el proyecto.



En el proceso **Implementación** de Scrum por cada uno de los sprints realizados, se actualiza el Scrumboard con base en el tablero Kanban y el Impediment Log a través de las reuniones diarias de quince minutos para ir refinando al Backlog Priorizado y la gráfica Burn-up Chart.

4.4.4.3 Proceso 3: Implementación

4.4.4.3.1 Crear entregables

En esta fase inicial del proceso de implementación, se establecen los criterios para avalar los entregables de cada sprint y así poder dar por concluido cada uno de los sprints que se implementan. Para realizar los sprints, se utiliza el tablero utilizado en el medido ágil Kanban.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Sprint Backlog 3.- Scrumboard 4.- Impediment log	1.- Experiencia del equipo	1.- Entregables del sprint

Entregables de sprints:

Sprint 1

Fuentes de información UGC (User Generated Content) en área de turismo con datos recopilados y lista de páginas web ubicadas en la primera posición que cumplan los criterios de estructura.

Sprint 2

Páginas web de turismo filtradas y descargadas relacionadas al caso de estudio.

Sprint 3

Páginas web de turismo extraídas con datos del usuario e idioma y organizadas con una estructura similar al servidor.

Sprint 4

Datos de páginas web configurados con delimitadores y palabras clave agrupadas en categorías.

Scrumboard actualizado:

Scrumboard 1º Sprint			
Historia de usuario	Para hacer	En progreso	Hecho
HU01			T01, T02
HU02			T03, T04
HU03		T05, T06	
HU04		T07, T08	
HU05		T09, T10	
HU06		T11, T12	
HU07	T13, T14		
HU08	T15, T16		

Scrumboard 2º Sprint			
Historia de usuario	Para hacer	En progreso	Hecho
HU01			T01, T02
HU02			T03, T04
HU03			T05, T06
HU04			T07, T08
HU05		T09, T10	
HU06		T11, T12	
HU07	T13, T14		
HU08	T15, T16		

Scrumboard 3º Sprint			
Historia de usuario	Para hacer	En progreso	Hecho
HU01			T01, T02
HU02			T03, T04
HU03			T05, T06
HU04			T07, T08
HU05			T09, T10
HU06			T11, T12
HU07		T13, T14	
HU08		T15, T16	

Scrumboard 4º Sprint			
Historia de usuario	Para hacer	En progreso	Hecho
HU01			T01, T02
HU02			T03, T04
HU03			T05, T06
HU04			T07, T08
HU05			T09, T10
HU06			T11, T12
HU07			T13, T14
HU08			T15, T16

Impediment log actualizado

- Falta de conocimiento de un miembro del equipo.
- No asiste un miembro del equipo por motivo de trabajo.
- Conocimiento de las herramientas de programación.
- Defectos de configuraciones.

4.4.4.3.2 Realizar Daily Standup

En esta fase, a la misma hora y lugar durante quince minutos se reúne el equipo de desarrollo para revisar sus avances e impedimentos para concluir sus actividades diarias y actualizar el diagrama Burn-up Chart.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum	1.- Daily Standup	1.- Sprint Burn-up actualizado
2.- Scrum Master	2.- Tres preguntas diarias	
3.- Sprint Burn-up Chart		
4.- Impediment log		

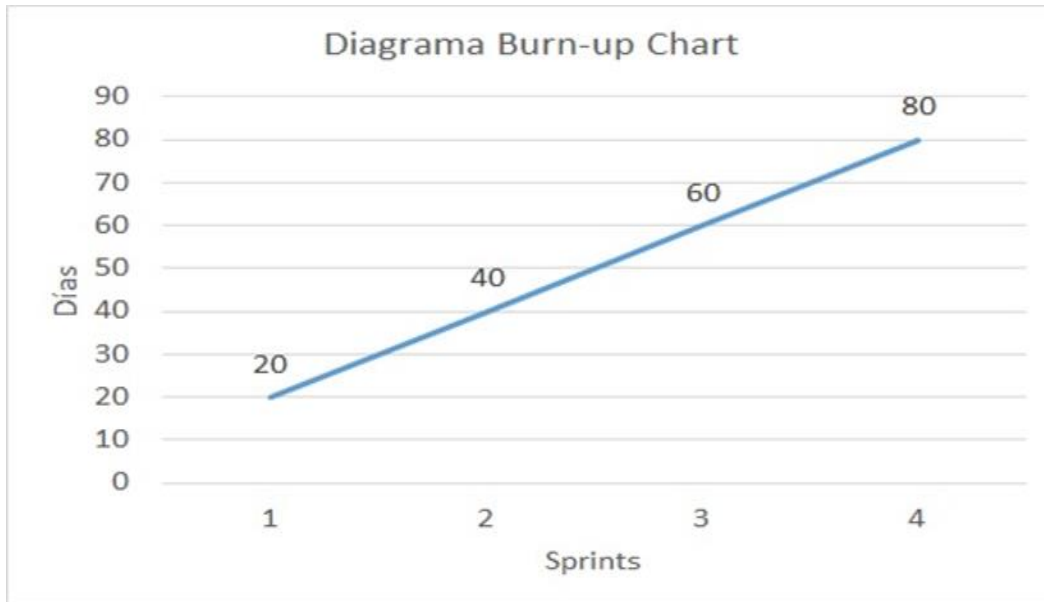
Preguntas realizadas al Equipo de desarrollo:

¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?

¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?

¿Qué impedimento evita que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?

Burn-up Chart



4.4.4.3 Refinar el Backlog Priorizado del Producto.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Backlog priorizado del producto	1.- Reunión de revisión del Backlog priorizado del producto	1.- Backlog priorizado del producto actualizado

Product Backlog priorizado						
ID	Épica	Historias de Usuario	Prioridad	Estado	Estimacion	Sprint
E1	Búsqueda de páginas web de turismo.	HU01. Estructurar páginas web. HU02. Seleccionar páginas web.	A	Iniciado	20 Días	Uno
E2	Selección y descarga de datos de turismo.	HU01. Configurar paginas web. HU02. Descarga de páginas web.	B	En progreso	20 Días	Dos
E3	Procesamiento de datos de turismo.	HU01. Extraer datos de paginas web. HU02. Organizar datos páginas web.	C	No iniciado	20 Días	Tres
E4	Análisis de contenido web de turismo.	HU01. Configurar datos de paginas web. HU02. Categorización de palabra clave.	D	No iniciado	20 Días	Cuatro

En el proceso **Revisión y retrospectiva** de Scrum, se valida que las historias de usuario hayan sido concluidas satisfactoriamente en cada uno de los sprints verificando la lista de entregables. Finalmente, se hace retrospectiva de cada sprint para encontrar elementos de mejora aplicables al siguiente sprint.

4.4.4.4 Proceso 4: Revisión y retrospectiva

4.4.4.4.1 Demostrar y validar el sprint

En la primera fase del proceso se demuestra y valida la terminación de cada uno de los sprints a través de la lista de entregables descritos con el resultado a obtener de cada una de las historias de usuario.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum 2.- Backlog priorizado del producto 3.- Sprint Backlog 4.- Criterios de terminado 5.- Criterios de aceptación de las historias de usuario	1.- Reuniones de revisión del sprint	1.- Entregables aceptados

Entregables aceptados:

Entregables		
Historia de usuario	Descripción	Épica
HU01	Páginas web de turismo estructuradas	Búsqueda de páginas web de turismo.
HU02	Páginas web de turismo seleccionada.	
HU03	Páginas web de turismo configuradas.	Selección y descarga de datos de turismo.
HU04	Páginas web de turismo descargadas.	
HU05	Datos de páginas web de turismo extraídos.	Procesamiento de datos de turismo.
HU06	Páginas web de turismo organizadas.	
HU07	Datos configurados de páginas web de turismo.	Análisis de contenido web de turismo.
HU08	Palabras clave de páginas web categorizadas.	

4.4.4.2 Retrospectiva del sprint

En esta fase, se realiza una reunión para analizar el proceso de cada sprint y realizar una lista con los aspectos que se pueden mejorar en el siguiente sprint.

Entrada	Herramientas	Salidas
1.- Scrum Master 2.- Equipo Scrum 3.- Demostrar y validar el sprint	1.- Reunión de retrospectiva del sprint	1.- Mejorar procesables acordadas

Elementos de mejora

- Buscar antecedentes de proyectos similares.
- Comprender el software utilizado.
- Perfeccionamiento de la Metodología Scrum.
- Perfeccionamiento de la estimación de tareas.

En el proceso **Lanzamiento** de Scrum, se realizan los acuerdos de entregables funcionales y el formulario de retrospectiva a través de reuniones en las cuales se redactan los documentos para asentar la realización de dichas actividades.

4.4.4.5 Proceso 5: Lanzamiento.

4.4.4.5.1 Enviar entregables

En esta fase, se redacta un documento en el cual se establecen los acuerdos de entrega de productos funcionales satisfactorios.

Entrada	Herramientas	Salidas
1.- Product Owner 2.- Stakeholders 3.- Entregables aceptados 4.- Cronograma de planificación del lanzamiento	1.- Método de desplazamiento organizacional	1.- Acuerdo de entregables funcionales

El acta de entrega-recepción se puede revisar en el Anexo E.

[Acuerdo de entregables funcionales.docx](#)

4.4.4.5.2 Retrospectiva del proyecto

En esta fase, se realiza una reunión entre Product Owner, Scrum Master y Equipo de desarrollo, para hacer retrospectiva del proyecto y establecer aspectos de mejora para siguientes proyectos similares.

Entrada	Herramientas	Salidas
1.- Equipo principal de Scrum	1.- Reunión de retrospectiva del proyecto	1.- Mejoras procesables acordadas 2.- Elementos de acción asignados y fechas límite

El formulario de la reunión de retrospectiva se puede revisar en el siguiente enlace (colocar como anexo):

[Reunión de retrospectiva.docx](#)

Capítulo 5. Conclusiones y Trabajo futuro

En este capítulo se abordan las conclusiones y trabajos futuros respecto a este trabajo de investigación.

5.1 Conclusiones

Los objetivos planteados para este trabajo de tesis se cumplieron de forma satisfactoria. La revisión de la literatura nos permitió dimensionar la trascendencia actual que los sistemas Big Data tienen en los procesos de negocio de múltiples empresas. Por lo tanto, es pertinente encontrar técnicas o metodologías que ayuden a reducir la incertidumbre de los sistemas Big Data. Durante la investigación se revisaron características generales de metodologías tradicionales de desarrollo de software tales como cascada, proceso unificado o modelo en espiral las cuales tienen enfoque predictivo, ciclos limitados, gestión controlada y planeación a largo plazo con dificultad para introducir cambios de requisitos durante el desarrollo. Por lo tanto, el enfoque planificado es difícil de adaptar en proyectos con requisitos dinámicos como los de desarrollo de sistemas Big Data.

Por otra parte, en la literatura también se revisaron los enfoques ágiles cuyas características generales son la capacidad de adaptación al cambio a través de iteraciones cortas e incrementales, dichas características permiten que el enfoque ágil se adapte a proyectos con desafíos dinámicos tales como los sistemas Big Data. Posteriormente con base en la investigación y mapeo de características de las metodologías ágiles, se concluye que la metodología ágil Scrum es la más utilizada por empresas que quieren integrar la agilidad en sus procesos de negocio.

Como complemento y apoyo a Scrum, en este trabajo de investigación se utilizó el elemento Plannig Poker de Extreme Programming para la estimación de tiempos respecto a historias de usuario y tareas. También se aplicó el tablero Kanban de la metodología homónima para visualizar y gestionar el flujo de tareas. Scrum tiene la capacidad de adaptarse a todo tipo de proyectos, integrar cambios de requisitos, realizar iteraciones cortas y colaboración constante entre cliente y equipo de trabajo para realizar entregas de valor continuo enfocados en los objetivos comerciales de las empresas.

En esta investigación se concluye que no se puede tomar una metodología universal, por lo tanto, es recomendable analizar las características del proyecto para determinar que metodología es conveniente implementar. A diferencia de las metodologías de desarrollo iterativo, Scrum presenta características más acordes para tratar con sistemas del tipo Big Data.

5.2 Trabajos futuros

En este trabajo se presenta una alternativa de cómo aplicar la Metodología Scrum en las etapas de un sistema Big Data. Como trabajos futuros se contempla ampliar la cantidad de mapeos con otros casos de estudio caracterizados en sistemas Big Data específicos por áreas, tales como: educación, deportes, economía o salud para aplicar Scrum y/o alguna otra metodología. Sistematizar a través de herramientas de software el proceso de mapeo entre las características ágiles y las de sistemas Big Data.

Referencias

- Abrahamsson, P., Salo, O., & Warsta, J. (2002). Agile Software Development Review and Analysis. *VTT Technical Research Center of Finland*, (478), 112. <https://doi.org/10.1076/csed.12.3.167.8613>
- Ahmad, G., Soomro, T. R., & Broh, M. N. (2014). Agile Methodologies: Comparative Study and Future Direction. *European Academic Research*, 1(11), 8826–8841. Retrieved from <http://euacademic.org/UploadArticle/273.pdf>
- Al-Jaroodi, J., Hollein, B., & Mohamed, N. (2017). Applying software engineering processes for big data analytics applications development. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*. <https://doi.org/10.1109/CCWC.2017.7868456>
- Al-Jaroodi, J., & Mohamed, N. (2017). Characteristics and requirements of big data analytics applications. *Proceedings - 2016 IEEE 2nd International Conference on Collaboration and Internet Computing, IEEE CIC 2016*, 426–432. <https://doi.org/10.1109/CIC.2016.062>
- Al-Zewairi, M., Biltawi, M., Etaiwi, W., & Shaout, A. (2017). Agile Software Development Methodologies: Survey of Surveys. *Journal of Computer and Communications*, 05(05), 74–97. <https://doi.org/10.4236/jcc.2017.55007>
- Ali, K. (2017). A Study of Software Development Life Cycle Process Models. *SSRN Electronic Journal*, 4(9), 92–95. <https://doi.org/10.2139/ssrn.2988291>
- Antonio Fernando Cruz Santos, I. P. T., & Ota'vivo Manoel Pereira Siqueira, and A. A. de O. (2018). *Big Data: A Systematic Review*. 558(October 2017), 501–506. <https://doi.org/10.1007/978-3-319-54978-1>
- Arndt, T. (2018). Big Data and software engineering: prospects for mutual enrichment. *Iran Journal of Computer Science*, 1(1), 3–10. <https://doi.org/10.1007/s42044-017-0003-0>
- Awad, M. A. (2019). Software Development Methodologies. *Project Management of Large Software-Intensive Systems*, 49–66. <https://doi.org/10.1201/9780429027932-4>
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from <http://agilemanifesto.org/>
- Cubana, R. C. (2017). Big Data: El Monstruo que pretende dominar el mundo. Retrieved from Revista Cultural Cubana website: <http://www.lajiribilla.cu/articulo/big-data-el-monstruo-que-pretende-dominar-el-mundo>
- Dharmapal, S. R., & Sikamani, K. T. (2016). Big data analytics using agile model. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 1088–1091. <https://doi.org/10.1109/ICEEOT.2016.7754854>
- Fernández-Manzano, E.-P., Neira, E., & Clares-Gavilán, J. (2016). Gestión de datos en el negocio audiovisual: Netflix como caso de estudio. *El Profesional de La Información*, 25(4), 568–576. <https://doi.org/10.3145/epi.2016.jul.06>
- Franková, P., Drahošová, M., & Balco, P. (2016). Agile Project Management Approach and its Use in Big Data Management. *Procedia Computer Science*, 83(Ant), 576–583. <https://doi.org/10.1016/j.procs.2016.04.272>
- Galgano, G. (2008). *Introducción a la metodología lean*. 28.
- Iacovelli, A., & Souveyet, C. (2008). Framework for agile methods classification. *CEUR Workshop*

- Proceedings*, 341, 91–102.
- John Gantz, D. R. (2013). THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. *IDC IView: IDC Analyze the Future, 2007*, 1–7.
- Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: Issues and challenges moving forward. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 995–1004. <https://doi.org/10.1109/HICSS.2013.645>
- Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: Issues, challenges, tools and Good practices. *2013 6th International Conference on Contemporary Computing, IC3 2013*, 404–409. <https://doi.org/10.1109/IC3.2013.6612229>
- Ladrero, I. (2018). Las 4 V's del Big Data. Retrieved from <https://www.baoss.es/las-4-vs-del-big-data/>
- Leon, M. (2017). ¿Qué son las Metodologías Ágiles? Retrieved from <https://www.coworkingesplugues.com/metodologias-agiles/>
- Madaan, A., Sharma, V., Pahwa, P., Das, P., & Sharma, C. (2018). Hadoop: Solution to unstructured data handling. *Advances in Intelligent Systems and Computing*, 654, 47–54. https://doi.org/10.1007/978-981-10-6620-7_6
- Madhavji, N. H., Miranskyy, A., & Kontogiannis, K. (2015). *Big Picture of Big Data Software Engineering*. 11–14. <https://doi.org/10.1109/BIGDSE.2015.10>
- Marine-Roig, E., & Anton Clavé, S. (2015). Tourism analytics with massive user-generated content: A case study of Barcelona. *Journal of Destination Marketing and Management*, 4(3), 162–172. <https://doi.org/10.1016/j.jdmm.2015.06.004>
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1–6. <https://doi.org/10.1145/2693208.2693233>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes Revista Multidisciplinaria de Investigación ISSN: 2550-6862*. <https://doi.org/10.1145/3120459.3120483>
- Mukherjee, S., & Shaw, R. (2016). Big Data- Concepts, Applications, Challenges and Future Scope. *International Journal of Advanced Research in Computer and Communication Engineering Vol., 5(2)*, 66–74. <https://doi.org/10.17148/IJARCC.2016.5215>
- Pérez, M. J. (2012). Guía Comparativa de Metodologías Ágiles. *Universidad de Valladolid*, 3–117. Retrieved from <https://uvadoc.uva.es/bitstream/10324/1495/1/TFG-B.117.pdf>
- Philip Chen, C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314–347. <https://doi.org/10.1016/j.ins.2014.01.015>
- Prodintec. (2010). Introducción al lean manufacturing. *Fundación PRODINTEC*, 77.
- Ríos, J. S. P. de los. (2015). *Implementación del método Kanban en Colombia*. 205.
- Saltz, J. S., & Shamsurhin, I. (2016). Big data team process methodologies: A literature review and the identification of key factors for a project's success. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2872–2879. <https://doi.org/10.1109/BigData.2016.7840936>
- Salvador, C., Nakasone, A., & Pow-Sang, J. A. (2014). A systematic review of usability techniques in agile methodologies. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2590651.2590668>
- Salza, P., Musmarra, P., & Ferrucci, F. (2019). Agile and Lean Concepts for Teaching and Learning. In *Agile and Lean Concepts for Teaching and Learning*. <https://doi.org/10.1007/978-981-13-2751-3>
- Sánchez, J. (2018). *Challenges of Traditional and Agile Software Processes*.
- Santos, M. Y., Oliveira e Sá, J., Andrade, C., Vale Lima, F., Costa, E., Costa, C., ... Galvão, J. (2017). A Big Data system supporting Bosch Braga Industry 4.0 strategy. *International Journal of Information Management*, 37(6), 750–760. <https://doi.org/10.1016/j.ijinfomgt.2017.07.012>
- SCRUMstudy. (2017). *Guía de conocimiento de Scrum*. Retrieved from www.scrumstudy.com
- Sharma, S. (2012). Agile Processes and Methodologies: A Conceptual Study. *International Journal on Computer Science and Engineering*, 4(5), 892–898.
- Singh, K. N., Behera, R. K., & Mantri, J. K. (2019). Big data ecosystem: Review on architectural

- evolution. *Advances in Intelligent Systems and Computing*, 813, 335–345.
https://doi.org/10.1007/978-981-13-1498-8_30
- Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, 70, 263–286.
<https://doi.org/10.1016/j.jbusres.2016.08.001>
- Solutions, L. (2019). 7 desperdicios, Mura, Muri, Muda. Retrieved from
<https://leansolutions.co/conceptos-lean/lean-manufacturing/7-desperdicios-mura-muri-muda-las-3-mu/>
- SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi, M. H. (2019). *Analysis of Software Development Methodologies*. 5(5). Retrieved from <http://msc-di.googlecode.com/svn-history/r10/trunk/tese.pdf>
- Srinivasu, M. A., Koushik, A., & Santhosh, E. B. (2017). Big Data Challenges and Solutions. *International Journal of Computer Sciences and Engineering*, 5(10), 250–255.
<https://doi.org/10.26438/ijcse/v5i10.250255>
- Tapias, D. (2014). *Escuela Politécnica Superior*.
- Tarwani, S., & Chug, A. (2016). Agile methodologies in software maintenance: A systematic review. *Informatica (Slovenia)*, 40(4), 415–426. <https://doi.org/10.31449/inf.v40i4.1182>
- TechAmerica, & Foundation. (2012). Demystifying Big Data. *Computing in Science and Engineering*, 13(6), 10–12. <https://doi.org/10.1109/MCSE.2011.99>
- Tjobod, V. E. (2015). Big data : A review Cjh E bu b ; B S f w jf x. *2013 International Conference on Collaboration Technologies and Systems (CTS)*, (May 2013), 42–47.
<https://doi.org/10.1109/CTS.2013.6567202>
- Version One. (2019). The 13th annual STATE OF AGILE(TM) Report. *CollabNet | VersionOne*, 13.
- Wells, J. D. (2009). The Rules of Extreme Programming. Retrieved from
<http://www.extremeprogramming.org/>
- Zicari, R. V. (n.d.). *Big Data: Challenges and Opportunities*. 103–128.

Anexo A: Metodología Lean

En este anexo, se detalla origen, características y funcionamiento de la Metodología Ágil Lean.

Lean

El término significa manufactura esbelta o ágil y es el nombre con que se conoce en occidente al sistema de producción de la empresa Toyota.

Su objetivo es la satisfacción del cliente, a través de entregar productos y servicios de calidad que el cliente necesita, cuando lo necesita y en la cantidad requerida al precio correcto, utilizando la mínima cantidad de materiales, equipamiento, trabajo y tiempo. En la figura 16 se observa el objetivo Lean de producir más con menos recursos (Galgano, 2008).



Figura 16 Lean: Mejorar reduciendo (Galgano, 2008)

La filosofía Lean tiene como punto de partida los siguientes aspectos:

- ✓ Valor: capacidad de hacer aquello por lo que el cliente va a pagar.
- ✓ Productividad: relación entre la cantidad de bienes producidos y los recursos utilizados para su obtención.

En la figura 17 se observa el modelo de funcionamiento de Lean:

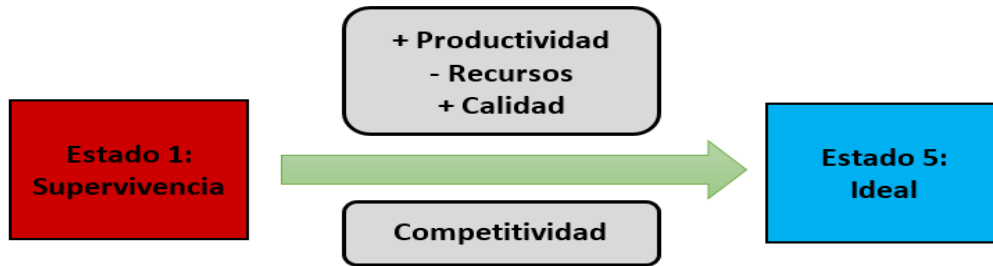


Figure 17 Filosofía Lean (Prodintec, 2010)

Beneficios (Prodintec, 2010)

- Ahorros financieros
- Flexibilidad
- Mejora conocimiento de procesos
- Procesos robustos
- Reducción de inventarios
- Reducción de tiempos de entrega

Principios (Prodintec, 2010)

Creación de valor

- Realizar aquello por lo que el cliente va a pagar.
- El valor lo crea el productor en forma de productos o servicios.

Eliminar el desperdicio

Lean distingue tres tipos de desperdicio en los procesos productivos que consumen recursos sin aportación de valor.

- Muda o desperdicio (Martín, 2010): cualquier otra cosa distinta a la cantidad mínima de equipos, materiales, partes, espacio y tiempo del trabajador que son absolutamente necesarios para dar valor al producto. En figura 18 se observa el ejemplo de muda o desperdicio.

Tipos de muda:

- | | |
|---------------|-----------------------|
| - Defectos | - Sobre-procesamiento |
| - Esperas | - Sobreproducción |
| - Inventario | - Transporte |
| - Movimientos | |



Figura 18 Muda o desperdicio (Solutions, 2019)

- Mura o irregularidad (Martín, 2010): está relacionado con los cuellos de botella que interrumpen el flujo normal de trabajo en la tarea de un operador. En la figura 19 se observa el ejemplo de mura o irregularidad:



Figura 19 Mura o irregularidad (Solutions, 2019)

- Muri o sobrecarga (Martín, 2010): Implica condiciones estresantes para trabajadores, máquinas y por consecuencia los procesos de trabajo. En la figura 20 se observa el ejemplo de muri o sobrecarga.



Figura 20 Muri o sobrecarga (Solutions, 2019)

Flujo continuo

El producto debe ir adquiriendo valor sin que paren las etapas creadoras de valor a lo largo de la cadena tal como se observa en la figura 21.

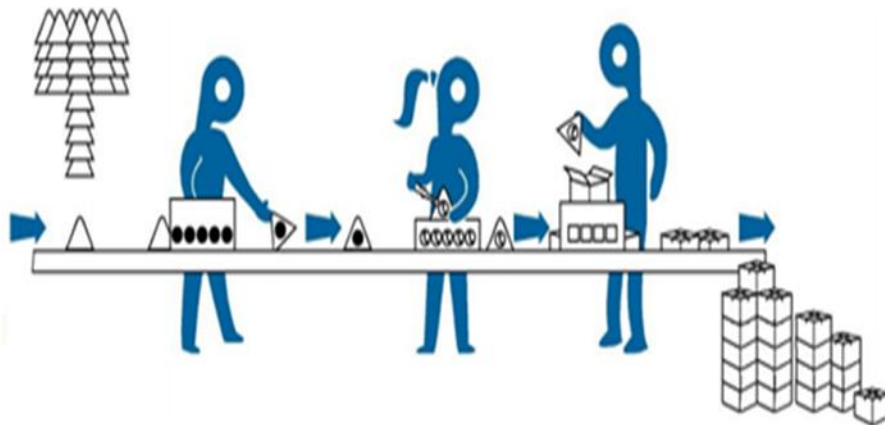


Figura 21 Flujo continuo (Prodintec, 2010)

Velocidad del cliente

Se sincronizan los procesos con el cliente, sólo se fabrica cuando el cliente solicita producto o servicio.

Mejora continua

Tiene como base el método Kaizen como la acumulación de pequeños cambios de mejora continua.

Anexo B: Metodología Kanban

En este anexo, se detalla origen, características y funcionamiento de la Metodología Ágil Kanban.

Kanban

Es una herramienta del método de abastecimiento Justo a tiempo, el cual pertenece a la filosofía de producción Lean (Ríos, 2015). La palabra es de origen japonés y se compone de los términos:

Kan= “visual”

Ban = “insignia”

Kanban = Insignia visual

Su objetivo es gestionar como se van completando las tareas, y en la actualidad se ha utilizado en la gestión de proyectos de desarrollo software ya que es una metodología adaptativa que tiene la menor resistencia al cambio.

Kanban ha ganado popularidad en la industria del software ya que proporciona un medio para visualizar y limitar el progreso del trabajo durante el proceso de desarrollo de software (Matharu et al., 2015).

Características (Matharu et al., 2015)

- Tablero kanban: herramienta de visualización de flujo de trabajo para optimizar el trabajo y guía el flujo de trabajo dividiendo el trabajo en categorías.
- Maximiza productividad: programa y optimiza el flujo de trabajo, maximizando la productividad del equipo al reducir el tiempo de inactividad.
- Entrega continua: lanzamiento de pequeñas partes del producto en iteraciones sucesivas está dirigido a satisfacer los requisitos dinámicos de los clientes.
- Minimización de residuos: ejecutar tareas solo si es necesario para eliminar la sobreproducción con el fin de reducir trabajo y tiempo perdido.
- Límites de trabajo en progreso: limitar el trabajo en progreso para optimizar el flujo de trabajo parcial o total del sistema de acuerdo con su capacidad.

Reglas (Pérez, 2012)

Visualizar el trabajo y etapas del ciclo de producción.

- Utiliza notas en una pizarra como técnica para visualizar dentro de columnas o estados necesarios la descripción y estimación de duración de la tarea.
- Tiene como base el desarrollo incremental, dividiendo el trabajo en partes.

En la figura 22 se muestran posibles estados o columnas de un tablero Kanban los cuales pueden ampliarse o reducirse de acuerdo a las necesidades del proyecto.



Figura 22 Tablero Kanban (Pérez, 2012)

Determinar el límite del trabajo en curso.

- Definir el máximo de tareas a realizarse en cada fase del ciclo de trabajo como límite del trabajo en progreso.
- Para iniciar una tarea es primordial terminar una previa.
- El objetivo de establecer límites de tareas es detectar cuellos de botella.

Medir el tiempo en completar una tarea.

- Es la métrica más conocida del Kanban y se utiliza para medir lo que ven y esperan los clientes.
- Otra métrica utilizada es Ciclo de Tiempo, con la que se mide el rendimiento del proceso desde que el trabajo sobre una tarea inicia hasta que termina.

Roles

En la metodología Kanban no ocupan roles para evitar crear una identidad en el individuo asociada a tareas específicas, con esto se disminuye la resistencia emocional al cambio y se ve como una ventaja en el equipo la ausencia de un papel asignado a un individuo.

Anexo C: Metodología Extreme Programming

En este anexo, se detalla origen, características y funcionamiento de la Metodología Ágil Extreme Programming.

Metodología Extreme Programming (Wells, 2009)

El primer proyecto de Programación Extrema se inició el 6 de marzo de 1996. Esta metodología es exitosa porque enfatiza la satisfacción del cliente y el trabajo en equipo. Los gerentes, clientes y desarrolladores son socios iguales en un equipo colaborativo, con un entorno simple pero efectivo que les permite ser altamente productivos. El equipo se auto-organiza alrededor del problema para resolverlo de la manera más eficiente posible.

El aspecto más sorprendente de la Programación Extrema son las reglas simples, las cuales se muestran en el diagrama de flujo de la figura 23.

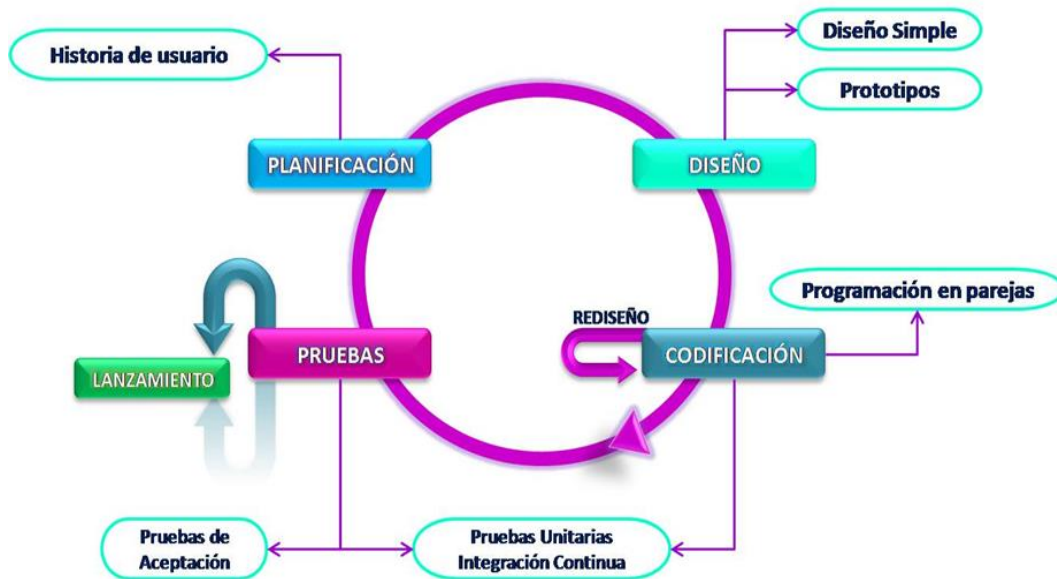


Figura 23 Flujo de XP (Wells, 2009)

Fases (Pérez, 2012)

Exploración

Mediante la redacción de sencillas historias de usuario los programadores obtienen en dos semanas la visión general del sistema y estimaciones primarias de tiempos de desarrollo.

Planificación

Fase corta, en la que clientes, gerentes y desarrolladores acuerdan el orden de implementación de las historias de usuario, y su relación con las entregas.

Iteraciones

Es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a las iteraciones las cuales miden el progreso del proyecto.

Producción

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste y si es deseable por el cliente, no se pone el sistema en producción hasta que la funcionalidad este completa.

Reglas (Wells, 2009)

Planificación

Historias de Usuarios

- Escritas por el usuario como actividades que el sistema tiene que hacer por ellos.
- Utilizada para crear estimaciones de tiempo para la reunión de planificación del lanzamiento.
- Proporcionan detalles suficientes para realizar una estimación de riesgo razonablemente baja de cuánto tardará en implementarse la historia.
- Tienen estimación de 1 a 3 semanas como tiempo de desarrollo ideal para implementar código.

Planificación de lanzamiento

- Utilizado para crear planes de iteración para cada iteración individual.
- Los técnicos toman decisiones técnicas y empresarios decisiones comerciales.
- Permite estimar en términos de semanas cada historia de usuario.
- La velocidad del proyecto se utiliza para determinar la cantidad de historias que se pueden implementar por tiempo o alcance.
- Un proyecto se puede cuantificar mediante cuatro variables: Alcance, recursos, tiempo y calidad.

Pequeños lanzamientos frecuentes

- Lanzamiento de versiones iterativas del sistema a los clientes frecuentemente.
- Publicar pequeñas unidades de funcionalidad en el entorno del cliente.

Desarrollo iterativo

- Divide los proyectos en una docena de iteraciones con duración de 1 a 3 semanas.
- Concentra el esfuerzo en completar las tareas más importantes según lo elegido por el cliente, en lugar de terminar lo elegido por los desarrolladores.

Planificación de la iteración

- Historias de usuario y pruebas fallidas se dividen en tareas de programación.
- Las historias de usuario están en el idioma del cliente, las tareas en el idioma del desarrollador.

Gerencia

Equipo de trabajo con espacio abierto

- Facilitar la programación de pares y alentar a las personas a trabajar juntas con sentimientos de igual valor y contribuciones.
- Inclusión de un área grande para reuniones de pie diarias.
- Mesa de conferencias para discusiones grupales espontáneas a lo largo del día.
- Pizarras blancas para bocetos de diseño y notas importantes o paredes en blanco donde las tarjetas de historia de los usuarios pueden ser grabadas.

Ritmo sostenible, medible, predecible

- Si es necesario, enfocarse en una sola tarea completada que muchas tareas incompletas.
- Use la reunión de planificación de lanzamiento para cambiar el alcance o el calendario del proyecto.
- Ayuda a planificar tus lanzamientos e iteraciones

Reuniones diarias de pie

- Tienen como propósito La comunicación entre todo el equipo.
- Útiles para comunicar problemas, soluciones y promover el enfoque del equipo.
- La asistencia puede basarse en quiénes son realmente necesarios y contribuirán.
- Durante la reunión los desarrolladores informan al menos tres cosas: lo que se logró ayer, lo que se intentará hoy y los problemas que causan demoras.

Velocidad del proyecto

- Mide la cantidad de trabajo que se realiza en el proyecto.
- Son la suma de historias de usuario terminadas durante la iteración.

Mover a las personas

- Evita la pérdida de conocimientos y codificar cuellos de botella.
- Fomenta que todos en el equipo sepan mucho del código en cada sección.

Diseño

Simplicidad

- Dentro del proyecto el equipo decide lo que es simple tomando en cuenta cuatro cualidades subjetivas: Probable, comprensible, navegable y explicable.

Metáforas del sistema

- Elegir nombres para sus objetos con el que todos puedan relacionarse.

Tarjetas de Clase, Responsabilidades y Colaboración

- Permite a las personas romper con el modo de pensamiento procedimental y apreciar plenamente la tecnología de objetos.
- Permiten que equipos del proyecto contribuyan al diseño.

Soluciones de punta

- Solo resuelva el problema que se examina e ignora las demás preocupaciones.
- Su objetivo es reducir el riesgo de un problema técnico y aumentar la confiabilidad de la estimación de una historia de usuario.

Funcionalidades innecesarias

- Mantener el código listo para cambios inesperados si se trata de un diseño simple.
- Enfoque solo en lo que está programado para hoy.

Refactorización

- Refactorizar durante el proyecto ahorra tiempo y aumenta la calidad.
- Refactorizar mantiene el diseño simple y evita desorden o complejidad innecesarios.
- Mantenga el código limpio y conciso para entenderlo, modificarlo y extenderlo.

Codificación

Cliente siempre disponible

- Historias de usuario escritas por el cliente, con la ayuda de los desarrolladores, para permitir estimaciones de tiempo y asignar prioridad.
- Los clientes ayudan a asegurar que las funciones del sistema estén cubiertas por historias.
- En la reunión de planificación de lanzamiento se definen pequeños lanzamientos incrementales para permitir que la funcionalidad se libere al cliente antes de tiempo.
- Los clientes prueban el sistema antes y dan a los desarrolladores comentarios antes de liberar el lanzamiento.

Pruebas de unidad

- Crear pruebas primero, antes del código, resultará mucho más fácil y rápido para crear el código.
- Crear pruebas unitarias ayuda al desarrollador a considerar realmente lo que se necesita hacer.

Programación por pares

- Aumenta la calidad del software sin afectar el tiempo de entrega.
- Dos personas trabajando en una sola computadora agregan tanta funcionalidad como dos que trabajan por separado, excepto que su calidad será mayor.
- La programación en pareja es una habilidad social que requiere tiempo para aprender.
- La programación por pares no es una relación profesor-estudiante.

Integración secuencial

- Es efectivo tener desarrolladores de clases específicas e integrarlas en el repositorio.
- El desarrollo se realiza en paralelo, mientras que la integración es secuencial.
- Integrar código en el repositorio cada poco tiempo, siempre que sea posible.
- La integración continua detecta problemas de compatibilidad en etapas iniciales.
- Una computadora dedicada a los lanzamientos secuenciales funciona bien cuando el equipo de desarrollo se ubica conjuntamente.

Propiedad colectiva

- Alienta a todos a contribuir con nuevas ideas en todos los segmentos del proyecto.
- Todos los desarrolladores pueden cambiar cualquier línea de código para agregar funcionalidad, corregir errores, mejorar diseños o refactorizar.
- Evita que alguna persona se convierte en un cuello de botella para los cambios.
- Evita depender de una persona, que puede abandonar el proyecto, a cargo de las clases específicas.

Pruebas

Pruebas unitarias

- Se publican en el repositorio de códigos junto con el código que prueban.
- Permiten la propiedad colectiva y evita que la funcionalidad se dañe.
- Permiten la refactorización y pueden verificar que los cambios en la estructura no generen cambios en la funcionalidad.
- Proporcionan una red de seguridad de pruebas de regresión y pruebas de validación para refactorizar e integrar de manera efectiva.

Pruebas de aceptación

- Son pruebas del sistema que representan resultados esperados del sistema.
- Los clientes verifican su exactitud y los puntajes obtenidos para decidir cuales pruebas fallidas son de mayor prioridad.

Valores (Wells, 2009)

Sencillez

Solo se realiza lo solicitado y necesario para maximizar el valor creado para la inversión realizada hasta la fecha. Se avanza con pasos pequeños y sencillos respecto a la meta y se resuelven las fallas según van ocurriendo. Cuando se crea algo de alto valor, es preferible mantenerlo a largo plazo por costos razonables.

Comunicación

Todos los elementos del equipo se comunican cara a cara todos los días. Los elementos del equipo trabajan juntos en todo, desde los requisitos hasta el código y crean la mejor solución para el problema.

Comentarios

Para entregar un software que funcione se toman como prioritaria cada iteración. Se muestra el software frecuentemente para analizarlo y realizar los cambios necesarios. Se habla del proyecto y se adapta el proceso al proyecto.

Respeto

Todos en el equipo otorgan y reciben el respeto como miembro valioso del equipo. Todos aportan valor incluso si es simplemente entusiasmo. El respeto entre desarrolladores y clientes es mutuo. La gerencia respeta nuestro derecho a aceptar responsabilidad y recibir autoridad sobre nuestro propio trabajo.

Valor

Es prioridad decir la verdad acerca del progreso y las estimaciones. Nos adaptaremos a los cambios cuando ocurran.

Roles (Pérez, 2012)

Ciente

- Define proyecto, objetivos, toma decisiones e identifica valor de su historia.
- Responde a las siguientes preguntas: ¿Qué debería hacer esta nueva característica?, ¿Cómo sabremos cuando está lista?, ¿Cuánto podemos gastar?, ¿Cuándo comenzamos a trabajar sobre ella?
- Tiene derecho a maximizar la inversión, determinar funciones siguientes, medir el progreso del proyecto lanzando pruebas de aceptación y cambiar el alcance del proyecto para enfrentar cambios en la planeación.
- Es responsable de confiar en las decisiones técnicas de los desarrolladores, analizar riesgos, seleccionar historias con mayor valor y participar en el equipo dando directrices de manera pronta y precisa.

Programador

- Responde a las siguientes preguntas: ¿Cómo implementamos esta funcionalidad?, ¿Cuánto nos va a llevar?, ¿Cuáles son los riesgos?
- Tiene derecho a estimar su trabajo, no tomar decisiones de negocio e implementar las funcionalidades que cubren las necesidades del cliente.
- Es responsable de seguir las directrices del equipo, desarrollar funcionalidades necesarias y comunicarse de forma fluida con el cliente.

Encargado de pruebas

Ayuda al cliente a definir y escribir las pruebas de aceptación de las historias de usuario y responsable realizar pruebas periódicas para informar de los resultados al equipo.

Encargado de seguimiento

Realiza el seguimiento del proyecto de acuerdo a la planificación para determinar si el proyecto lleva una velocidad dentro del tiempo de iteración.

Entrenador

No es un rol cubierto en todos los equipos de XP, por lo tanto, es útil para guiar y orientar al equipo cuando son principiantes con la metodología XP.

Gestor

Puede ser el cliente, y como gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado.

Anexo D: Metodología Scrum

En este anexo, se detalla origen, características y funcionamiento de la Metodología Scrum.

Scrum (SCRUMstudy, 2017).

El origen de Scrum se deriva de la palabra francesa Melé cuyo significado se refiere a un tipo de jugada en el Rugby en donde todos los jugadores de ambos equipos se agrupan en una formación, en la cual lucharan por obtener el balón que se introduce por el centro. En la década de los 80's Hirotaka Takeuchi y Ikujiro Nonaka propusieron que el desarrollo de productos no debe ser como una carrera de relevos secuencial, sino que debería ser análogo al del juego de rugby, donde el equipo trabaja en conjunto pasando el balón hacia atrás y hacia adelante a medida que se desplaza en unidad por el campo. En la figura 24 se observa el marco general de trabajo de Scrum.

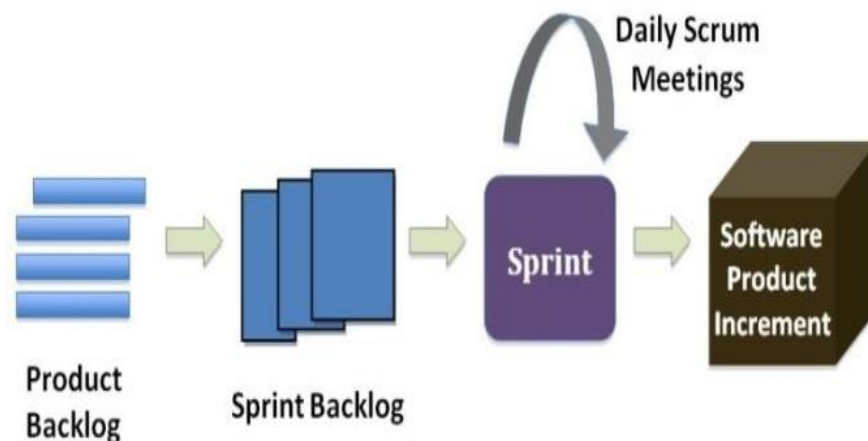


Figura 24 Metodología Scrum (Matharu et al., 2015)

Posteriormente, Ken Schwaber y Jeff Sutherland desarrollaron el concepto de Scrum y su aplicabilidad al desarrollo de software durante una presentación en la Conferencia internacional sobre programación, lenguajes y aplicaciones orientadas a objetos en 1995 en Austin, Texas. Desde entonces, Scrum ha aumentado en popularidad y actualmente es el método de desarrollo de proyectos favorito de muchas organizaciones a nivel mundial.

Ventajas de Scrum

- Adaptabilidad
- Alta velocidad
- Ambiente de alta confianza
- Ambiente innovador
- Centrado en el cliente
- Entrega anticipada de alto valor
- Entrega continua de valor
- Entregables efectivos
- Mejora continua
- Motivación
- Proceso de desarrollo eficiente
- Resolución rápida de problemas
- Responsabilidad colectiva
- Retroalimentación continua
- Ritmo sostenible
- Transparencia

Scrum versus Gestión tradicional de proyectos

En la tabla 8 se observa la comparación del funcionamiento de características entre la gestión tradicional y la metodología ágil Scrum:

Tabla 8 Scrum vs Gestión tradicional (SCRUMstudy, 2017)

Característica	Scrum	Gestión tradicional
Cambio	Actualizaciones al Backlog Priorizado del Producto	Sistema formal de gestión del cambio
Documentación	Según se requiera	Integral
Énfasis	Personas	Procesos
Estilo de gestión	Descentralizado	Centralizado
Garantía de calidad	Centrada al cliente	Centrada al proceso
Liderazgo	Colaborativo	Control
Medición del rendimiento	El valor del negocio	Conformidad con el plan
Organización	Auto organizada	Gestionada
Participación del cliente	Alta durante el proyecto	Variante dependiente del ciclo de vida del proyecto
Planificación adelantada	Baja	Alta
Priorización de requerimientos	Según el valor del negocio	Fijo
Procesos	Iterativos	Lineales
Retorno sobre la inversión	Al inicio y largo del proyecto	Al final del proyecto

Principios de Scrum

Se pueden aplicar a todo tipo de proyecto en diversas organizaciones, se deben aplicar sin modificar, a fin de garantizar la aplicación efectiva del marco de referencia de Scrum. En la figura 25 se observan los principios aplicables a lo siguiente:

- Portafolios, programas, y/o proyectos de cualquier industria.
- Productos, servicios, o resultados que se entregue a los stakeholders.
- Proyectos de cualquier tamaño y complejidad.

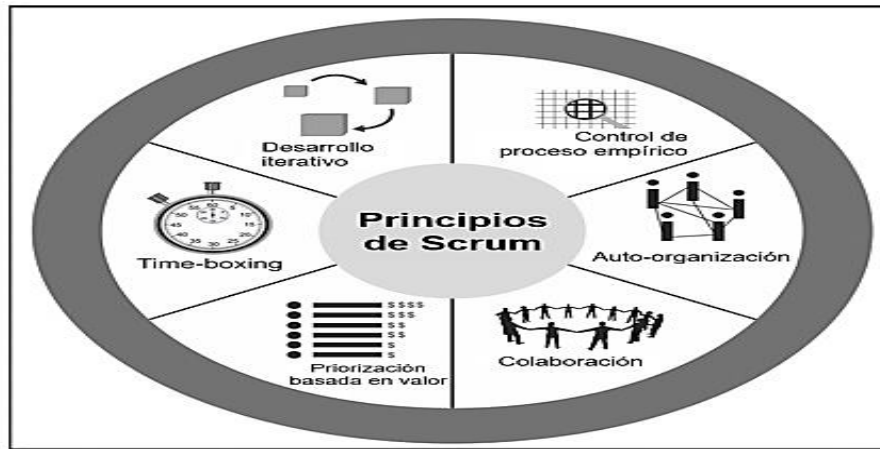


Figure 25 Principios de Scrum (SCRUMstudy, 2017)

1. Control de proceso empírico

Scrum basa las decisiones en la observación y la experimentación en lugar de la planificación inicial detallada. El control del proceso empírico se basa en tres ideas principales: transparencia, inspección y adaptación.

Transparencia

Promueve un flujo de información fácil y transparente en toda la organización creando una cultura de trabajo abierta. En la figura 26 se observa cómo funciona la transparencia:

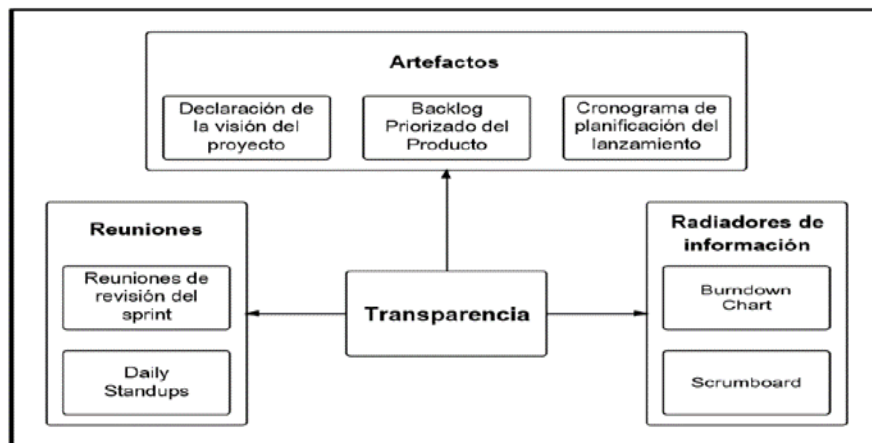


Figura 26 Transparencia en Scrum (SCRUMstudy, 2017)

La transparencia se representa mediante lo siguiente:

- Declaración de la visión del proyecto (Project Vision Statement) que pueden ver todos los stakeholders y el Equipo Scrum.
- Backlog Priorizado del Producto abierto con historias de usuario priorizadas que todos pueden ver tanto dentro como fuera del Equipo Scrum.
- Cronograma de planificación del lanzamiento (Release Planning Schedule) que se coordina a través de múltiples equipos Scrum.

- Clara visibilidad sobre el progreso del equipo a través del uso de Scrumboard, Burndown Chart y otros radiadores de información.
- Daily Standups que se llevan a cabo durante el proceso de realizar Daily Standup en las que todos los miembros del equipo informan sobre lo que hicieron el día anterior, lo que van a hacer hoy y los problemas que les impide completar sus tareas en el sprint actual.
- Las reuniones de revisión del sprint se llevan a cabo durante el proceso de demostrar y validar el sprint, donde el Equipo Scrum muestra los entregables del sprint que potencialmente se pueden enviar a los Product Owners y a los stakeholders.

Inspección

Se representa mediante lo siguiente:

- Uso de un Scrumboard común y otros radiadores de información que muestran el progreso del Equipo Scrum en completar las tareas del sprint actual.
- Recopilación de la retroalimentación del cliente y otros stakeholders durante los procesos de Desarrollar de épicas, Crear Backlog Priorizado del Producto y Realizar planificación del lanzamiento.
- La inspección y aprobación de los entregables por parte del Product Owner y el cliente en el proceso de Demostrar y validar el sprint.

En la figura 27 se observa el concepto de inspección en Scrum.

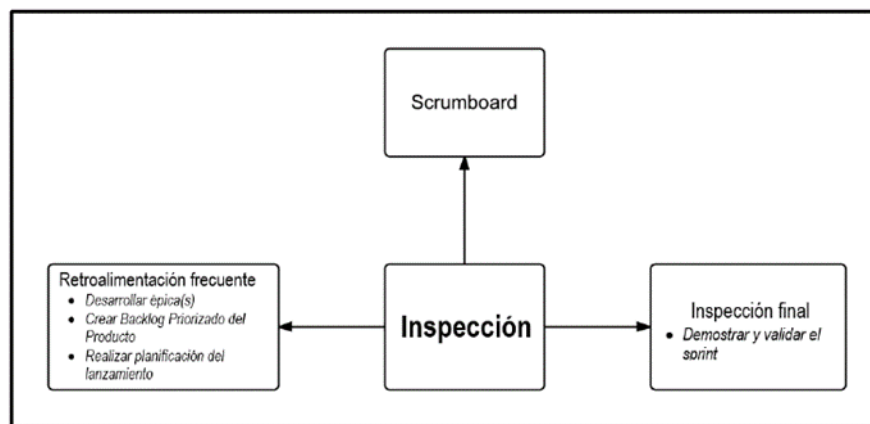


Figura 27 Inspección en Scrum (SCRUMstudy, 2017)

Adaptación

Sucede cuando el equipo principal de Scrum y los stakeholders aprenden mediante la transparencia y la inspección, y después se adaptan al hacer mejoras en el trabajo que llevan a cabo. En la figura 28 se observa de forma general la adaptación:

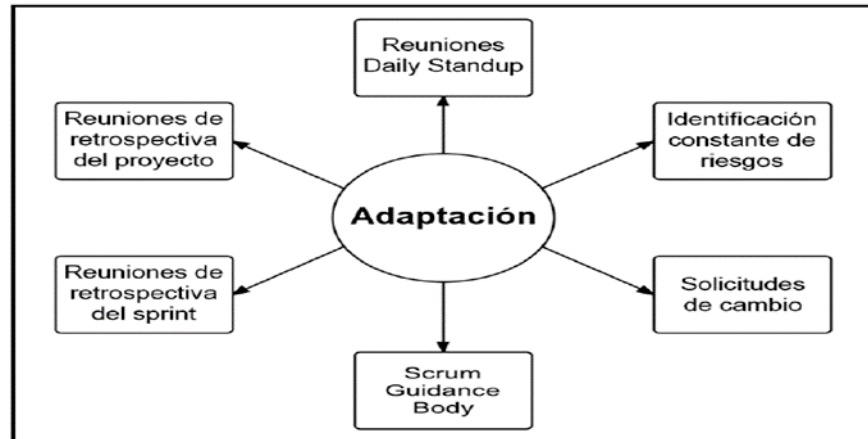


Figura 28 Adaptación en Scrum (SCRUMstudy, 2017)

Algunos ejemplos de adaptación incluyen:

- En los Daily Standups los miembros del Equipo Scrum hablan abiertamente sobre los impedimentos para completar sus tareas y buscan la ayuda de otros miembros.
- Se identifican los riesgos a lo largo del proyecto. Los riesgos identificados se convierten en entradas para varios procesos de Scrum, incluyendo el de Crear Backlog Priorizado del Producto, de Refinamiento del Backlog Priorizado del Producto y de Demostrar y validar el sprint.
- Las mejoras pueden resultar en solicitudes de cambios que se discuten y aprueban durante los procesos de Desarrollar épica(s), Crear Backlog Priorizado del Producto y Refinamiento del Backlog Priorizado del Producto.
- El Scrum Guidance Body interactúa con los miembros del Equipo Scrum durante los procesos de Crear historias de usuario, Estimar tareas, Crear entregables y Refinamiento del Backlog Priorizado del Producto para ofrecer orientación y también proporcionar conocimientos según sea necesario.
- En el proceso de Retrospectiva del sprint se determinan las mejoras aceptadas con base en las salidas del proceso de Demostrar y validar el sprint.
- En la reunión de retrospectiva del proyecto, los participantes documentan las lecciones aprendidas y realizan revisiones en busca de oportunidades para mejorar los procesos y atender ineficiencias.

2. Auto-organización

El liderazgo preferido de Scrum es el servicial enfocado en el logro de resultados y se logra cuando los empleados cuentan con motivación propia ya que aceptan mayores responsabilidades.

Por lo tanto, los equipos auto organizados ofrecen más valor. La auto organización significa que una vez que la visión del producto se define en el proceso de Crear la visión del proyecto, se identifica Product Owner, Scrum Master y Equipo Scrum. En la figura 29 se observan las metas del equipo auto organizado:

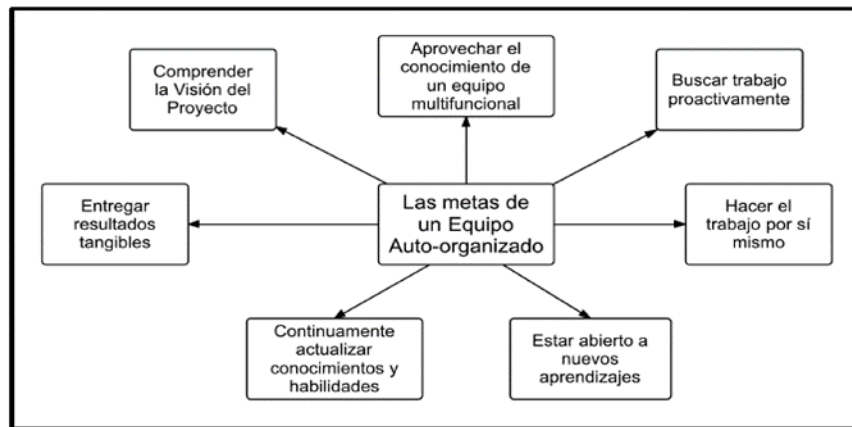


Figure 29 Objetivos de equipo auto organizado (SCRUMstudy, 2017)

Los principales objetivos de los equipos auto-organizados son los siguientes:

- Entender la visión del proyecto y por qué aporta valor a la organización.
- Estimar historias de usuario durante el proceso de Estimar historias de usuario y asignarse tareas durante el proceso de Crear el Sprint Backlog.
- Identificar tareas en forma durante el proceso de Identificación de tareas.
- Aplicar la experiencia de ser un equipo interfuncional al trabajar en las tareas durante el proceso de Crear entregables.
- Entregar resultados tangibles que sean aceptados por el cliente y otros stakeholders durante el proceso de Demostrar y validar el sprint.
- Resolver problemas individuales analizándolos durante los Daily Standups.
- Aclarar dudas y tener la disposición de aprender cosas nuevas.
- Actualizar conocimientos y habilidades continuamente.
- Estabilidad de los miembros del equipo durante el proyecto. Cambiar los miembros del equipo solo si es inevitable.

3. Colaboración

Es cuando el equipo principal de Scrum trabaja e interactúa con los stakeholders para crear y validar los resultados del proyecto a fin de cumplir con los objetivos que se plantean en la visión del proyecto. La colaboración se produce cuando un equipo trabaja en conjunto para cotejar los aportes del otro a fin de producir algo más grande. En la figura 30 se observa la forma de colaborar en proyectos Scrum:



Figura 30 Colaboración en proyectos Scrum (SCRUMstudy, 2017)

Dimensiones básicas de trabajo en la colaboración:

- Conocimiento: personas que trabajan juntas están al tanto del trabajo de los demás.
- Articulación: los colaboradores distribuyen el trabajo en unidades y las dividen entre los miembros del equipo y cuando el trabajo esté hecho se reintegran.
- Apropiación: se adapta la tecnología a la situación individual.

Beneficios de la colaboración en los proyectos Scrum.

- La necesidad de cambios debido a requisitos mal clarificados se reduce al mínimo.
- Los riesgos se identifican y se atienden con eficiencia.
- Se logra el verdadero potencial del equipo.
- Se garantiza la mejora continua a través de las lecciones aprendidas.

4. Priorización basada en el valor

La priorización se define como la determinación del orden y separación de lo que debe hacerse ahora, de lo que debe hacerse después. En la figura 31 se observa el modelo de priorización a través del valor.

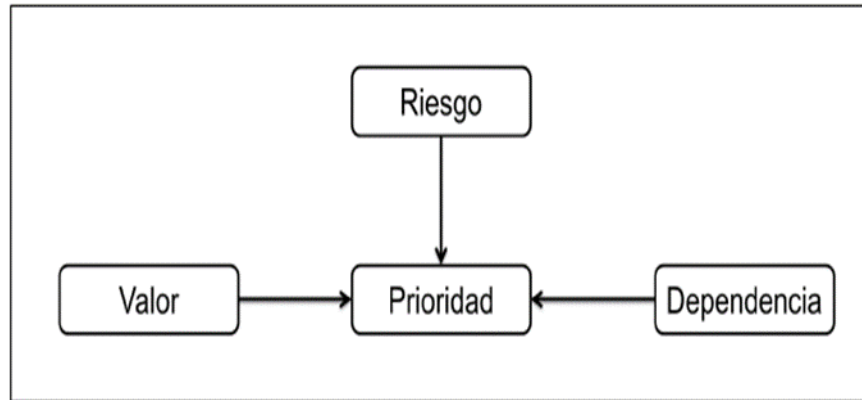


Figura 31 Priorización basada en el valor (SCRUMstudy, 2017)

La finalidad de Scrum es entregar productos o servicios valiosos para el cliente en forma oportuna y continua. Para lograrlo, Scrum utiliza la priorización basada en valor (Value-based Prioritization) como uno de los principios básicos que impulsa la estructura y funcionalidad de todo el framework de Scrum, el cual ayuda a que los proyectos se beneficien mediante la capacidad de adaptación y el desarrollo iterativo del producto o servicio.

El Product Owner interpreta las entradas y necesidades de los proyectos de los stakeholders para crear el Backlog Priorizado del Producto. Por lo tanto, mientras se priorizan las historias de usuario en el Backlog Priorizado del Producto, se consideran los siguientes tres factores:

1. Valor
2. Riesgo o incertidumbre
3. Dependencias

Los resultados de la priorización son entregables que satisfacen los requisitos del cliente ofreciendo el máximo valor de negocio en el menor tiempo posible.

5. Bloque de tiempo

Propone la fijación de cierta cantidad de tiempo para cada proceso y actividad en un proyecto para garantizar que los miembros del Equipo Scrum no ocupen demasiado tiempo en un trabajo determinado y evitar desperdiciar energía en un trabajo para el cual tienen poca claridad. En la figura 32 se observa el funcionamiento del bloque de tiempo.

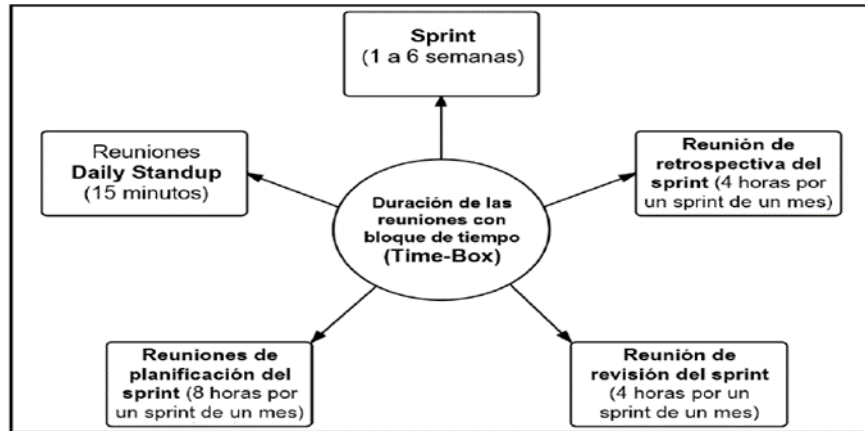


Figura 32 Bloques de tiempo para reuniones de Scrum (SCRUMstudy, 2017)

Ventajas del Time boxing:

- ✓ Proceso de desarrollo eficiente
- ✓ Menos gastos generales
- ✓ Alta velocidad para los equipos

Times-boxes de Scrum

- Sprint: es una iteración con un time-box de una a seis semanas de duración durante el cual el Scrum Master guía, facilita y protege al Equipo Scrum de impedimentos tanto internos como externos durante el proceso de Crear entregables.
- Daily standup: breve reunión diaria de 15 minutos para informar el avance del proyecto respondiendo las siguientes preguntas:

¿Qué he hecho desde la última reunión?

¿Qué tengo planeado hacer antes de la siguiente reunión?

¿Qué impedimentos (si los hubiera) estoy enfrentando en la actualidad?

- Reunión de planificación del sprint: se hace antes del sprint y se divide en dos partes:
 - Definición del objetivo: en la primera parte de la reunión, el Product Owner explica las historias de usuario de más alta prioridad o los requerimientos en el Backlog Priorizado del Producto al Equipo Scrum. Después el equipo Scrum en colaboración con el Product Owner se compromete con las historias de usuario, las cuales definen la meta del sprint.
 - Identificación y estimación de tareas: El Equipo Scrum decide cómo completar los elementos seleccionados en el Backlog Priorizado del Producto seleccionados para cumplir con la meta del sprint. Las historias de usuario comprometidas y las Effort Estimated Tasks se incluyen en el Backlog Priorizado del Producto al que se le dará seguimiento.

- Reunión de revisión del sprint: tiene un time-box de cuatro horas en un sprint de un mes y se revisa el incremento del producto para compararlo con los criterios de aceptación acordados y aceptar o rechazar las historias de usuario completadas.
- Reunión de retrospectiva del sprint: el Equipo Scrum se reúne para revisar y reflexionar sobre el sprint anterior en relación a los procesos que se siguieron, las herramientas empleadas, la colaboración, mecanismos de comunicación, y otros aspectos de interés para el proyecto. El equipo discute lo que salió bien durante el sprint anterior y lo que no salió bien para aprender y mejorar sprints futuros.

6. Desarrollo iterativo

Scrum desarrolla iteraciones de entregables para cumplir con el objetivo de ofrecer el máximo valor empresarial en un mínimo periodo de tiempo. Debido a que, en la mayoría de proyectos complejos el cliente regularmente no define requisitos concretos ni está seguro de cómo será el producto final, se aplica el modelo iterativo para asegurar que los cambios que solicite el cliente se incluyan como parte del proyecto.

En cada sprint, el proceso de Crear entregables se utiliza para desarrollar las salidas del sprint. En grandes proyectos, varios equipos interfuncionales trabajan en paralelo a través de los sprints, proporcionando soluciones potencialmente entregables al final de cada sprint, los cuales se aceptan o rechazan por el Product Owner, aceptan o rechazan con base a los criterios de aceptación del proceso de Demostrar y validar el sprint. En las figuras 33 y 34 se observa el estilo de desarrollo del enfoque tradicional y la metodología Scrum.

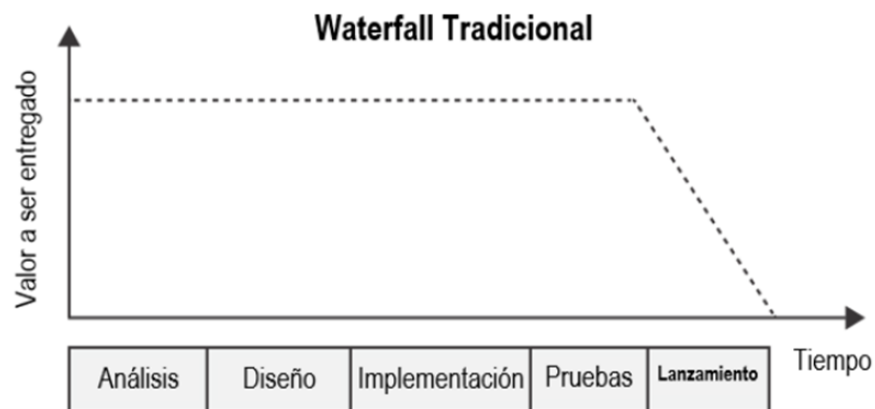


Figura 33 Iteraciones en modelo tradicional (SCRUMstudy, 2017)

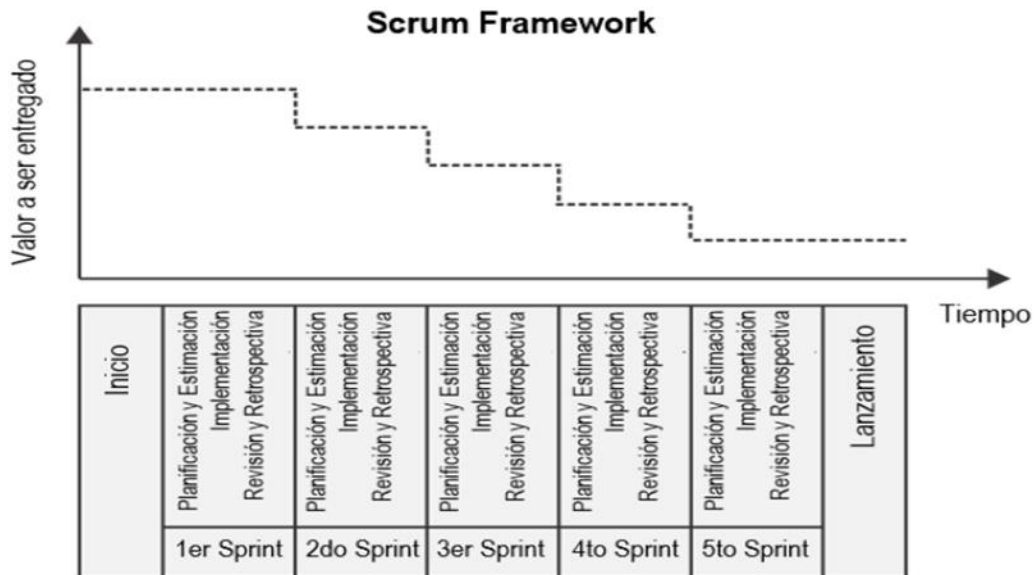


Figura 34 Iteraciones en modelo Scrum (SCRUMstudy, 2017)

Aspectos de Scrum

Se gestiona a lo largo del proyecto, en la figura 35 se observa brevemente los seis aspectos de la metodología Scrum:



Figura 35 Aspectos de Scrum (SCRUMstudy, 2017)

1. Organización

Permite entender los roles y responsabilidades definidos en un proyecto a fin de asegurar la implementación exitosa de Scrum. La organización se aplica a lo siguiente:

- Portafolios, programas y/o proyectos de cualquier sector.
- Productos, servicios o resultado que se entregue a los stakeholders.
- Proyectos de cualquier tamaño y complejidad.

Roles centrales de un proyecto Scrum

Se requieren obligadamente para crear el producto del proyecto, están comprometidos con el proyecto, y son responsables del éxito de cada sprint del proyecto y del proyecto en su totalidad. Los roles centrales son el Product Owner, el Scrum Master y el Equipo Scrum. En conjunto se les conoce como el equipo principal de Scrum. Es importante tener en cuenta que, de estos tres roles, ningún rol tiene autoridad sobre los otros.

➤ Product Owner

Es la persona responsable de maximizar el valor del negocio para el proyecto. Se asegura de que el Equipo Scrum entregue valor además de representar la voz del cliente. En la tabla 9 se observa las responsabilidades del Product Owner en los diferentes procesos de Scrum.

Tabla 9 Responsabilidades del Product Owner en los procesos de Scrum (SCRUMstudy, 2017)

Proceso	Responsabilidades del Product Owner
8.1 Crear la visión del proyecto	<ul style="list-style-type: none">• Define la visión del proyecto• Ayuda a crear el acta constitutiva del proyecto y su presupuesto
8.2 Identificar al Scrum Master y Stakeholders	<ul style="list-style-type: none">• Ayuda a finalizar la elección del Scrum Master para el proyecto• Identifica stakeholders
8.3 Formar el Equipo Scrum	<ul style="list-style-type: none">• Ayuda a determinar quiénes serán los miembros del Equipo Scrum• Ayuda a desarrollar un plan de colaboración• Ayuda a desarrollar el plan del equipo con el Scrum Master
8.4 Desarrollar épicas	<ul style="list-style-type: none">• Crea épicas y prototipos
8.5 Crear Backlog Priorizado del Producto	<ul style="list-style-type: none">• Prioriza los elementos en el Backlog Priorizado del Producto• Define los criterios de terminado
8.6 Realizar la planificación del lanzamiento	<ul style="list-style-type: none">• Elabora el cronograma de planificación del lanzamiento• Ayuda a determinar la duración del sprint
9.1 Crear historias de usuario	<ul style="list-style-type: none">• Ayuda a crear historias de usuario• Define los criterios de aceptación para cada historia de usuario
9.2 Estimar historias de usuario	<ul style="list-style-type: none">• Aclara las historias de usuario
9.3 Comprometer historias de usuario	<ul style="list-style-type: none">• Facilita el Equipo Scrum y compromete historias de usuario
9.4 Identificar tareas	<ul style="list-style-type: none">• Explica las historias de usuario al Equipo Scrum mientras elabora la lista de tareas
9.5 Estimar tareas	<ul style="list-style-type: none">• Brinda orientación y aclaraciones al Equipo Scrum en la estimación de los esfuerzos para las tareas
9.6 Crear el Sprint Backlog	<ul style="list-style-type: none">• Aclara los requerimientos al Equipo Scrum mientras elabora el Sprint Backlog

10.1 Crear entregables	<ul style="list-style-type: none"> • Aclara los requerimientos empresariales al Equipo Scrum
10.3 Refinar Backlog Priorizado del producto	<ul style="list-style-type: none"> • Refina el Backlog Priorizado del Producto.
11.1 Demostrar y validar el sprint	<ul style="list-style-type: none"> • Acepta/rechaza los entregables • Proporciona la retroalimentación necesaria al Scrum Master y a los equipos Scrum • Actualiza el plan de lanzamiento y el Backlog Priorizado del Producto
12.1 Enviar entregables	<ul style="list-style-type: none"> • Ayuda a enviar los lanzamientos del producto y se coordina con el cliente
12.2 Retrospectiva del proyecto	<ul style="list-style-type: none"> • Participa en las reuniones de retrospectiva del sprint

➤ Scrum Master

Es un facilitador que asegura que el Equipo Scrum esté dotado de un ambiente propicio para completar con éxito el desarrollo del producto. Enseña las prácticas de Scrum a los participantes en el proyecto, elimina los impedimentos que enfrenta el equipo y se asegura de que se estén siguiendo los procesos de Scrum.

El Scrum Master está en el mismo nivel jerárquico que las otras personas de equipo Scrum, las cuales si aprende a facilitar proyectos en el Equipo Scrum pueden convertirse en Scrum Master de un proyecto o sprint. En la tabla 11 se observa las responsabilidades del Scrum Master en los diferentes procesos de Scrum.

Tabla 10 Responsabilidades del Scrum Master en los procesos de Scrum (SCRUMstudy, 2017)

Proceso	Responsabilidades del Scrum Master
8.2 Identificar al Scrum Master y Stakeholders	<ul style="list-style-type: none"> • Ayuda a identificar los stakeholders para el proyecto
8.3 Formar el Equipo Scrum	<ul style="list-style-type: none"> • Facilita la selección del Equipo Scrum • Facilita la creación del plan de colaboración y el plan de desarrollo del equipo • Garantizar que los recursos de respaldo estén disponibles para el funcionamiento del proyecto sin problemas
8.4 Desarrollar de épicas	<ul style="list-style-type: none"> • Facilitar la creación de épica(s) y prototipos (Personas)
8.5 Crear Backlog Priorizado del Producto	<ul style="list-style-type: none"> • Ayuda al Product Owner en la creación del Backlog Priorizado del Producto y en la definición de los criterios de terminado
8.6 Realizar la planificación del lanzamiento	<ul style="list-style-type: none"> • Coordina la creación del cronograma de planificación del lanzamiento • Determina de la duración del sprint
9.1 Crear historias de usuario	<ul style="list-style-type: none"> • Ayuda al Equipo Scrum en la creación de historias de usuario y sus

	<p>criterios de aceptación</p>
9.2 Estimar historias de usuario	<ul style="list-style-type: none"> Organiza las reuniones del Equipo Scrum para estimar historias de usuario
9.3 Comprometer historias de usuario	<ul style="list-style-type: none"> Organiza reuniones del Equipo Scrum para comprometer historias de usuario
9.4 Identificar tareas	<ul style="list-style-type: none"> Ayuda al Equipo Scrum a crear la lista de tareas para el siguiente sprint
9.5 Estimar tareas	<ul style="list-style-type: none"> Ayuda al Equipo Scrum a estimar el esfuerzo necesario para completar las tareas acordadas para el sprint
9.6 Crear el Sprint Backlog	<ul style="list-style-type: none"> Ayuda al Equipo Scrum a desarrollar el Sprint Backlog y el Sprint Burndown Chart
10.1 Crear entregables	<ul style="list-style-type: none"> Ayuda al Equipo Scrum a crear los entregables acordados para el sprint Ayuda a actualizar el Scrumboard y el Impediment Log
10.2 Realizar Daily Standup	<ul style="list-style-type: none"> Se asegura de que el Scrumboard y el Impediment Log permanezcan actualizados
10.3 Refinar Backlog Priorizado del producto	<ul style="list-style-type: none"> Organiza las reuniones de revisión del Backlog Priorizado del Producto
11.1 Demostrar y validar el sprint	<ul style="list-style-type: none"> Facilita la presentación de los entregables completados por el Equipo Scrum para la aprobación del Product Owner
11.2 Retrospectiva del sprint	<ul style="list-style-type: none"> Garantiza que exista un ambiente ideal para el Equipo Scrum del proyecto en los sucesivos sprints
12.2 Retrospectiva del proyecto	<ul style="list-style-type: none"> Representa al equipo principal de Scrum para proporcionar lecciones del proyecto actual en caso de ser necesario

➤ Equipo Scrum

Es un grupo responsable de entender los requerimientos del negocio especificados por el Product Owner, de estimar las historias de usuarios y de la creación final de los entregables del proyecto. En la tabla 11 se observa las responsabilidades del Equipo Scrum en los diversos procesos de Scrum.

Tabla 11 Responsabilidades del Equipo Scrum en los procesos de Scrum (SCRUMstudy, 2017)

Proceso	Responsabilidades del Equipo Scrum
8.3 Formar el Equipo Scrum	<ul style="list-style-type: none"> Proporciona aportes para la creación del plan de colaboración y del plan de desarrollo del equipo

8.4 Desarrollar de épicas	<ul style="list-style-type: none"> • Asegura una comprensión clara de la épica(s) y prototipos (Personas)
8.5 Crear Backlog Priorizado del Producto	<ul style="list-style-type: none"> • Entiende las historias de usuario en el Backlog Priorizado del Producto
8.6 Realizar la planificación del lanzamiento	<ul style="list-style-type: none"> • Está de acuerdo con los demás miembros del equipo principal de Scrum sobre la duración del sprint • Busca clarificación sobre los nuevos productos o cambios, si los hay, en los productos existentes en el Backlog Priorizado del Producto.
9.1 Crear historias de usuario	<ul style="list-style-type: none"> • Proporciona aportes al Product Owner en la creación de historias de usuario
9.2 Estimar historias de usuario	<ul style="list-style-type: none"> • Estima las historias de los usuarios aprobadas por el Product Owner
9.3 Comprometer historias de usuario	<ul style="list-style-type: none"> • Compromete las historias de usuario a realizarse en un sprint
9.4 Identificar tareas	<ul style="list-style-type: none"> • Desarrolla una lista de tareas con base en las historias de usuario y dependencias acordadas
9.5 Estimar tareas	<ul style="list-style-type: none"> • Estima el esfuerzo de las tareas identificadas y, de ser necesario, actualiza la lista de tareas
9.6 Crear el Sprint Backlog	<ul style="list-style-type: none"> • Desarrolla el Sprint Backlog y el Sprint Burndown Chart
10.1 Crear entregables	<ul style="list-style-type: none"> • Elabora los entregables • Identifica riesgos y ejecuta acciones de mitigación de riesgos, si los hay. • Actualiza el Impediment Log y las dependencias
10.2 Realizar Daily Standup	<ul style="list-style-type: none"> • Actualiza el Burndown Chart, el Scrumboard y el Impediment Log • Discute los problemas que enfrenta cada miembro y busca soluciones para motivar al equipo • Identifica riesgos, si lo hay. • Presenta solicitudes de cambio, si se requieren
10.3 Refinar Backlog Priorizado del producto	<ul style="list-style-type: none"> • Participa en las reuniones de revisión del Backlog Priorizado del Producto
11.1 Demostrar y validar el sprint	<ul style="list-style-type: none"> • Muestra los entregables completados al Product Owner para su aprobación
11.2 Retrospectiva del sprint	<ul style="list-style-type: none"> • Identifica oportunidades de mejora, si las hay, del Sprint actual y decide si está de acuerdo sobre las posibles mejoras viables para el próximo sprint
12.2 Retrospectiva del proyecto	<ul style="list-style-type: none"> • Participa en la reunión de retrospectiva del proyecto

En la figura 36 se observa cómo interactúan los cuatro roles centrales Scrum:

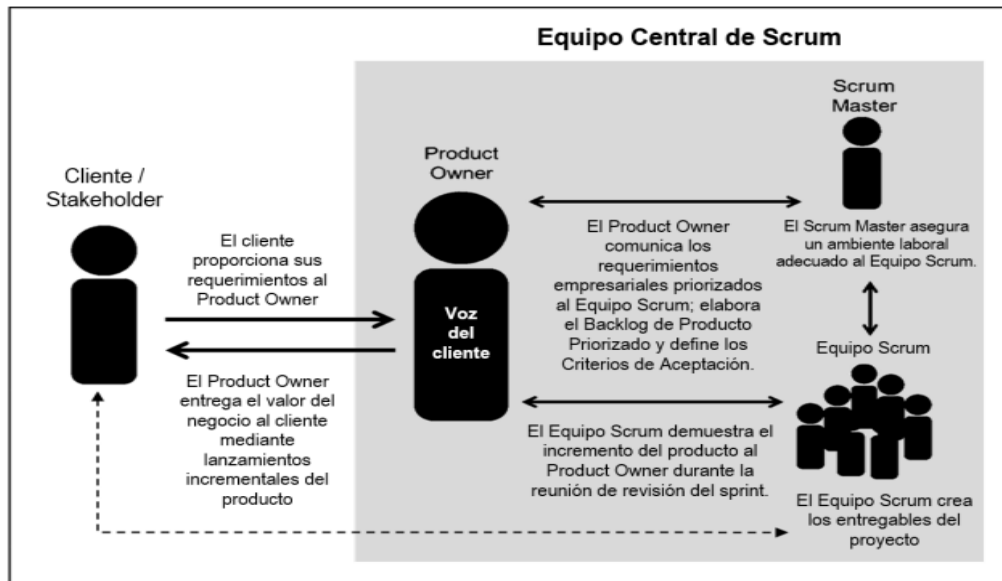


Figura 36 Interacción de roles centrales de Scrum (SCRUMstudy, 2017)

Selección de personal.

En la tabla 12 enumeran las características deseables para los roles centrales de Scrum.

Tabla 12 Roles centrales de Scrum (SCRUMstudy, 2017)

Características deseables de los roles centrales de Scrum		
 Dueño del Producto	 Maestro Scrum	 Equipo de Desarrollo
<ul style="list-style-type: none"> • Accesible • Conoce procesos Scrum • Decisivo • Dominio del negocio • Experto en Scrum • Excelente comunicación • Control de incertidumbres • Orientado a metas • Pragmático • Proactivo 	<ul style="list-style-type: none"> • Accesible • Experto en Scrum • Habilidad de coordinación • Introspectivo • Líder servicial • Mentor • Moderador • Motivador • Perceptivo • Soluciona problemas 	<ul style="list-style-type: none"> • Altamente motivados • Auto organizados • Colaborativos • Conocimiento de Scrum • Expertos técnicos • Independientes • Introspectivos • Orientados a objetivos • Proactivos • Responsables

Roles no centrales de un proyecto Scrum

Son aquellos roles que no son obligatoriamente necesarios para el proyecto Scrum y pueden no participar en el proceso de Scrum. Sin embargo, es importante tener conocimiento sobre estos roles no centrales, ya que podrían desempeñar un rol importante en algunos proyectos de Scrum.

- Stakeholders

Es un término colectivo que incluye a clientes, usuarios y patrocinadores, que generalmente interactúan con el Product Owner, el Scrum Master y el Equipo Scrum para proporcionarles las entradas y facilitar la creación del producto del proyecto, servicio o resultado. Los stakeholders influyen en el proyecto a lo largo del desarrollo del mismo. También pueden desempeñar un rol en los procesos importantes de Scrum tales como Desarrollar épicas, Crear Backlog Priorizado del Producto, Realizar la planificación del lanzamiento y Retrospectiva del sprint.

- Cliente

Persona u organización que adquiere el producto, servicio o resultado del proyecto. Para cualquier organización, dependiendo del proyecto, puede haber clientes internos (dentro de la misma organización) como clientes externos (fuera de la organización).

- Usuarios

Individuo u organización que utiliza directamente el producto, servicio o resultado del proyecto. Al igual que los clientes, para cualquier organización puede haber usuarios internos y externos. En algunas industrias los clientes y los usuarios pueden ser los mismos.

- Patrocinador

Persona u organización que provee recursos y apoyo para el proyecto. Es también el stakeholder a quien todos le deben rendir cuentas al final.

En ocasiones, la misma persona u organización puede desempeñar múltiples roles de Stakeholders.

- Vendedores

Incluyen a individuos u organizaciones externas que ofrecen productos y servicios que no están dentro de las competencias básicas de la organización del proyecto.

- Scrum Guidance Body

Es un rol opcional, aunque altamente recomendado para formalizar las prácticas organizacionales relacionadas a Scrum. Por lo general, se compone de un grupo de documentos y grupo de expertos involucrados en definir los objetivos relacionados a la calidad, regulaciones gubernamentales, seguridad y otros parámetros clave de la organización.

Estos objetivos guían el trabajo que lleva a cabo el Product Owner, el Scrum Master y el Equipo Scrum. El Scrum Guidance Body también ayuda a captar las mejores prácticas que deben utilizarse en todos los proyectos de Scrum en la organización.

2. Justificación del negocio

Se basa en el concepto de entrega impulsada por el valor (Value driven Delivery). Una característica clave de cualquier proyecto es la incertidumbre sobre los resultados. Es imposible garantizar el éxito de un proyecto, independientemente del tamaño o complejidad del mismo. Realizar entrega temprana de resultados, y por lo tanto de valor, proporciona oportunidad para la reinversión y demuestra el valor del proyecto. La adaptabilidad de Scrum permite que los objetivos y procesos del proyecto cambien si cambia su justificación del negocio.

La justificación del negocio aplica a lo siguiente:

- Portafolios, programas y/o proyectos en cualquier industria.
- Productos, servicios o resultados entregables a los stakeholders.
- Proyectos de cualquier tamaño o complejidad.

Definición de proyecto, programa y portafolios

- Proyecto: es un emprendimiento colaborativo para crear nuevos productos o servicios, o para obtener resultados como los que se definen en la declaración de la visión del proyecto (Project Vision Statement).
- Programa: es un grupo de proyectos relacionados con la finalidad de entregar resultados de negocio definidos en la declaración de la visión del programa (Program Vision Statement).
- Portafolio: es un grupo de programas relacionados con la finalidad de entregar resultados de negocio como se define en la declaración de la visión del portafolio (Portfolio Vision Statement).

Ejemplos:





- Empresa constructora
 - Proyecto: construcción de una casa
 - Programa: construcción de un complejo de viviendas
 - Portafolio: todos los proyectos de vivienda de la empresa

- Empresa de tecnología de la información
 - Proyecto: desarrollo del módulo de carrito de compras
 - Programa: desarrollo de un sitio web de comercio electrónico
 - Portafolio: todos los sitios web desarrollados por la empresa

Entrega basada en valor

A diferencia de los modelos tradicionales, en Scrum el concepto de la entrega basada en valor hace que la metodología sea muy atractiva para los interesados en el negocio y para la alta gerencia. En la tabla 13 se observa el funcionamiento de la entrega basada en valor.

Tabla 13 Valor basado en proyectos de Scrum (SCRUMstudy, 2017)

Entrega basada en valor en proyectos Scrum				
Priorizar los requerimientos con base al valor entregado a los clientes y usuarios.		Gestionar riesgos y cambios permitiendo a los stakeholders volver a priorizar después de cada sprint		Creación de incrementos entregables del producto al final de cada sprint
Proyectos tradicionales				
No se prioriza los requerimientos con base en el valor de negocio		Requerimientos fijos para todo el proyecto o cambios poco frecuentes		El valor se obtiene al final del proyecto

Responsabilidades en la justificación del negocio.

En la tabla 14 se observan las responsabilidades en la justificación del negocio en orden jerárquico.

Tabla 14 Responsabilidades en la justificación de negocio (SCRUMstudy, 2017)

Portfolio Product Owner	<ul style="list-style-type: none"> • Entrega el valor para los portafolios • Crea la justificación del negocio para los portafolios • Proporciona una guía de valor para los programas • Aprueba la justificación del negocio para los programas
Program Product Owner	<ul style="list-style-type: none"> • Entrega el valor de los programas • Crea la justificación del negocio para los programas • Proporciona una guía de valor para los proyectos • Aprueba la justificación del negocio para los proyectos
Product Owner	<ul style="list-style-type: none"> • Entrega el valor de los proyectos • Crea la justificación del negocio para los proyectos • Le confirma el logro de beneficios a los stakeholders

Importancia de la justificación del negocio.

Es importante evaluar la viabilidad de un proyecto, no solo antes de comprometerse a gastos o inversiones considerables en las etapas iniciales, sino también a lo largo del ciclo de vida del proyecto. Un proyecto debe cancelarse si se considera que no es viable; escalando la decisión a los stakeholders pertinentes y a la alta gerencia. La justificación del negocio de un proyecto debe ser evaluada al inicio de este, en intervalos predefinidos durante todo el proyecto y en cualquier momento cuando surgen grandes problemas o riesgos que amenacen su viabilidad.

Factores para determinar la justificación del negocio.

- | | |
|------------------------------|-----------------------------------|
| 1. Razonamiento del proyecto | 5. Riesgos mayores |
| 2. Necesidades del negocio | 6. Escalas de tiempo del proyecto |
| 3. Beneficio del proyecto | 7. Costos del proyecto |
| 4. Costo de oportunidad | |

3. Calidad

Se define como la capacidad con la que cuenta el producto o los entregables para cumplir con los criterios de aceptación y de alcanzar el valor de negocio que el cliente espera. Las tareas relacionadas a la calidad (desarrollo, pruebas y documentación) se completan como parte del sprint por el mismo equipo. Esto asegura que la calidad sea inherente a cualquier entregable que se crea como parte de un sprint. A tales entregables de proyectos Scrum, que son potencialmente se les conoce como “terminado”.

Scrum adopta un enfoque de mejora continua hasta la conclusión del proyecto, donde el equipo aprende de sus experiencias y de la participación de los stakeholders para mantener actualizado el Backlog Priorizado del Producto con cualquier cambio en los requerimientos.

Criterios de aceptación y Backlog Priorizado del Producto

El Backlog Priorizado del Producto es un documento de requisitos que define el alcance del proyecto, proporcionando una lista de prioridades de las características del producto o servicio a ser entregado por el proyecto. Las características necesarias se describen en forma de historias de usuario. Dichas historias son requisitos específicos señalados por varios stakeholders que se relacionan con el producto o servicio propuesto. En la figura 37 se observa el funcionamiento del Backlog priorizado.

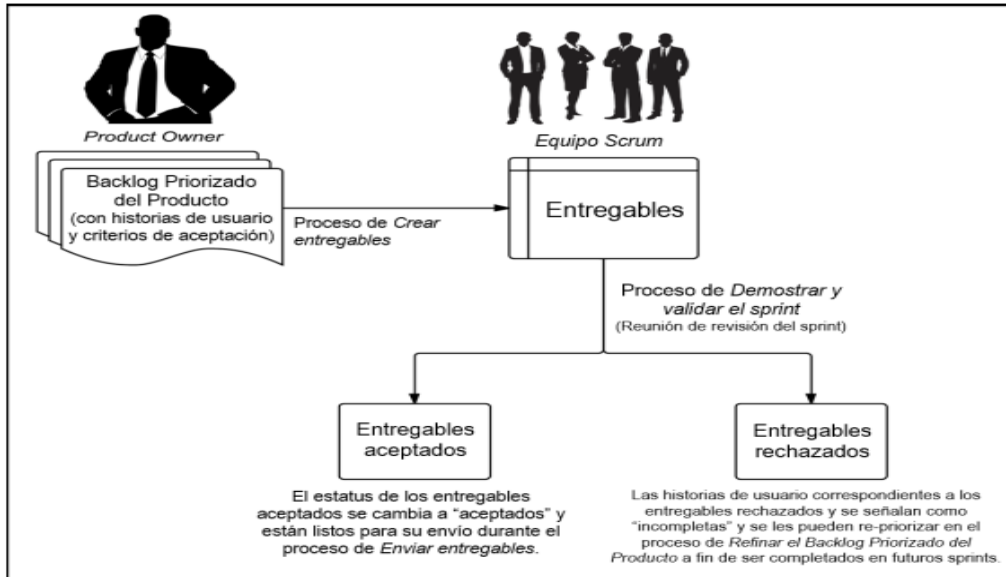


Figura 37 Backlog Priorizado del Producto (SCRUMstudy, 2017)

Redacción de criterios de aceptación

Los criterios de aceptación son únicos en cada historia de usuario y no son un sustituto de la lista de requerimientos. En la figura 38 se observa un ejemplo de los criterios de aceptación para una historia de usuario.

Ejemplo:

Prototipo del cliente: Jani es una profesionista de 36 años, está casada y tiene tres hijos. Es una mujer ocupada y exitosa que equilibra su vida profesional y personal. Se siente cómoda con la tecnología y le gusta adoptar productos y servicios innovadores. Siempre está conectada al Internet a través de múltiples dispositivos y regularmente hace compras en portales de comercio electrónico.

Historia de usuario: “Como compradora en línea, debo contar con la posibilidad de ahorrar y ver mi orden desde cualquiera de mis dispositivos para poder completar el proceso de orden cuando mejor me convenga”.

Criterios de aceptación:

- Todas las órdenes en progreso deben guardarse como pendiente (draft) cada 5 segundos en la cuenta del usuario que haya iniciado una sesión.
- Las nuevas órdenes pendientes deben mostrarse como notificaciones en cualquier dispositivo de donde el usuario inicie sesión

Figura 38 Criterios de aceptación (SCRUMstudy, 2017)

Definición de terminado

Son una serie de reglas aplicables a todas las historias de usuario en un determinado sprint. Al igual que con los criterios de aceptación, se deben cumplir todas las condiciones de los criterios de terminado para que la historia de usuario se considere terminada.

El Equipo Scrum debe utilizar una lista de verificación de los criterios de terminado generales para garantizar que una tarea está terminada y de que el resultado cumpla con la definición de terminado.

Criterios mínimos de terminado.

En la tabla 15 se pueden observar los criterios de aceptación para el proyecto, el programa y el portafolio del Product Owner.

Tabla 15 Criterios de terminado en cascada (SCRUMstudy, 2017)

Chief Product Owner/ Product Owner	<ul style="list-style-type: none"> • Establece los criterios mínimos de terminado del proyecto, mismos que incluyen los criterios de terminado del programa. • Revisa los entregables del proyecto
Program Product Owner	<ul style="list-style-type: none"> • Establece los criterios mínimos de terminado para todo el programa, mismos que incluyen los criterios de terminado del portafolio. • Revisa los entregables del programa
Portfolio Product Owner	<ul style="list-style-type: none"> • Establece los criterios mínimos de terminado de todo el portafolio. • Revisa los entregables del portafolio

Ciclo de planificar, hacer, verificar y actuar

Las directrices de la gerencia definen la calidad y pueden ofrecer un ambiente conductivo que motive a sus empleados para que mejoren la calidad en forma continua, cada empleado podrá contribuir hacia una calidad superior en el producto. En la figura 39 se observa el funcionamiento del ciclo PDCA:

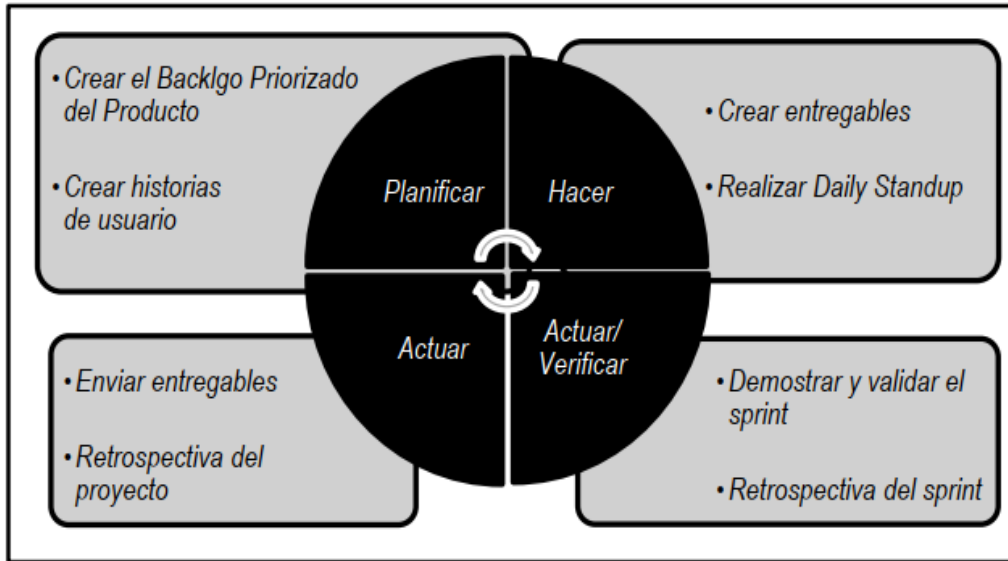


Figura 39 Ciclo PDCA en Scrum (SCRUMstudy, 2017)

4. Cambio

Un principio fundamental de Scrum es su reconocimiento de que:

- Los clientes cambian de opinión acerca de lo que necesitan durante un proyecto
- Es muy difícil que los clientes definan todos los requisitos al inicio del proyecto.

Los proyectos Scrum aceptan los cambios mediante el uso de breves iteraciones que incorporan la retroalimentación del cliente en cada entregable de la iteración. Scrum personifica un principio primordial del Manifiesto Ágil: “Responder al cambio, en vez de seguir un plan”.

La práctica de Scrum se basa en convertir a ventaja competitiva la aceptación del cambio. Se prioriza la flexibilidad en lugar de seguir un plan estricto y predefinido. Esto significa que es esencial abordar la gestión de proyectos de una manera adaptativa que permita el cambio durante el desarrollo del producto o de los ciclos rápidos de servicio. Scrum es eficaz en todo tipo de proyectos e industrias donde el mercado del producto es volátil, o cuando se requiere que el equipo sea lo suficientemente flexible para incorporar los requisitos cambiantes. Scrum es especialmente útil para proyectos complejos con mucha incertidumbre porque utiliza la transparencia, la inspección y la adaptación.

Solicitudes de cambio aprobadas y no aprobadas

Las solicitudes de cambio permanecen como no aprobadas hasta que se obtiene una aprobación formal. En la mayoría de los proyectos, el 90 % de las solicitudes de cambio podrían clasificarse como pequeños cambios que deben aprobarse por el Product Owner. Los cambios que están más allá del nivel de tolerancia del Product Owner pueden necesitar de la aprobación de los stakeholders que trabajan con él. En las figuras 40 y 41 se observa el ejemplo de aprobación de cambios y la actualización del backlog:

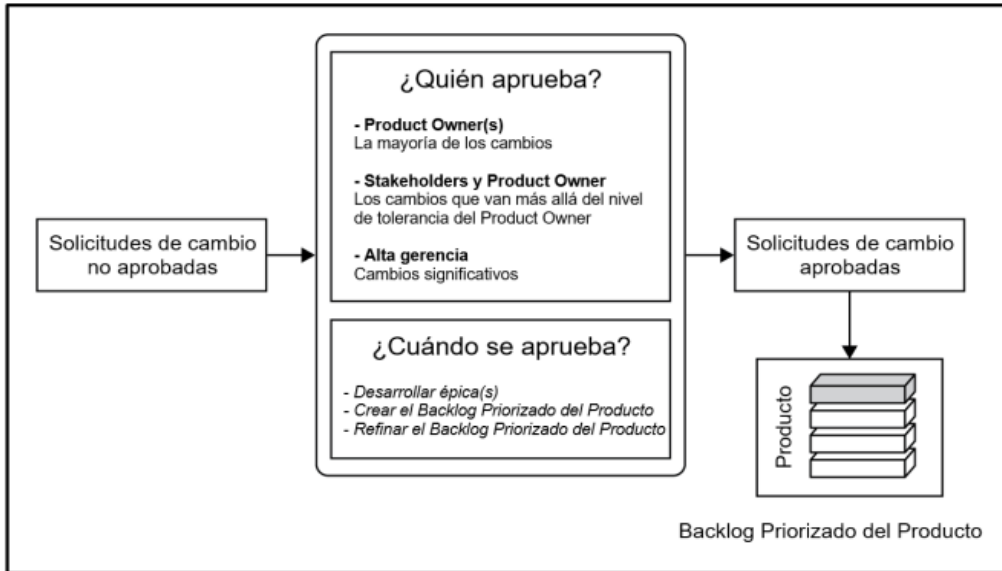


Figura 40 Proceso de aprobación de cambios (SCRUMstudy, 2017)

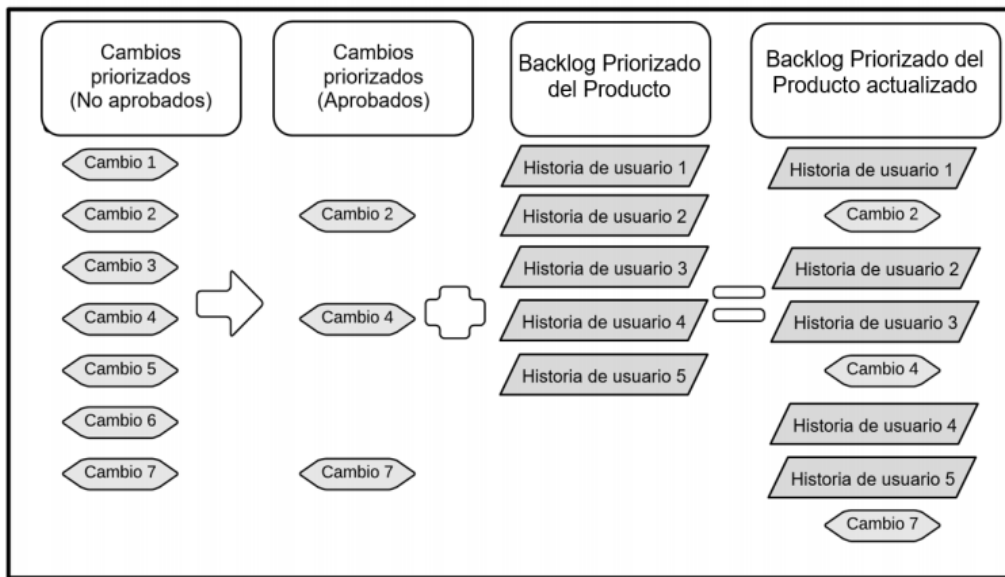


Figura 41 Actualización de Back Log (SCRUMstudy, 2017)

Cambio en Scrum

Scrum ayuda a las organizaciones a ser más flexibles y receptivas al cambio. Por lo tanto, se utiliza el desarrollo iterativo, aplicando sus características y principios para lograr el equilibrio entre la flexibilidad de que las solicitudes de cambio se pueden crear y aprobar en cualquier momento durante el proyecto, priorizándolas cuando se crea o se actualiza el Backlog Priorizado del Producto y la estabilidad permanente al refinar el Sprint Backlog no permitiendo la interferencia con el Equipo Scrum durante un sprint.

Lograr la flexibilidad

Scrum facilita la flexibilidad a través de la transparencia, la inspección y la adaptación para lograr los resultados de negocio más valiosos. La flexibilidad en Scrum se logra a través de cinco características claves:

- **Desarrollo de productos iterativos**

El enfoque iterativo e incremental de desarrollo de productos y servicios para incorporar cambios en cualquier paso en el proceso de desarrollo. Las solicitudes de cambio para el proyecto pueden provenir de múltiples fuentes de la siguiente manera: stakeholders, equipo principal de Scrum o alta gerencia.

- **Asignación de un time-box**

Asignar time-boxes proporciona la estructura necesaria para los proyectos Scrum, los cuales tienen un elemento de incertidumbre, son dinámicos por naturaleza y propensos a cambios frecuentes. Estos bloques de tiempo ayudan a medir el progreso del proyecto y permiten que el equipo identifique fácilmente cuándo se necesitará modificar un proceso o método.

- **Equipos interfuncionales**

Permiten a los miembros del equipo enfocarse en los resultados deseados del sprint. El equipo tiene objetivos definidos durante cada sprint y la flexibilidad para realizar cambios en los objetivos antes de iniciar el siguiente sprint. Ser auto organizados garantiza que los miembros del Equipo Scrum decidan como hacer el trabajo del proyecto sin la gestión de las tareas por un alto directivo.

- **Priorización basada en el cliente**

Se determina con base al valor proporcionado al cliente. Primero, al inicio de un proyecto, se documenta en Backlog Priorizado del Producto los requisitos iniciales son priorizados en función del valor que cada requisito proveerá. Si se realiza una solicitud para un requisito nuevo o cambio a uno ya existente, se evalúa durante el proceso de Refinar el Backlog Priorizado del Producto y si se considera que el cambio proporcionará más valor que otros requisitos existentes, se añadirá y priorizará de acuerdo a la versión actualizada del Backlog Priorizado del Producto.

- **Integración continua**

Para reducir el riesgo de que varios miembros del equipo hagan cambios en componentes redundantes, los miembros del Equipo Scrum pueden incorporar características nuevas y modificadas en las entregas siempre que sea posible.

5. Riesgo

Se define como eventos inciertos que afectan los objetivos de un proyecto y pueden contribuir a su éxito o fracaso. Los riesgos que tienen un impacto positivo en el proyecto se les conoce como oportunidades, mientras que las amenazas son riesgos que afectan negativamente al proyecto. Los riesgos deben ser identificados, evaluados y atendidos con base a dos factores: la probabilidad de ocurrencia y el impacto en el caso de tal ocurrencia.

Procedimiento de gestión de riesgos

Se compone de cinco pasos, mismos que deben llevarse a cabo en forma iterativa durante el proyecto:

- Identificación de riesgos

Utilizar las siguientes técnicas para identificar todos los riesgos potenciales:

- Revisar las lecciones aprendidas de Retrospectiva del sprint o del proyecto
- Listas de verificación de riesgos
- Lista corta de riesgos
- Lluvia de ideas
- Estructura de distribución de riesgos

- Evaluación de riesgos

Se lleva a cabo en relación a la probabilidad, proximidad e impacto. La probabilidad de riesgo es la probabilidad de su ocurrencia, mientras que la proximidad se refiere a cuándo pudiera suscitarse un riesgo. El impacto es el efecto probable del riesgo sobre un proyecto u organización. Las técnicas de evaluación de riesgos que se pueden utilizar son las siguientes:

- Reunión de riesgos
- Árboles de probabilidad
- Análisis de Pareto
- Cuadrícula de probabilidad e impacto
- Valor monetario esperado

- Priorización de riesgos

En la figura 42 se observa como dar prioridad al riesgo que habrá de incluirse en el Backlog Priorizado del Producto.

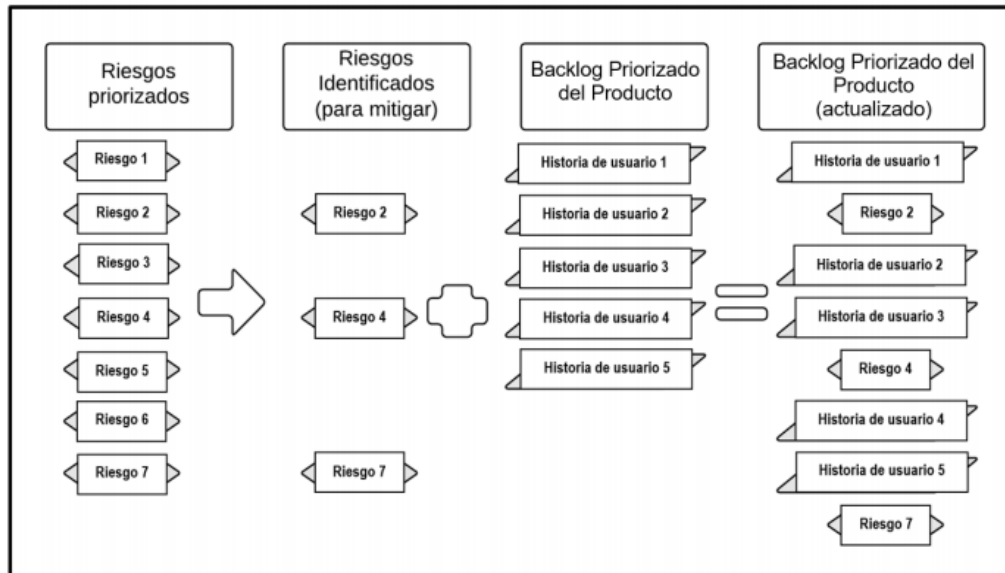


Figure 42 Proceso de priorización de riesgos (SCRUMstudy, 2017)

- Mitigación de riesgos

La naturaleza iterativa de Scrum con sus ciclos rápidos de respuesta y retroalimentación permite que las fallas se detecten de forma temprana, por lo tanto, tiene una función de mitigación natural construida dentro del sistema.

- Comunicación de riesgos

Se proporciona a los stakeholders información relacionada al riesgo donde se incluya el impacto potencial, así como los planes para enfrentar cada riesgo. La comunicación siempre está en curso y debe ocurrir a la par de los cuatro pasos secuenciales discutidos hasta ahora: identificación, evaluación, priorización y mitigación de riesgos.

Minimización de riesgos por medio de Scrum

Al ser un proceso ágil e iterativo, el framework de Scrum minimiza inherentemente el riesgo. Las siguientes prácticas de Scrum facilitan la gestión efectiva del riesgo:

- La flexibilidad reduce el riesgo relacionado al entorno empresarial.
- La retroalimentación reduce el riesgo relacionado a las expectativas.
- La propiedad del equipo reduce la estimación de riesgo.
- La transparencia reduce el riesgo de no detección.
- La entrega iterativa reduce el riesgo de inversión.

Procesos de Scrum

Son las actividades específicas y el flujo de un proyecto de Scrum tal como se observa en la en las siguientes tablas y figuras.

Tabla 16 Proceso de Scrum Inicio (SCRUMstudy, 2017)

Fases y Procesos fundamentales	
Inicio	
<p>1. Crear la visión del proyecto</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Caso de negocio del proyecto <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Reunión de la visión del proyecto <p>Salidas</p> <ul style="list-style-type: none"> 1.- Product Owner identificado 2.- Reunión de la visión del proyecto 	<p>2. Identificar al Scrum Master y al stakeholder</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Product Owner 2.- Reunión de la visión del proyecto <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Criterios de selección <p>Salidas</p> <ul style="list-style-type: none"> 1.- Scrum Master identificado 2.- Stakeholder identificado
<p>3. Formar el equipo Scrum</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Product Owner 2.- Scrum Master 3.- Reunión de la visión del proyecto <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Selección del equipo Scrum <p>Salidas</p> <ul style="list-style-type: none"> 1.- Equipo Scrum identificado 	<p>4. Desarrollar épicas</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Declaración de la visión del proyecto <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Reuniones de grupo de usuarios <p>Salidas</p> <ul style="list-style-type: none"> 1.- Épicas 2.- Prototipos
<p>5. Crear el Backlog Priorizado del Producto</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Épicas 3.- Prototipos <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Métodos de priorización de historias de usuario <p>Salidas</p> <ul style="list-style-type: none"> 1.- Backlog Priorizado del Producto 2.- Criterios de terminado 	<p>6. Realizar la planificación del lanzamiento</p> <p>Entradas</p> <ul style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Stakeholders 3.- Declaración de la visión del proyecto 4.- Backlog Priorizado del Producto 5.- Criterios de terminado <p>Herramientas</p> <ul style="list-style-type: none"> 1.- Sesiones de planificación del lanzamiento 2.- Métodos de priorización del lanzamiento <p>Salidas</p> <ul style="list-style-type: none"> 1.- Cronograma de planificación del lanzamiento 2.- Duración del sprint

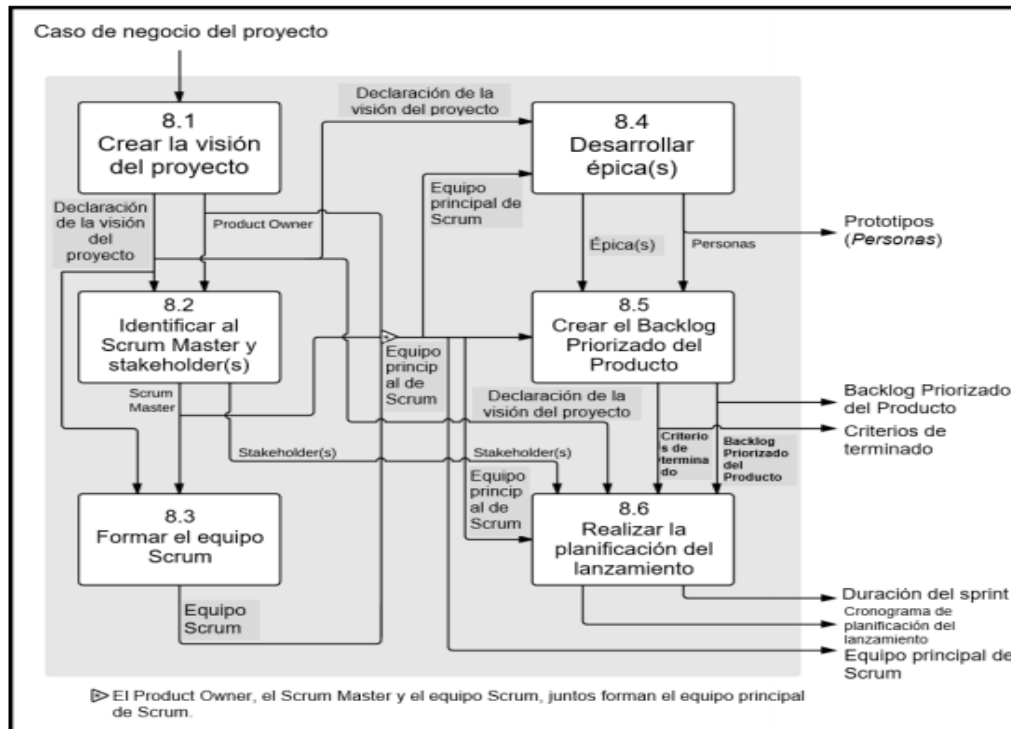


Figura 43 Flujo de datos de la fase de Inicio (SCRUMstudy, 2017)

Tabla 17 Proceso de Scrum Planificación y estimación (SCRUMstudy, 2017)

Fases y Procesos fundamentales	
Planificación y estimación	
<p>7. Crear historias de usuario</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Backlog Priorizado del Producto 3.- Criterios de terminado 4.- Prototipos <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Experiencia en la redacción de historias de usuario <p>Salidas</p> <ol style="list-style-type: none"> 1.- Historias de usuario 2.- Criterios de aceptación de historias de usuario 	<p>8. Estimar historias de usuario</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Historias de usuario <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Métodos de estimación <p>Salidas</p> <ol style="list-style-type: none"> 1.- Historias del usuario estimadas
<p>9. Comprometer historias de usuario estimadas</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Historias del usuario estimadas 3.- Duración del sprint 	<p>10. Identificar tareas</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Historias de usuario comprometidas

<p>Herramientas</p> <p>1.- Reuniones de planificación del sprint</p> <p>Salidas</p> <p>1.- Historias de usuario comprometidas</p>	<p>Herramientas</p> <p>1.- Reuniones de planificación del sprint</p> <p>Salidas</p> <p>1.- Lista de tareas</p>
<p>11. Estimar tareas</p> <p>Entradas</p> <p>1.- Equipo principal de Scrum</p> <p>2.- Lista de tareas</p> <p>Herramientas</p> <p>1.- Reuniones de planificación del sprint</p> <p>2.- Criterios de estimación</p> <p>3.- Métodos de estimación</p> <p>Salidas</p> <p>1.- Esfuerzo estimado para la lista de tareas</p>	<p>12. Crear el Sprint Backlog</p> <p>Entradas</p> <p>1.- Equipo principal de Scrum</p> <p>2.- Esfuerzo estimado para la lista de tareas</p> <p>3.- Duración del sprint</p> <p>Herramientas</p> <p>1.- Reuniones de planificación del sprint</p> <p>Salidas</p> <p>1.- Sprint Backlog</p> <p>2.- Sprint Burndown Chart</p>

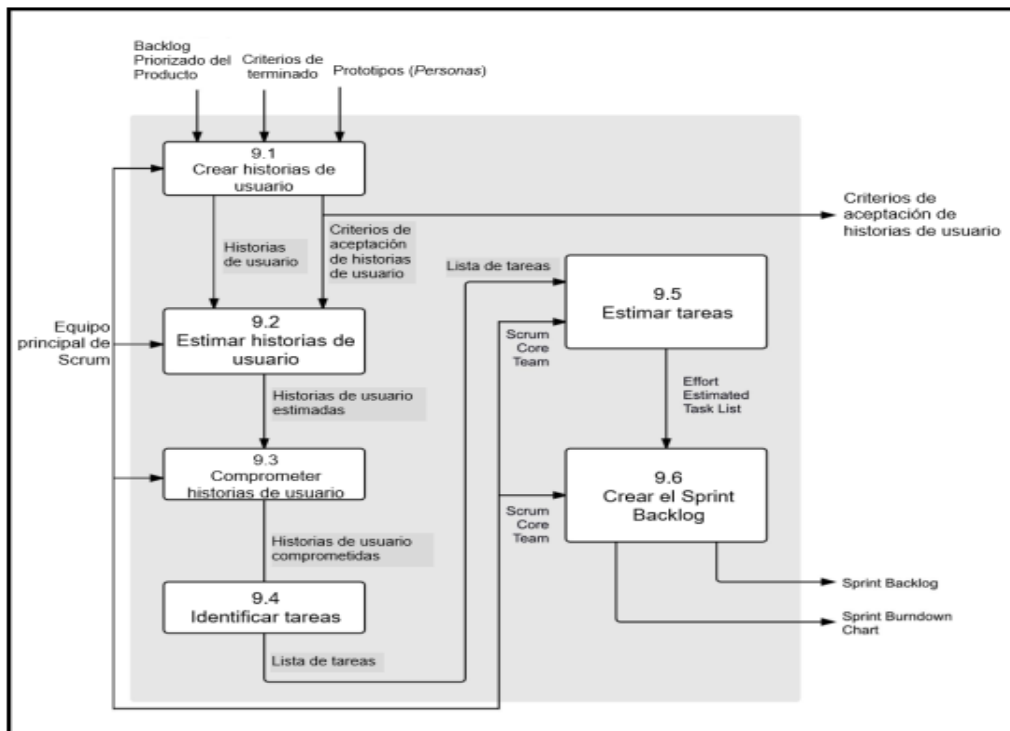


Figura 44 Flujo de datos de la fase de planificación y estimación (SCRUMstudy, 2017)

Tabla 18 Proceso de Scrum Implementación (SCRUMstudy, 2017)

Fases y Procesos fundamentales	
Implementación	
<p>13. Crear entregables</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Sprint Backlog 3.- Scrumboard 4.- Impediment log <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Experiencia del equipo <p>Salidas</p> <ol style="list-style-type: none"> 1.- Entregables del sprint 2.- Scrumboard actualizado 3.- Impediment log actualizado 	<p>14. Realizar Daily Standup</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo Scrum 2.- Scrum Master 3.- Sprint Burndown Chart 4.- Impediment log <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Daily Standup 2.- Tres preguntas diarias <p>Salidas</p> <ol style="list-style-type: none"> 1.- Sprint Burndown Chart actualizado 2.- Impediment log actualizado
<p>15. Refinar el Backlog Priorizado del Producto</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Backlog priorizado del producto <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reunión de revisión del Backlog Priorizado del Producto <p>Salidas</p> <ol style="list-style-type: none"> 1.- Backlog Priorizado del Producto actualizado 	

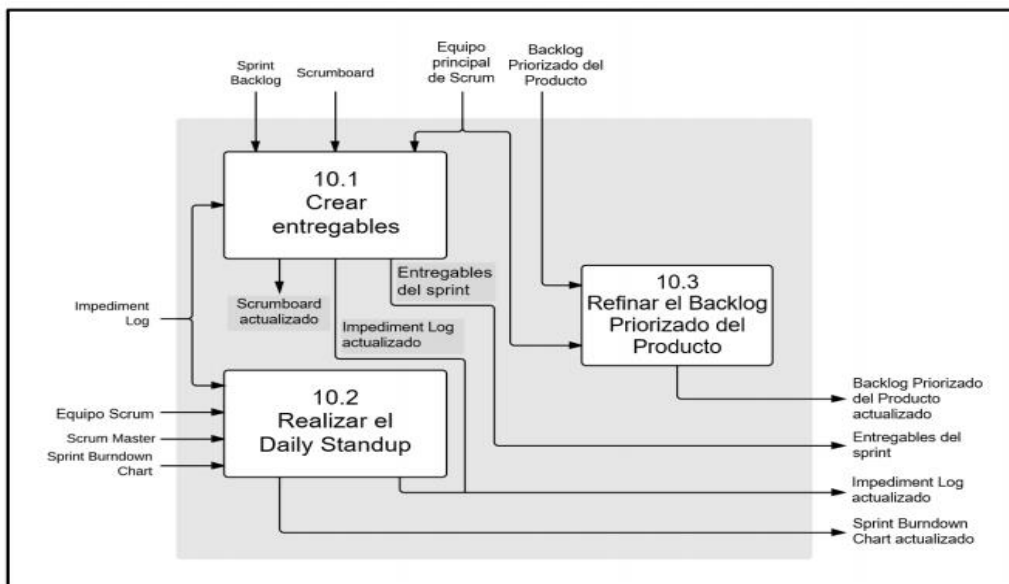


Figura 45 Flujo de datos de la fase de implementación (SCRUMstudy, 2017)

Tabla 19 Proceso de Scrum Revisión y retrospectiva (SCRUMstudy, 2017)

Fases y Procesos fundamentales	
Revisión y retrospectiva	
<p>16. Demostrar y validar el sprint</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum 2.- Entregables del sprint 3.- Sprint Backlog 4.- Criterios de terminado 5.- Criterios de aceptación de las historias del usuario <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reuniones de revisión del sprint <p>Salidas</p> <ol style="list-style-type: none"> 1.- Entregables aceptados 	<p>17. Retrospectiva del sprint</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Scrum Master 2.- Equipo Scrum 3.- Demostrar y validar el sprint <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reunión de retrospectiva del sprint <p>Salidas</p> <ol style="list-style-type: none"> 1.- Mejoras procesables acordadas

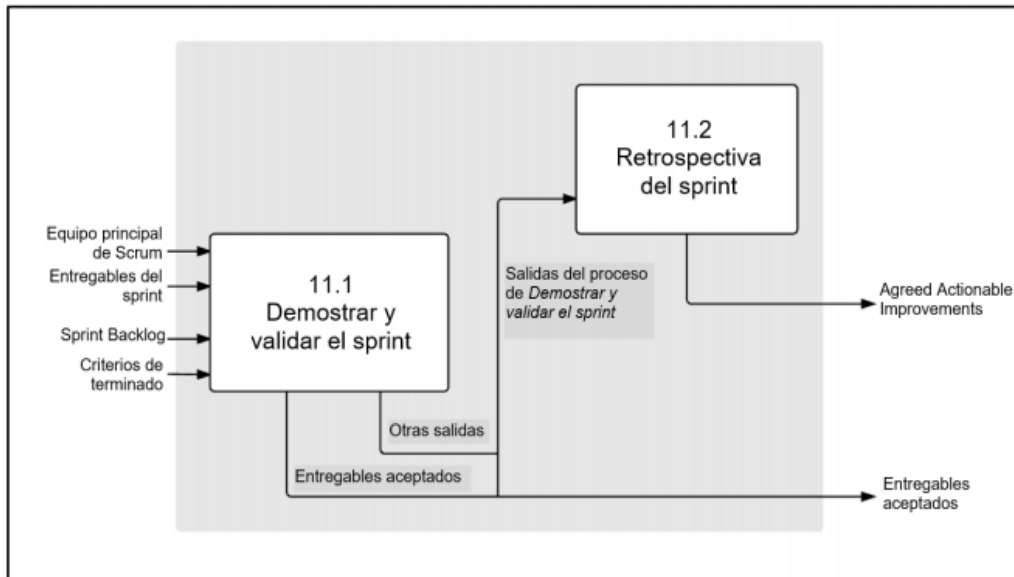


Figura 46 Flujo de datos de la fase de revisión y retrospectiva (SCRUMstudy, 2017)

Tabla 20 Proceso de Scrum Lanzamiento (SCRUMstudy, 2017)

Fases y Procesos fundamentales	
Lanzamiento	
<p>18. Enviar entregables</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Product Owner 2.- Stakeholders 3.- Entregables aceptados 	<p>19. Retrospectiva del proyecto</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principal de Scrum <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reunión de retrospectiva del proyecto

<p>4.- Cronograma de planificación del lanzamiento</p> <p>Herramientas</p> <p>1.- Métodos de desplazamiento organizacional</p> <p>Salidas</p> <p>1.- Acuerdo de entregables funcionales</p>	<p>Salidas</p> <p>1.- Mejoras procesables acordadas</p> <p>2.- Elementos de acción asignados y fechas limite</p>
---	--

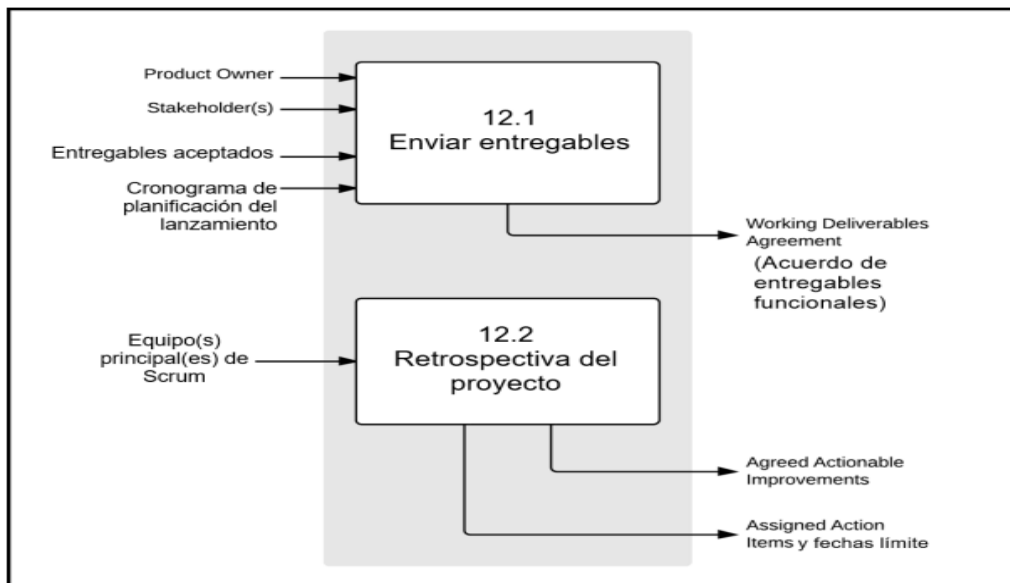


Figura 47 Flujo de datos de la fase de lanzamiento (SCRUMstudy, 2017)

Procesos adicionales

Se aplican en proyectos a gran escala que requieren coordinación entre múltiples equipos. Los motivos para definir procesos adicionales para grandes proyectos son:

- Mayor interacción y dependencias entre los equipos Scrum.
- La necesidad de colaborar en un equipo de Product Owners.
- Manejar conflictos, resolver problemas y fijar prioridades entre los equipos Scrum.
- Especialización para lograr realizar tareas específicas en un proyecto grande, ya que quizá no todos los equipos Scrum cuenten con tales habilidades.
- Definir ciertas normas y estándares que deben cumplir todos los equipos Scrum.
- Los requerimientos para establecer un ambiente en un proyecto grande que después utilizarán todos los equipos Scrum.

- Coordinar los resultados de varios equipos Scrum a fin de crear el lanzamiento de un proyecto grande.

En las tablas 21 y tabla 22 se puede observar la aplicación de Scrum para grandes proyectos y para la empresa.

Tabla 21 Scrum para grandes proyectos (SCRUMstudy, 2017)

Scrum para grandes proyectos	
<p>Scrum para grandes proyectos</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Declaración de la visión del proyecto 2.- Chief Product Owner 3.- Chief Scrum Master 4.- Identificar el ambiente 5.- Recomendaciones del Scrum Guidance Body <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reunión de plan de ambiente <p>Salidas</p> <ol style="list-style-type: none"> 1.- Plan de preparación del lanzamiento 	<p>Realizar y coordinar sprints</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipos principales 2.- Grandes equipos principales 3.- Definición de terminado 4.- Criterios de aceptación de historias de usuario <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reuniones de Scrum de Scrum 2.- Experiencia del equipo <p>Salidas</p> <ol style="list-style-type: none"> 1.- Entregables del sprint 2.- Cronograma de planificación del lanzamiento actualizado
<p>Preparar el lanzamiento de grandes proyectos</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Equipo principales 2.- Grandes equipos principales 3.- Cronograma de planificación del lanzamiento <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Planes de comunicación <p>Salidas</p> <ol style="list-style-type: none"> 1.- Producto enviable 	

Tabla 22 Scrum para la empresa (SCRUMstudy, 2017)

Scrum para la empresa	
<p>Crear componentes de programa o portafolio</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Misión y visión de la compañía 2.- Portfolio Product Owner 3.- Portfolio Scrum Master 4.- Program Product Owner 	<p>Revisar y actualizar el Scrum Guidance Body</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Regulaciones 2.- Mejoramientos recomendados del Scrum Guidance Body

<p>5.- Program Scrum Master</p> <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Planes de comunicación 2.- Planes de recursos humanos de la empresa <p>Salidas</p> <ol style="list-style-type: none"> 1.- Criterios mínimos de terminado 2.- Criterios de aceptación de historias de usuario 3.- Recursos compartidos 4.- Stakeholders identificados 	<p>Herramientas</p> <ol style="list-style-type: none"> 1.- Criterios de selección de miembros <p>Salidas</p> <ol style="list-style-type: none"> 1.- Recomendaciones actualizadas del Scrum Guidance Body 				
<p>Crear y refinar el backlog del portafolio</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Misión y visión de la compañía 2.- Backlog Priorizado del Portafolio 3.- Backlog Priorizado del Programa 4.- Portfolio Product Owner 5.- Portfolio Scrum Master 6.- Program Product Owner 7.- Program Scrum Master <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reuniones de revisión del Backlog Priorizado del Programa o el Portafolio 2.- Técnicas de comunicación <p>Salidas</p> <ol style="list-style-type: none"> 1.- Backlog del Programa o el Portafolio Priorizado Actualizado 2.- Mejoramientos recomendados del Scrum Guidance Body 3.- Fechas límite de implementación actualizadas para los proyectos 	<p>Coordinar los componentes del portafolio</p> <p>Entradas</p> <ol style="list-style-type: none"> 1.- Definición de terminado 2.- Dependencias conocidas 3.- Backlog Priorizado del Programa o Portafolio 4.- Portfolio Product Owner 5.- Portfolio Scrum Master 6.- Program Product Owner 7.- Program Scrum Master <p>Herramientas</p> <ol style="list-style-type: none"> 1.- Reunión de Scrum de Scrums <p>Salidas</p> <ol style="list-style-type: none"> 1.- Registro de impedimentos actualizados 2.- Dependencias actualizadas 				
<p style="text-align: center;">Retrospectiva de lanzamientos del portafolio</p> <p style="text-align: center;">Entradas</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td>1.- Portfolio Product owner</td> <td>2.- Portfolio Scrum Master</td> </tr> <tr> <td>3.- Program Product Owner</td> <td>4.- Program Scrum Master</td> </tr> </table> <p style="text-align: center;">Herramientas</p> <ol style="list-style-type: none"> 1.- Reuniones de retrospectiva del programa o portafolio <p style="text-align: center;">Salidas</p> <ol style="list-style-type: none"> 1.- Mejoras acordadas 2.- Elementos de acción asignados y fechas limite 		1.- Portfolio Product owner	2.- Portfolio Scrum Master	3.- Program Product Owner	4.- Program Scrum Master
1.- Portfolio Product owner	2.- Portfolio Scrum Master				
3.- Program Product Owner	4.- Program Scrum Master				

Anexo E: Análisis de contenido generado por el usuario

En este anexo se explica brevemente las fases generales de la mayoría de los sistemas Big Data además de las cuatro etapas con paradigma de procesamiento por lotes de las que consta el caso de estudio referente al análisis de contenido generado por usuarios en la ciudad de Barcelona.

Fases de sistemas Big Data.

De forma general, en la figura 48 se observan las fases identificables en el ciclo de vida de la mayoría de los sistemas Big Data. Etapas que permiten a las empresas visualizar toda la cadena de valor de los datos.

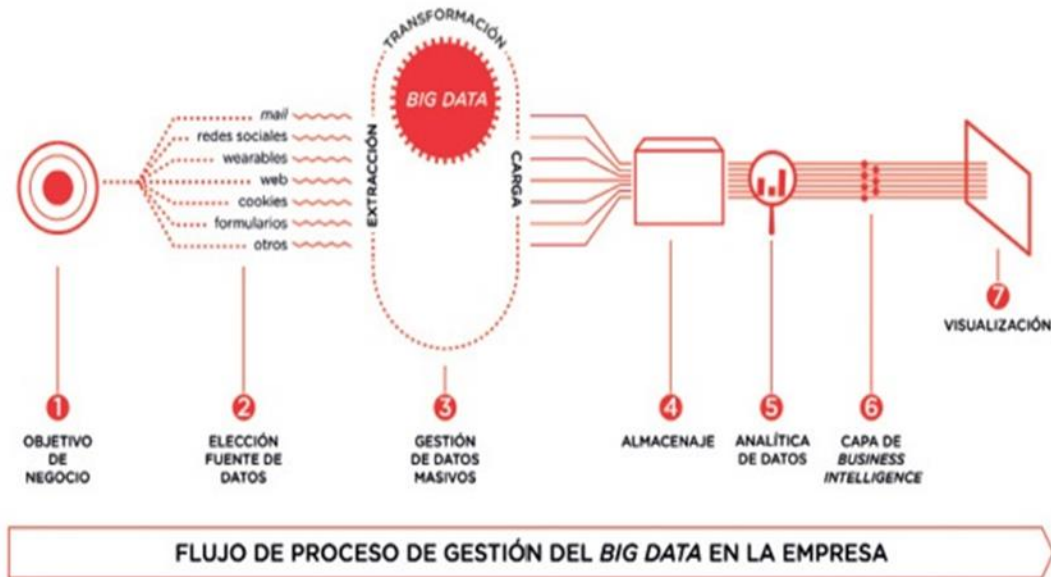


Figura 48 Fases generales de Big Data (Fernández-Manzano, Neira, & Clares-Gavilán, 2016)

El objetivo del caso de estudio es analizar el comportamiento de la imagen transmitida en línea de las redes sociales de Barcelona de más de 100,000 blogs de viajes relevantes escritos en inglés por turistas que han visitado la ciudad durante el periodo 2005-2015.

El caso de estudio de análisis de contenido generado por el usuario de la ciudad de Barcelona como se observa en la figura 49, consta de cuatro etapas las cuales tienen como base el paradigma de procesamiento por lotes, primero se almacenan los datos, después son analizados.

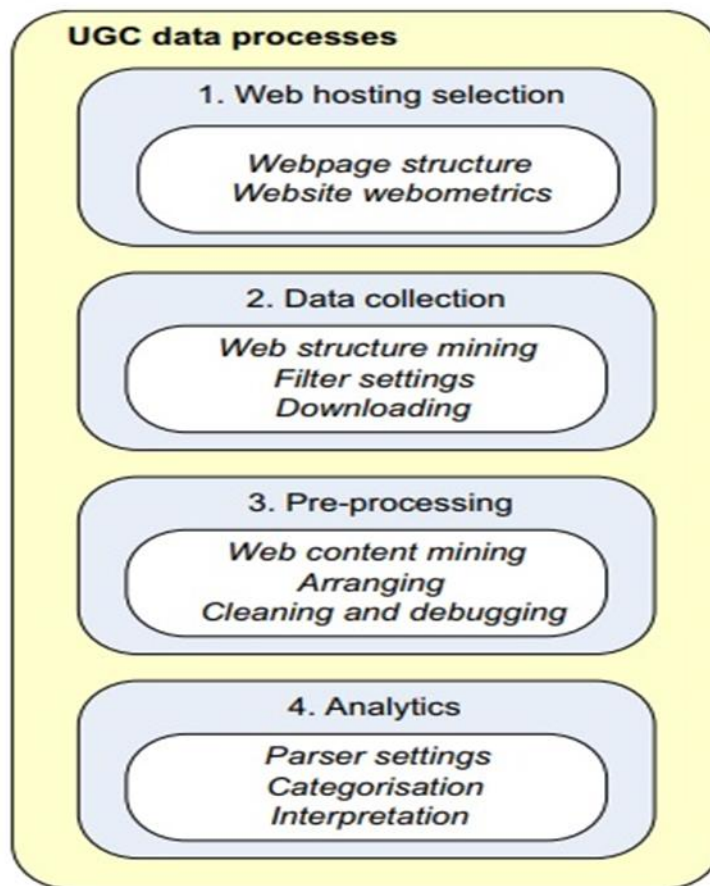


Figura 49 Etapas de procesamiento UGC (Marine-Roig & Anton Clavé, 2015)

Etapa 1. Selección de alojamiento web.

Se seleccionan los contenidos generados por el usuario.

- Estructura del sitio web.

Se seleccionan fuentes del contenido generado por usuarios que tienen en su estructura de informatización la recopilación de datos y disposición sistemática de la información. El área de turismo tiene como atributos principales: post-título, destino y fecha de viaje.

- Web-metría del sitio web.

Se utilizaron dos motores de búsqueda, con la consulta 'blog de viajes' O 'revisión de viajes' el 1 de marzo de 2015, se obtuvieron 15.800.000 resultados en BlogSearch.Google.com y se obtuvieron 1.880.000 resultados en Bing.com. A continuación, se seleccionaron los sitios web ubicados en la primera posición, siempre que cumplieran las condiciones mencionadas anteriormente y contuvieran más de 100 entradas relacionadas con el estudio de caso. Once sitios web se incluyeron en esta primera selección: GetJealous.com, MyTripJournal.com, StaTravel.com, TravBuddy.com, TravelBlog.org (TB), TravelJournals.net, TravellersPoint.com, TravelPod.com (TP), TripAdvisor.com (TA), Venere.com y VirtualTourist.com (VT).

Etapa 2. Recopilación de datos.

Localización y descarga de las páginas web seleccionadas.

- Minería de estructuras web.

Para recopilar datos, se identificaron los hipervínculos de las páginas web relevantes relacionadas con el estudio de caso: blogs de viajes, páginas de viajes, diarios de viaje y reseñas de viajes sobre Barcelona.

- Configuraciones de filtro.

Con la información de los hipervínculos, se descargó la cantidad mínima posible de archivos, para no sobrecargar el tráfico de alojamiento web y ahorrar espacio en el disco duro local. Se aplican los siguientes filtros:

- Filtro de nivel.

Se refiere al número de clics del mouse en los hipervínculos que son necesarios para pasar de la página de inicio a la última página necesaria.

- Filtro de tipo de archivo.

Permite restringir la descarga por extensión de nombre de archivo.

- Filtro de URL.

El filtro uniforme de localización de recursos (URL) permite actuar sobre cualquiera de sus partes utilizando palabras clave incluidas y excluidas: protocolo (http, https, ftp), servidor, dominio, subdirectorios (carpetas) y nombres de archivo.

- Filtro de contenido.

Permiten al usuario especificar las palabras clave separadas con símbolos espaciales y secuencias de palabras exactas para buscar en todo el texto de las páginas descargadas o dentro de etiquetas HTML.

- Descarga de datos.

Se descargan las páginas web relacionadas con el estudio de caso utilizando la copiadora web Offline Explorer Enterprise (OEE) por su capacidad para descargar hasta 100 millones de URL por proyecto. También ofrece el procesamiento multiproceso más rápido posible de archivos descargados mediante el uso de todos los núcleos de CPU.

Etapa 3. Pre procesamiento de datos.

Se identifican diferentes campos de texto, texto de anclaje y bloques de contenido principal, así como eliminar etiquetas HTML. Se extrae información de las páginas web descargadas, lo que permite estructurar la ruta local a los archivos HTML para que puedan ordenarse y segmentarse en diferentes criterios, como destino, fecha, idioma, nacionalidad y tema.

- Minería de contenido web.

Se extrae información útil de los contenidos de la página web.

- Encontrar la ciudad natal del usuario.

Dado que todos los destinos tienen una guía de viaje con un hipervínculo estructurado geográficamente, se puede automatizar la extracción del país de origen del viajero para poder clasificarlos por nacionalidad.

- Detección de lenguaje.

Se utiliza un programa Java basado en la Biblioteca de detección de idiomas (LDL) el cual detecta hasta 53 idiomas. Este sistema tiene como base el clasificador Naive Bayes LDL, capaz de detectar cada idioma con un grado de precisión superior al 99%. El análisis se debe realizar en el contenido textual sin etiquetas HTML después de la etapa de limpieza, porque lo que el usuario ha escrito realmente representa solo una parte mínima del contenido de la página web.

- Organización de datos.

El proceso de descarga almacenó archivos en el disco duro local, siguiendo una estructura paralela a la del servidor web. Para identificar y clasificar los blogs y OTR de viajes, la estructura de los archivos se organizó utilizando el siguiente patrón de carpetas y archivos:

```
\ host \ brand \ town \ date_lang_isFrom_pageId_topic.htm
```

Las barras diagonales inversas del patrón representan subdirectorios o carpetas, los guiones unen las palabras compuestas, y los guiones bajos separan las diferentes palabras o números del nombre del archivo. Cada elemento también sigue un patrón para facilitar la manipulación de archivos:

- anfitrión.

Acrónimo de dos letras que representa el sitio web de alojamiento (Ejemplo, TB: TravelBlog.org, VT: VirtualTourist.com).

- marca.

Abreviatura de cinco letras (Ejemplo: CBarc: Costa Barcelona [Costa de Barcelona]; pBarc: Paisatges Barcelona [paisajes de Barcelona]) de la marca territorial.

- pueblo.

Pueblo o ciudad de destino (Ejemplo: L'Hospitalet de Llobregat: L-Hospitalet-de-Llobregat)

- fecha.

Fecha de blog de viaje u OTR en formato AAAAMMDD, basado en el estándar ISO 8061, que permite el ordenamiento numérico o alfabético de los archivos por fechas.

- lang.

Código de idioma de dos letras según la norma ISO 639-1 (Ejemplo: Ca: catalán, en: inglés). Si no se pudo detectar el idioma (Ejemplo: entrada multilingüe o texto insuficiente), se usó el código "xx".

- isFrom.

Código de país de dos letras de acuerdo con el estándar ISO-3166-2 (Ejemplo: GB: Reino Unido, EE. UU: Estados Unidos de América). Aunque los estados son únicos, algunos usuarios escriben su ciudad o región solamente, y no pueden clasificarse debido al problema de los homónimos, especialmente en países de habla inglesa. En casos así o cuando la ciudad natal no está disponible, podemos escribir el código "ZZ".

- pageld.

El sitio web identificado debe ser único. En el caso del ejemplo que se muestra: código de destino, código de miembro y cuaderno de viaje o código OTR separados por guiones bajos.

- tema.

Abreviatura de cuatro letras del tema de la OTR. Las clasificaciones más comunes de OTR son cosas que hacer, hoteles y restaurantes.

- Limpieza de datos.

El objetivo de esta fase es preservar el contenido de las páginas web generado por el usuario sin modificar su formato HTML y eliminar contenido innecesario como menús de navegación, anuncios, avisos de derechos de autor que no tienen relación con la imagen percibida del destino que este estudio pretende analizar. Para esta actividad se utilizó Microsoft Expression Web 4.

- Depuración de datos.

El idioma catalán utiliza caracteres especiales como cedillas y vocales con acentos, que son problemáticos para la codificación en sitios web y desconcertantes para los usuarios de habla inglesa. Las codificaciones especiales y los errores ortográficos distorsionan el análisis y tuvieron que corregirse, especialmente aquellos que afectan los nombres de destinos y atracciones, y los nombres propios en general como problemas de codificación de caracteres, errores ortográficos y traducciones.

Etapa 4. Análisis de contenido.

El análisis cuantitativo del contenido se realizó a través de un analizador de contenido del sitio web, Site Content Analyzer (SCA) el cual analiza en línea y sin conexión las palabras clave, sugiere frases relevantes y analiza la estructura de enlaces. SCA descubre la frecuencia, densidad y peso de las palabras clave para cada página web. Posteriormente, las palabras clave se agrupan en categorías para analizar la imagen percibida de los usuarios.

- Configuración del analizador.

SCA permite la configuración de muchas opciones como delimitadores y zonas de análisis, pero las configuraciones más interesantes son la lista negra, las palabras compuestas y el peso de la etiqueta y la posición.

- Lista negra.

En los documentos, hay muchas palabras insignificantes que ayudan a construir oraciones, pero no representan ningún contenido. Estas palabras deben identificarse y eliminarse antes de indexar los documentos. La lista negra se usa para incluir estas palabras como algunos adverbios, artículos, preposiciones, pronombres y conjunciones.

- Lista de palabras compuestas.

Consiste en grupos de palabras que tienen un significado conjunto. Las palabras compuestas tienen prioridad sobre la lista negra y sobre palabras clave simples. Por ejemplo, la 'Basílica de la Sagrada Familia' conserva la 'de' y la 'la' a pesar de estar incluida en la lista negra, y tiene preferencia sobre las palabras clave simples 'Basílica', 'Sagrada' y 'Familia'.

- Peso de la palabra clave.

El peso indica la importancia o visibilidad de la palabra clave o frase clave. Por ejemplo, se ha demostrado en estudios sobre flujo de clics y patrones de navegación que el contenido más difundido es el que se encuentra en la parte superior de una página web, en lugar de en el área inferior. SCA asigna un peso a las palabras clave de acuerdo con su posición y la etiqueta HTML que define su formato y características.

- Categorización.

Para el análisis de contenido cuantitativo se obtuvo la frecuencia de todas las palabras clave únicas. Estas palabras clave deben agruparse en categorías para permitir que se extraiga el conocimiento de los datos.

La investigación de análisis de contenido cuantitativo puede basarse en dos modelos de categorización: categorías preestablecidas o categorías discernidas del texto mismo.

Basado en trabajos anteriores y análisis de frecuencia preliminares se desarrollaron ocho amplias categorías relacionadas con factores de atracción, para analizar el componente cognitivo de la imagen de destino:

- Comida y Vino
- Patrimonio inmaterial
- Ocio y recreación
- Naturaleza y turismo activo
- Deportes
- Sol, mar y arena
- Patrimonio tangible
- Medio ambiente urbano.

Anexo F: Mapeo Big Data-Metodologías Ágiles

En este anexo, se presentan características de las cuatro metodologías ágiles investigadas para este trabajo, las cuales dan ventaja a la Scrum como la metodología ágil ideal para gestionar cualquier tipo de proyectos en las empresas.

Mapeo Big Data y Metodologías Ágiles

Debido a que los sistemas Big Data tienen la complejidad de manejar grandes cantidades de datos generados con rapidez casi en tiempo real y con múltiples fuentes de datos, las metodologías ágiles son una posible opción ya que cuentan con las características generales, explicadas en secciones anteriores, necesarias para ser adaptadas al desarrollo de sistemas Big Data.

Con base en los cuatro puntos de vista para clasificar métodos (Iacovelli & Souveyet, 2008): “uso”, “agilidad”, “aplicabilidad” y “productos”. En las tablas 23-26 se observa la revisión de los criterios de comparación entre las cuatro metodologías ágiles más implementadas en organizaciones de negocios.

Tabla 23 Punto de vista: Uso (Iacovelli & Souveyet, 2008)

Punto de vista	Aspecto / Método	Lean	Kanban	Xp	Scrum
Uso	Respeto a fechas de entrega	Falso	Falso	Falso	Cierto
	Cumplimiento de requisitos	Cierto	Cierto	Cierto	Cierto
	Satisfacción del usuario	Cierto	Falso	Falso	Cierto
	Entornos complicados	Cierto	Cierto	Cierto	Cierto
	Favorable ante el caos	Falso	Falso	Falso	Cierto
	Aumento de productividad	Cierto	Cierto	Cierto	Cierto

Tabla 24 Punto de vista: Agilidad (Iacovelli & Souveyet, 2008)

Punto de vista	Aspecto / Método	Lean	Kanban	Xp	Scrum
Agilidad	Iteraciones cortas	Cierto	Cierto	Cierto	Cierto
	Colaboración	Cierto	Cierto	Cierto	Cierto
	Centrado en personas	Cierto	Cierto	Cierto	Cierto
	Integración de cambios	Cierto	Cierto	Cierto	Cierto
	Peso ligero	Cierto	Cierto	Cierto	Cierto
	Cambio de requisitos funcionales	Cierto	Cierto	Cierto	Cierto

Tabla 25 Punto de vista: Aplicabilidad (Iacovelli & Souveyet, 2008)

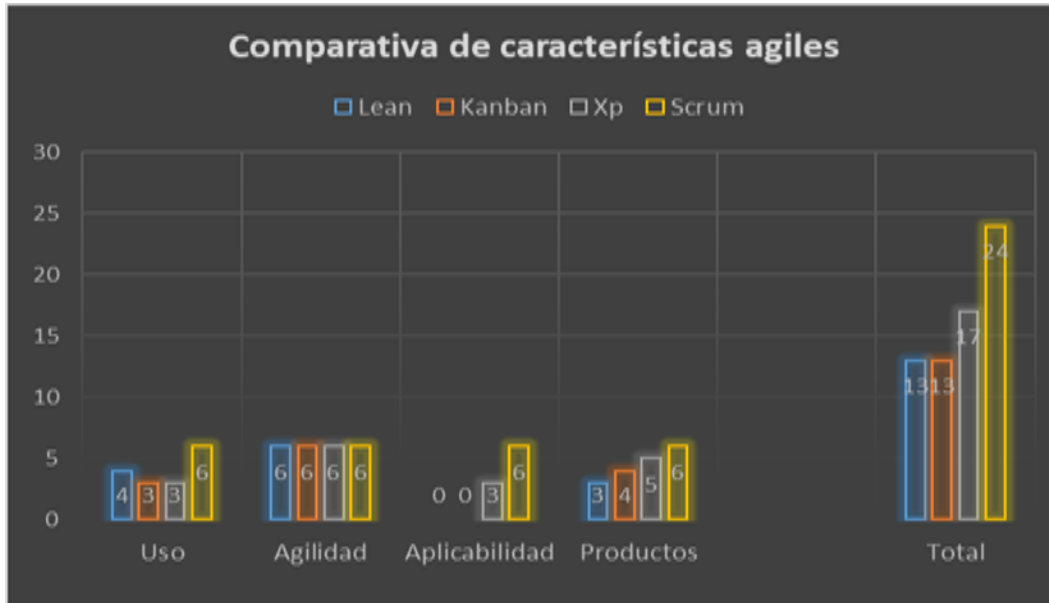
Punto de vista	Aspecto / Método	Lean	Kanban	Xp	Scrum
Aplicabilidad	Tamaño del proyecto	Pequeño	Pequeño	Pequeño	Pequeño/Grande
	Complejidad del proyecto	Baja	Baja	Baja	Alta
	Grado de interacción con el cliente	Baja	Baja	Alta	Alta
	Grado de interacción con usuarios finales	Baja	Baja	Baja	Alta
	Grado de interacción entre miembros del equipo	Baja	Baja	Alta	Alta
	Grado de integración de la novedad	Baja	Baja	Alta	Alta

Tabla 26 Punto de vista: Productos (Iacovelli & Souveyet, 2008)

Punto de vista	Aspecto/Método	Lean	Kanban	Xp	Scrum
Productos	Gestión de proyectos	Falso	Falso	Falso	Cierto
	Definición de requisitos	Falso	Falso	Cierto	Cierto
	Código	Cierto	Cierto	Cierto	Cierto
	Pruebas unitarias	Cierto	Cierto	Cierto	Cierto
	Pruebas de integración	Cierto	Cierto	Cierto	Cierto
	Pruebas de sistema	Cierto	Cierto	Cierto	Cierto
	Modelos de diseño	Falso	Falso	Falso	Cierto

Los resultados que se muestran en las tablas anteriores permiten observar que la metodología ágil Scrum es la mejor evaluada respecto a los atributos de los cuatro puntos de vista propuestos por (Iacovelli & Souveyet, 2008) para clasificar métodos.

En la gráfica 8 se observan los resultados comparativos entre los cuatro puntos de vista para clasificar métodos respecto a las cuatro metodologías ágiles más utilizadas entre organizaciones. Por otra parte, en la figura 50, se observa las diversas empresas que en México implementan en sus procesos de negocios la metodología ágil Scrum.



Grafica 8 Características ágiles



Figura 50 Empresas en México aplicando Scrum (SCRUMstudy, 2017)

Anexo G: Documentos

En este anexo, se adjuntos los ejemplos de documentos:

- ** Acuerdo de entregables funcionales.
- ** Reunión de retrospectiva.

Análisis Big Data de turismo en Barcelona, España

Acta: Acuerdo de entregables funcionales

Cuernavaca, Morelos a 30 de noviembre de 2019

El presente documento es para avalar la entrega de las historias de usuario realizadas las cuales se pueden observar en la siguiente tabla.

Entregables		
Historia de usuario	Descripción	Épica
HU01	Páginas web de turismo estructuradas	Búsqueda de páginas web de turismo.
HU02	Páginas web de turismo seleccionada.	
HU03	Páginas web de turismo configuradas.	Selección y descarga de datos de turismo.
HU04	Páginas web de turismo descargadas.	
HU05	Datos de páginas web de turismo extraídos.	Procesamiento de datos de turismo.
HU06	Páginas web de turismo organizadas.	
HU07	Datos configurados de páginas web de turismo.	Análisis de contenido web de turismo.
HU08	Palabras clave de páginas web categorizadas.	

Firman por constancia la entrega-recepción los encargados del acto protocolario de entrega-recepción del análisis Big Data de turismo de Barcelona, España.

Entrega:

Recibe

Lic. Antonio Juan Capistran Abundez

Dr. Juan Carlos Rojas Pérez

Análisis Big Data de turismo en Barcelona, España

Acta: Reunión de retrospectiva.

Cuernavaca, Morelos a 30 de noviembre de 2019

El presente documento es para revisar tres aspectos, lo que salió bien durante el proyecto (aciertos), lo que no salió tan bien (errores) y las mejoras que pudieran hacerse en el próximo proyecto para evitar errores y mantener aciertos. En la siguiente tabla se puede observar las conclusiones de la reunión retrospectiva del proyecto.

Reunión retrospectiva del proyecto		
¿Qué salió bien durante el proyecto? (aciertos)	¿Qué no salió bien durante el proyecto? (errores)	¿Qué mejoras vamos a implementar en el próximo proyecto? (recomendaciones de mejora continua)
<ul style="list-style-type: none">Comunicación entre Scrum Master y Equipo de desarrollo.Desarrollo de sprints.	<ul style="list-style-type: none">Cocimiento apropiado de la metodología Scrum.Conocimiento del área de turismo.	<ul style="list-style-type: none">Conocer proyectos similares.Estudio previo del tema de análisis Big Data