

Universidad Autónoma de Baja California

Maestría y Doctorado en Ciencias e Ingeniería



“Modelo Integral de Atributos Para la Estimación de Historias de Usuario en Scrum”

Tesis

Para obtener el grado de:

Maestro en Ciencias

Presenta: Mayra Janeth Durán Rodríguez

Director de Tesis: Dr. J. Reyes Juárez Ramírez

Co-Director: Dr. Luis Guillermo Martínez Méndez

Tijuana, Baja California, agosto del 2020

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

FOLIO No. 294

Tijuana, B. C., a 3 de Agosto del 2020

C. Mayra Janeth Durán Rodríguez
Pasante de: **Maestro en Ciencias**
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

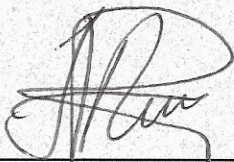
Es propuesto, por los C.C. Dr. J. Reyes Juárez Ramírez

Dr. Luis Guillermo Martínez Méndez

Quien será el responsable de la calidad de trabajo que usted presente, referido al
tema Modelo Integral de Atributos para la Estimación de Historias de Usuario en
Scrum.

El cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- ESTADO DEL ARTE
- III.- MODELO PROPUESTO
- IV.- PRUEBAS DEL MODELO
- V.- CONCLUSIONES, LIMITACIONES, APORTACIONES
Y TRABAJO FUTURO
- VI.- BIBLIOGRAFIA
- VII.- APÉNDICES

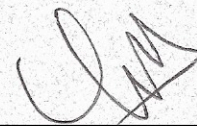


Dr. J. Reyes Juárez Ramírez
Director de Tesis

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



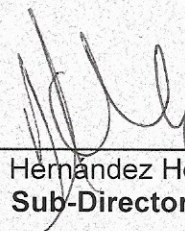
FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA
COORDINACIÓN DE
INVESTIGACIÓN Y POSGRADO



Dr. Luis Guillermo Martínez Méndez
Co- Director de Tesis



Dr. José Luis González Vázquez
Director



Q. Noemí Hernández Hernández
Sub-Directora

Dedicatoria

A Mis padres:

Luz Elena Rodríguez Esparza y Carlos Durán Gómez

*Con mucho amor y cariño, les dedico todo mi esfuerzo.
Por haberme dado la vida, por motivarme a ser mejor cada día, por su
apoyo incondicional en todo momento y por tantas cosas más, jamás
me cansare de agradecerles por todo lo que han hecho por mí.*

Agradecimientos

A dios y a la vida por permitirme concluir este gran sueño, por regalarme momentos felices y por darme valor, perseverancia y fortaleza para afrontar momentos difíciles.

A mis padres, porque son el mejor regalo que la vida me pudo dar, por su amor, por su apoyo durante toda mi carrera profesional, por sus consejos, por ser mi guía y ejemplo, por hacer de mi la persona que hoy soy, gracias por todo.

A mi familia, por siempre estar en los momentos más importantes, en especial a mi hermano Juan Durán por siempre darme palabras de apoyo, a mi sobrino José Enrique Cisneros, quien es mi motivación e inspiración y llena de alegría cada día de mi vida y a mis abuelos, quienes ya no están, pero siempre me acompañan.

A la Universidad Autónoma de Baja California, en especial a la Facultad de Ciencias Químicas e Ingeniería por ser mi segundo hogar desde licenciatura y haberme permitido formarme en ella, gracias a todos los profesores que fueron partícipes de este proceso. Al Consejo Nacional de Ciencia y Tecnología por el apoyo que me brindó para continuar con mi preparación académica.

A los miembros del Comité, director de tesis Dr. J. Reyes Juárez Ramírez, Co-Director Dr. Luis Guillermo Martínez Méndez, sinodales Dra. Samantha Paulina Jiménez Calleros y Dr. Guillermo Licea Sandoval, por su disposición, apoyo y valiosas aportaciones. En especial al Dr. Reyes por su confianza depositada en mí, su tiempo y paciencia y a la Dra. Sam, de quien aprendí mucho, por su apoyo y valioso tiempo dedicados a mi tesis, sus consejos y observaciones siempre fueron de gran ayuda.

A mis compañeros de posgrado, por los gratos momentos y experiencias compartidas, es especial a mis amigas Claudia Tona y Fernanda Murillo, por brindarme su amistad, por el constante apoyo y optimismo. Un agradecimiento especial a Claudia, quien ha compartido conmigo un largo camino profesional y me ha brindado su amistad y apoyo desde licenciatura.

A mis amigas Alexia, Saraí, Jaqueline, Liliana, Aimeé y Jazmín, que siempre han estado para escucharme, apoyarme y motivarme. A mi amiga Lupita Magdaleno, por apoyarme, aconsejarme y siempre impulsarme a buscar nuevos retos.

A los profesionistas, expertos y estudiantes de Ingeniería de software 2019-1 y 2019-2 que participaron en la realización de encuestas y experimentos, gracias por su apoyo y compromiso.

A todos aquellos que no mencione, pero estuvieron presentes durante el proceso de esta tesis.

A todas las personas que creyeron en mi ♥

Resumen

Actualmente, en Scrum, existen diferentes métodos para estimar historias de usuarios en términos de esfuerzo o complejidad. La mayoría de las técnicas existentes no consideran atributos desglosados; Estas técnicas no siempre son precisas. Aunque Planning Poker es el método más utilizado en Scrum para estimar historias de usuarios, es principalmente efectivo en equipos experimentados, ya que la estimación depende principalmente de la observación de expertos, pero es difícil cuando es utilizado por equipos sin o con poca experiencia. En esta investigación, presentamos una propuesta para la descomposición de la complejidad desglosándola, a fin de considerar atributos importantes para la estimación. Utilizamos una red bayesiana para representar esos atributos y sus relaciones. Los bordes de la red están ponderados por el juicio de expertos sobre la importancia de los atributos. Los nodos de la red representan los atributos. Durante la fase de estimación de la historia de usuario, los miembros del equipo Scrum introducen los valores para cada atributo; de esta manera, la red genera un valor para la complejidad de una historia de usuario, que se transforma en un número de tarjeta de Planning Poker, que representa los puntos de la historia. El propósito de esta investigación es proporcionar a los equipos de desarrollo sin experiencia o sin datos históricos, un método para estimar la complejidad de las historias de los usuarios a través de un modelo centrado en los aspectos humanos de los desarrolladores.

Abstract

Currently, in Scrum, there are different methods to estimate user stories in terms of effort or complexity. Most of the existing techniques consider attributes in a coarse grain level; these techniques are not always accurate. Although Planning Poker is the most used method in Scrum to estimate user stories, it is primarily effective in experienced teams since the estimation mostly depends on the observation of experts, but it is difficult when is used by inexperienced teams. In this paper, we present a proposal for complexity decomposition in a coarse grain level, in order to consider important attributes for complexity estimation. We use a Bayesian Network to represent those attributes and their relations. The edges of the network are weighted with the judge of experts about the importance of the attributes. The nodes of the network represent the attributes. During the user estimation phase, the Scrum team members introduce the values for each attribute; in this way, the network generates a value for the complexity of a user story, which is transformed in a Planning Poker card number, which represents the story points. The purpose of this research is to provide to development teams without experience or without historical data, a method to estimate the complexity of user stories through a model focused on the human aspects of developers.

Índice general

Índice de figuras.....	8
Índice de tablas	9
Lista de acrónimos	10
1. Introducción.....	11
1.1 Planteamiento del problema	13
1.2 Justificación.....	14
1.3 Objetivos	15
1.3.1 Objetivo general.....	15
1.3.2 Objetivos específicos	15
1.4 Preguntas de investigación	16
1.5 Beneficio esperado.....	16
1.6 Metodología.....	17
1.7 Organización del documento de tesis.....	18
2. Estado del arte.....	20
2.1 Conceptos básicos	20
2.1.1 Scrum	20
2.1.2 Planning Poker.....	26
2.1.3 Complejidad	29
2.1.4 Redes Bayesianas.....	30
2.2 Trabajos relacionados.....	35
3. Modelo propuesto	39
3.1 Revisión Sistemática de la Literatura.....	39
3.1.1 Formulación de preguntas de investigación	39
3.1.2 Definición de la estrategia de búsqueda.....	39
3.1.3 Definición de criterios de inclusión y exclusión	39
3.1.4 Evaluación de la calidad.....	40
3.1.5 Resultados	40
3.2 Atributos seleccionados.....	41

3.3	Técnicas para representar el modelo.....	43
3.4	Construcción de la red Bayesiana.....	44
3.4.1	Parte cualitativa.....	44
3.4.2	Parte cuantitativa.....	45
4.	Pruebas del modelo.....	53
4.1	Experimentos.....	53
4.1.1	Estudiantes.....	53
4.1.2	Profesionales.....	54
4.2	Resultados.....	54
4.3	Discusión.....	58
5.	Conclusiones, limitaciones, aportaciones y trabajo futuro.....	61
5.1	Conclusiones.....	61
5.2	Limitaciones.....	63
5.3	Aportaciones.....	63
5.4	Trabajo futuro.....	64
	Bibliografía.....	65
	Apéndices.....	73
	Apéndice A Encuestas.....	73
	A.1 Encuesta para validar atributos.....	73
	A.2 Encuesta para estimar a través de la red.....	76
	Apéndice B Historias de usuario.....	79
	B.1 Historias de usuario.....	79
	Apéndice C Publicaciones.....	80
	C.1 Publicaciones en JCR.....	80
	C.2 Publicaciones en conferencia.....	80

Índice de figuras

Figura 1. Funcionamiento del modelo propuesto.	17
Figura 2. Proceso de Scrum.....	21
Figura 3. Roles en Scrum	22
Figura 4. Formato detallado de una Historia de Usuario	25
Figura 5. Representación del valor de un punto de historia	26
Figura 6. Cartas utilizadas en Planning Poker	27
Figura 7. Proceso de estimación de Historias de Usuario	29
Figura 8. Relación entre nodos	30
Figura 9. Modelo propuesto.....	43
Figura 10. Red Bayesiana resultante.....	52

Índice de tablas

Tabla 1. Trabajos relacionados	36
Tabla 2. Atributos seleccionados para el primer nivel	42
Tabla 3. Grado de aceptación para cada atributo.....	46
Tabla 4. Pesos y frecuencias de los atributos	48
Tabla 5. Combinaciones de escalas y pesos para el nodo tamaño.....	51
Tabla 6. Tabla de probabilidades condicionales para el nodo tamaño.....	52
Tabla 7. Equivalencia entra cartas y puntos de historia	55
Tabla 8. Resultado de estudiantes	56
Tabla 9. Resultados de profesionales	57

Lista de acrónimos

Acrónimo	Significado
BP	Backlog del producto
ES	Equipo de Scrum
HU	Historia de Usuario
PH	Puntos de Historia
PO	Product Owner
PP	Planning Poker
PS	Planeación del Sprint
RB	Red Bayesiana
RSL	Revisión Sistemática de la Literatura
SM	Scrum Master
TPC	Tabla de Probabilidad Condicional

Capítulo 1

1. Introducción

En este capítulo se exponen los antecedentes del problema que se resuelve en este proyecto, se plantean los objetivos y se describen los beneficios de este proyecto.

Las metodologías ágiles cada vez más están siendo adoptadas por cualquier tipo de empresas debido a que permiten tener procesos organizados y mejorables. El uso de metodologías ágiles en el desarrollo de software está creciendo debido a la productividad asociada a estas metodologías, además de las diferentes ventajas que ofrece (Popli and Chauhan 2014). No obstante, las metodologías ágiles cuentan con debilidades en el proceso de estimación. Llevar una buena administración de un proyecto de software es fundamental para obtener un proyecto exitoso, es decir que termine dentro del tiempo, costo y calidad planeada (Popli and Chauhan 2013). Una de las metodologías ágiles más utilizadas en la actualidad es Scrum (Gutiérrez 2010).

Scrum es un método de desarrollo ágil, es un marco de trabajo que presenta un conjunto de prácticas para mantener la visibilidad, inspección y adaptación de proyectos de desarrollo de software (Pauly, Michalik, and Basten 2015; RUBIN 2013; Schwaber and Sutherland 2016; SCRUMstudy 2016; Srivastava, ASET, Amity University, Uttar Pradesh, Noida Campus, and Saraswat 2017).

Los principales componentes de Scrum son: los roles: Scrum Master (SM), Product Owner (PO) y el equipo de Scrum (ES); el Sprint, el cual es un periodo de tiempo en el cual se crea un incremento del producto; Backlog del Producto (BP), en el cual se priorizan los requisitos del usuario final; Historia de Usuario (HU), la cual representa un requisito del usuario; los eventos de Scrum: reunión de Planeación del Sprint (PS), reunión diaria de Scrum, reunión de revisión del Sprint y reunión de retrospectiva del Sprint (RUBIN 2013; Schwaber and Sutherland 2016; Srivastava, ASET, Amity University, Uttar Pradesh, Noida Campus, and Saraswat 2017).

Una primera versión de la HU es una descripción breve, simple y específica de un requerimiento (Lucassen et al. 2016). Una versión detallada de la HU define características y requisitos que proporcionan valor para el usuario o cliente (Schwaber and Sutherland 2013).

Además, estos requisitos proporcionan información para que el desarrollador haga una estimación razonable (López-Martínez et al. 2017).

Un proyecto de software implica la estimación del costo, esfuerzo, tamaño y duración (Lopez-Martinez et al. 2018). Los equipos de software tradicionales ofrecen estimaciones en un formato de tiempo: días, semanas, meses. Sin embargo, muchos equipos ágiles han pasado a los Puntos de Historia (PH), que se centran en la complejidad. El uso de los PH en Scrum se ha vuelto popular debido a que es un valor relativo que da una idea del tamaño o dificultad que implica una tarea con respecto a otra, es por este motivo que el valor que representan solo será relevante para un equipo (Lopez-Martinez 2017; Mahnič and Hovelja 2012).

El método más utilizado en Scrum para estimar HU es Planning Poker (PP). Este método consiste en asignar PH a cada HU para determinar su complejidad (Lopez-Martinez et al. 2018); todos los integrantes del ES participan en este proceso asignando un valor en PH a cada HU. Cada integrante del ES tiene una opinión diferente en el proceso de estimación y un mayor compromiso con el equipo (Lucassen et al. 2016). Se deben considerar muchos atributos al estimar los PH (Alostad, Abdullah, and Aali 2017; Dragicevic, Celar, and Turic 2017; López-Martínez et al. 2017; Lopez-Martinez 2017): Complejidad de la historia, riesgos, inexperiencia del equipo con el desarrollo de la historia, problemas de implementación, requisitos de implementación, interdependencias, entre otras.

Una gran controversia es el hecho de cómo hacer la estimación. La mejor estimación depende de la experiencia y el conocimiento previo de los desarrolladores, ya que estos factores son útiles para deducir y considerar todos los atributos y escenarios posibles durante el proceso de estimación. Por esa razón, el proceso de estimación puede ser problemático para los desarrolladores novatos, ya que no tienen suficiente experiencia (Scott and Pfahl 2018). La incertidumbre en los requisitos también tiene un impacto negativo en la estimación (Lenarduzzi 2015).

1.1 Planteamiento del problema

En Scrum, para estimar una HU se pueden utilizar varios métodos existentes, como PP (López-Martínez et al. 2017), Wideband Delphi (Mahnič and Hovelja 2012), Puño de cinco (SCRUMstudy 2016), Estimación por afinidad (SCRUMstudy 2016) y tamaños de camisetas (Peischl et al. 2010; Ziauddin, Shahid Kamal Tipu 2012), sin embargo, PP es el método más utilizado en Scrum.

PP es una técnica basada en el juicio de expertos (Mahnič and Hovelja 2012), la estimación se realiza para cada HU en PH que representan la complejidad para su implementación en el proyecto (Mahnič and Hovelja 2012). PP tiene muchos beneficios, sin embargo, este método tiene algunas desventajas porque los resultados siempre se basan en la experiencia del desarrollador (Ramirez-Noriega et al. 2016), además la decisión de cada miembro del equipo no siempre está clara porque consideran la complejidad en términos generales, cuando en realidad la complejidad está relacionada con otros atributos. El principal problema en la utilización de PP es la alta subjetividad al momento de estimar historias de usuario. Es por eso que utilizar PP resulta algunas veces complicado y poco confiable, sobre todo si se está trabajando con un equipo inexperto.

En la actualidad los métodos de estimación no siempre son precisos y la mayoría de ellos dependen de atributos como nivel de experiencia de los miembros del equipo, historial de proyectos previos, conocimiento del tema del proyecto a desarrollar, entre otros (Mahnič and Hovelja 2012; Mendes 2011; Zahraoui and Janati Idrissi 2015). Incluso muchas veces los desarrolladores no tienen conocimiento de que aspectos considerar al momento de realizar una estimación, desconocen información del proyecto, relacionada a las historias de usuario y a sus habilidades.

Los métodos de estimación existentes toman en cuenta aspectos del proyecto, técnicos y aspectos de la empresa, pero muy pocos consideran aspectos personales importantes para la estimación (Mahnič and Hovelja 2012; Raza et al. 2017). En (Dragicevic, Celar, and Turic 2017) se consideran 6 atributos para estimar esfuerzo, 5 atributos del tipo técnicos y 1 del tipo personal. En (Alostad, Abdullah, and Aali 2017) se utilizan 4 atributos considerando 3 del tipo técnico y 1 del tipo personal. En (López-Martínez et al. 2017) se utiliza un modelo de 5 atributos y propone

solo 1 atributo del tipo personal; en cambio (Zahraoui and Janati Idrissi 2015) en su modelo de 3 atributos no contemplan ningún atributo del tipo personal.

Aquellos métodos que consideran la experiencia no la desglosan o analizan (Alostad, Abdullah, and Aali 2017; Dragicevic, Celar, and Turic 2017; López-Martínez et al. 2017; Zahraoui and Janati Idrissi 2015), aun cuando Scrum depende en gran parte de los aspectos personales, pues estos siempre tienen una importante influencia durante la fase de estimación (Mahnič and Hovelja 2012).

Los enfoques tradicionales y las metodologías ágiles han considerado muchos atributos. Particularmente en el contexto ágil, no hay consenso sobre un conjunto de atributos, debido a la subjetividad en la implementación y no hay conjuntos de datos con datos históricos; La razón es la corta edad de estas metodologías.

El objetivo de este trabajo es facilitar el proceso de estimación sobre todo en desarrolladores con poca o nula experiencia, o sin datos históricos.

1.2 Justificación

- Es importante contar con un método preciso de estimación de historias de usuario, el cual brinde confianza tanto al equipo de trabajo como al cliente. El modelo a desarrollar debe facilitar la toma de decisiones, la asignación y estimación de trabajo, entre otros atributos; permitiendo al equipo entregar productos al cliente en tiempo y forma. Un método con estas características resultaría en una estimación más aproximada a la realidad.
- Esta investigación propone mejorar la precisión en la estimación en comparación con otros métodos existentes, para eso nos enfocaremos en los aspectos personales de los desarrolladores lo cual nos ayudaría a reducir la inseguridad de los desarrolladores al estimar.
- Una técnica de estimación de historias de usuario, el cual no considere la recopilación histórica de proyectos, sería de gran ayuda sobre todo en empresas de desarrollo de software que no cuentan con datos históricos y personal con un nivel elevado de experiencia en estimación historias de usuario.

- El método propuesto podrá ser aplicado en diferentes sectores: industria y educación. En la industria permitirá a los equipos estimar de manera precisa, dejando de lado las estimaciones consensuales. El beneficio para las empresas se verá reflejado en tiempo y costos, pues se reducirá el tiempo de estimación y los equipos de trabajo serán más productivos. Debido a que la estimación de historias de usuario depende de la experiencia y subjetividad del equipo, la aplicación de este método en la educación permitirá a los alumnos estimar con precisión y confianza aun cuando estos no tengan experiencia. Los resultados obtenidos servirán para evaluar la eficiencia del método, de los datos utilizados y para realizar futuro trabajo.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un modelo para la estimación de historias de usuario que considere los aspectos personales de los desarrolladores a través del desglose del atributo experiencia para proporcionar a los equipos de desarrollo sin experiencia o sin datos históricos un método que favorezca positivamente la precisión.

1.3.2 Objetivos específicos

- Extraer de la literatura atributos considerados importantes para la estimación de historias de usuario.
- Diseñar un modelo con atributos obtenidos en la revisión de la literatura basado en aspectos personales para estimar historias de usuario.
- Validar los atributos seleccionados con la opinión de los expertos a través de una encuesta.
- Definir una técnica para representar el modelo tomando en cuenta el juicio de expertos.
- Implementar el modelo propuesto en un ambiente académico y profesional, para comparar como difieren y se comportan ambos ambientes y conocer como esto impacta la estimación.
- Validar el modelo propuesto comparándolo con modelos existentes.

1.4 Preguntas de investigación

En esta sección se describe una serie de preguntas que deberán contestarse para concluir la investigación. A continuación, se describen cada una de ellas:

1. ¿Cuáles son los atributos que se consideran más importantes para estimar la complejidad de las historias de usuario? Es indispensable conocer los atributos que se consideran más importantes en el proceso de estimación, en vista de que se desconoce qué es lo que pasa por la mente de los desarrolladores que están estimando. Esto complica el proceso de estimación de los desarrolladores con poca o sin experiencia aumentando la incertidumbre en sus estimaciones.
2. ¿Cómo se diseñará el modelo propuesto? Para crear nuestro modelo primero se tienen que identificar los atributos, seguido de eso necesitamos seleccionar una técnica para representar nuestro modelo, requerimos una técnica que se adapte a nuestro modelo y que nos permita utilizar datos cualitativos. También es necesario tomar en cuenta la opinión de los profesionales para la construcción de nuestro modelo, ya que ellos cuentan con experiencia y buscamos que nuestro modelo estime como lo haría un profesional.
3. ¿Cómo será validado el modelo propuesto? Debemos diseñar un experimento que nos permita validar nuestro modelo en un ambiente real, tanto con estudiantes como con profesionales.

1.5 Beneficio esperado

El modelo propuesto en este trabajo nos da un modelo dividido en atributos, en su mayoría relacionados al aspecto humano, poniendo más atención al atributo experiencia sin dejar de lado aspectos técnicos importantes con el fin de mejorar la precisión en la estimación. Además, los estudiantes, desarrolladores sin experiencia o empresas de nueva creación pueden hacer uso de esta técnica debido a que no requieren datos históricos de proyectos previos. Como se muestra en la Figura 1, este modelo disminuirá la incertidumbre y facilitará la toma de decisiones de los equipos ya que los desarrolladores no siempre conocen que aspectos considerar al momento de realizar una estimación.



Figura 1. Funcionamiento del modelo propuesto.

1.6 Metodología

Etapa 1 Revisión de la literatura

- Se realizó una revisión sistemática de la literatura basada en la metodología B. Kitchenham (Kitchenham, Budgen, and Brereton 2018) para determinar los atributos más importantes para la estimación de historias de usuario.

Etapa 2: Diseño del modelo

- Se identificaron y analizaron los atributos que impactan más en la estimación de historias de usuario basados en la revisión de literatura.
- Se diseñó un modelo con los atributos seleccionados de la revisión de la literatura. El modelo se creó enfocado en los aspectos personales de los desarrolladores. Como resultado obtuvimos un modelo jerárquico formado por nodos padres e hijos.
- Se seleccionaron a las redes Bayesianas como la técnica para representar nuestro modelo.
- Se determinó la importancia y peso de cada atributo dentro del modelo, para obtener estas ponderaciones se solicitó la opinión de expertos a través de una encuesta digital, descriptiva con opción múltiple y haciendo uso de la escala Likert.
- Se determinaron unidades de medida y escala de valores para cada atributo de primer nivel.
- Se obtuvieron tablas de probabilidad condicional para cada atributo de segundo nivel.
- Se utilizó el software GeNIe 2.4 academic Para la formación de la red Bayesiana.

Etapa 3 Pruebas al modelo

- El modelo se implementó en dos escenarios:
 - Escenario 1. Ambiente académico: se utilizó una muestra de alumnos estudiantes de Ingeniería en Computación con conocimientos básicos en Scrum.
 - Escenario 2. Ambiente profesional: se utilizó una muestra de profesionales en la industria del software con conocimientos avanzados en Scrum.
- Los datos recolectados se almacenaron en una hoja de cálculo, separados por escenario, equipo, HU, etc.

Etapa 4 Análisis y validación de resultados

- Para analizar y validar el modelo propuesto se comparó con modelos existentes, se utilizaron pruebas de correlación de Spearman a las estimaciones de cada modelo.

1.7 Organización del documento de tesis

Esta sección describe de manera breve la organización de este documento.

El capítulo 2 expone el estado del arte, este se centra en las características principales de Scrum. También se describen las características de las redes Bayesianas, y porque se consideró que es apropiada para representar el modelo propuesto en este trabajo. Al final de este capítulo se presentan los trabajos relacionados.

El capítulo 3 muestra cómo se seleccionó el modelo propuesto considerando los atributos seleccionados. Asimismo, en este capítulo se explica cómo se construyó la parte cuantitativa y cualitativa de nuestra red Bayesiana.

El capítulo 4 presenta los experimentos aplicados a estudiantes y profesionales para la validación del modelo propuesto.

El capítulo 5 describe las conclusiones, limitaciones, aportaciones y trabajo futuro de nuestra investigación.

Las últimas secciones se describen las referencias que respaldan esta investigación y los anexos.

Capítulo 2

2. Estado del arte

En este capítulo se definen los conceptos básicos y trabajos relacionados que rodean esta investigación.

2.1 Conceptos básicos

2.1.1 Scrum

2.1.1.1 Fundamentos

Scrum es la metodología ágil más popular en la industria del software. Mediante el uso de las prácticas Scrum, varias compañías han mejorado su calidad y productividad. Scrum es un método ágil de desarrollo definido como un proceso iterativo, incremental y empírico para gestionar y controlar el desarrollo de un proyecto (Georgsson 2011; Usman, Börstler, and Petersen 2017). El objetivo de Scrum es mantener la visibilidad, inspección y adaptación de los proyectos (Edici n.d.).

Características de Scrum (Edici n.d.; Schwaber and Sutherland 2013):

- Un marco de Gestión de proyectos ágiles.
- Enfoque iterativo e incremental.
- Puede ser utilizado para cualquier tipo de desarrollo de software.
- Trabaja colaborativamente en equipos.
- Se realizan entregas parciales y regulares del producto final.
- Se entrega software funcional en periodos cortos.

Scrum parte de la planificación basada en objetivos comerciales que priorizan aquellos que aportan más valor y esperan detallar objetivos y tareas a medida que se acerca el momento de la construcción de estos objetivos (Popli and Chauhan 2013). La estimación para proyectos Scrum utiliza un enfoque de descomposición que cubre los siguientes pasos (Pressman n.d.):

- Cada HU se considera por separado para fines de estimación.
- La estimación de la historia del usuario puede basarse en datos históricos, un modelo empírico o experiencia.
- La historia del usuario se divide en el conjunto de tareas de ingeniería de software que deberán desarrollarse.

Scrum, tiene tres diferentes niveles de planificación: Planeación de liberación, Planeación del Sprint y la reunión diaria. Durante la reunión de planeación de liberación, se discuten los aspectos estratégicos básicos como los costos generales o funcionalidad de un proyecto de desarrollo. Los detalles operacionales en cambio se planifican de un Sprint a otro durante el desarrollo del proyecto (Overhage and Schlauderer, 2012). Un Sprint tiene una duración de 2 a 4 semanas y es el corazón de Scrum. Cada Sprint comienza con la reunión de Planeación del Sprint. El Product Owner y el Equipo de Scrum trabajan juntos para saber qué harán en el siguiente sprint (Lopez-Martinez et al. 2018).

El nivel más detallado de la planificación se lleva a cabo en las reuniones diarias las cuales tienen una duración de 5 minutos y es donde los miembros del equipo informan de su progreso en el desarrollo actual y las nuevas tareas son asignadas, el proceso de Scrum se muestra en la Figura 2.

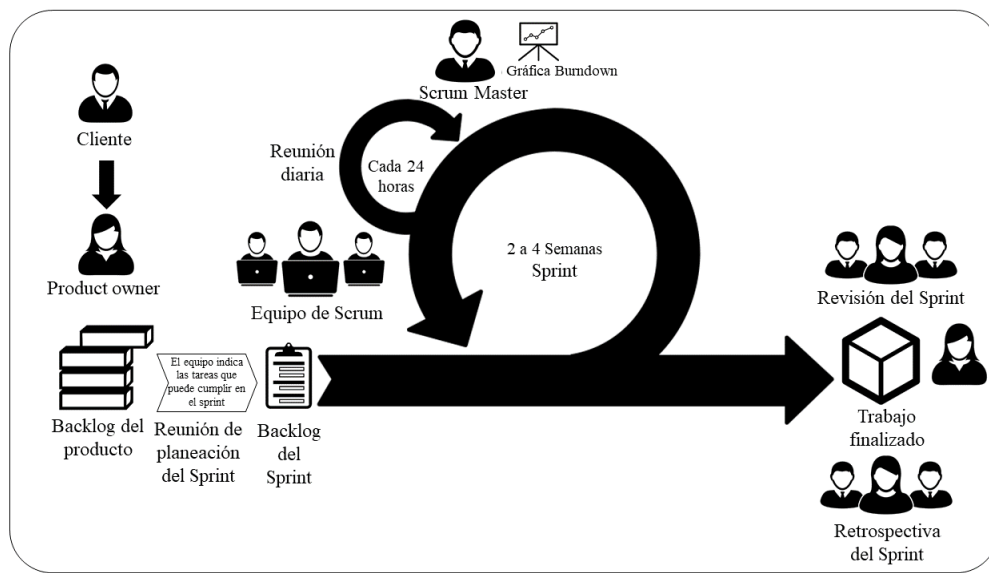


Figura 2. Proceso de Scrum

En comparación con las metodologías tradicionales, Scrum se basa en una estructura de equipo de auto-organizado, esto no sólo hace que los proyectos Scrum sean más transparentes y flexibles, también presume que los miembros del equipo tienen un alto compromiso y sentido de responsabilidad (Martinez 2017). Además de la colaboración en equipo, la colaboración con el cliente también juega un papel importante en los proyectos Scrum. Como propietario del producto, los clientes tienen su propio papel en el proyecto y se integran en varias etapas del proyecto. Por ejemplo, deben participar en las revisiones del Sprint para que estén siempre al tanto del estado actual del proyecto (Overhage and Schlauderer, 2012).

2.1.1.2 Roles

Scrum tiene tres roles principales: PO, SM y ES. El PO representa la voz del cliente; la principal responsabilidad de este rol es la creación del Backlog del producto BP, que es una lista de requisitos ordenados según la prioridad. El SM es responsable del proceso de Scrum, se lo enseña a cada uno de los involucrados en el proyecto y, si es necesario, ayuda al equipo con los problemas que surgen durante el proyecto (Lopez-Martinez et al. 2018).

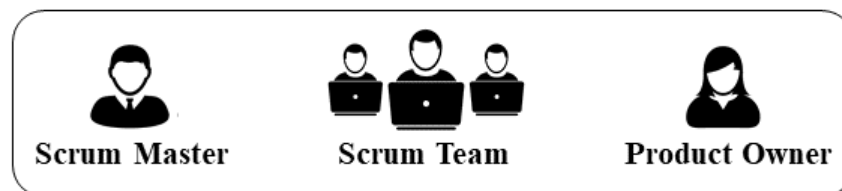


Figura 3. Roles en Scrum

Scrum Master

El SM es responsable de supervisar que el proceso de Scrum funcione adecuadamente, de enseñarlo a cada uno de los involucrados en el proyecto, si es necesario. Es considerado como un facilitador para el equipo de desarrollo de software, debe asegurarse de que cada uno sigue las reglas y prácticas. El SM hace que Scrum sea parte de la cultura de la organización mostrando sus beneficios. Además, el SM trabaja muy de cerca con el PO (Bass 2015). Las principales funciones de un SM son (Bass 2015; Edici n.d.):

- Eliminar impedimentos del equipo
- Asegurar la comunicación eficiente entre el ES y PO
- facilitar los diferentes eventos de Scrum.
- Guiar las reuniones diarias.
- Observar problemas o conflictos personales que deben resolverse.

Scrum Team

El ES son los encargados de conocer y realizar el trabajo necesario para desarrollar las historias de usuario. Además, son responsables del éxito de cada iteración y del proyecto en su conjunto. El ES selecciona las actividades que se realizan en cada Sprint; cada miembro del equipo selecciona las tareas que se compromete a realizar durante el Sprint (Jeff Sutherland, Neil Harrison 2014). Las principales características del ES son (Edici n.d.; Raza et al. 2017):

- Incluye una combinación de ingenieros de software, arquitectos de software, programadores, analistas, ingenieros de pruebas y diseñadores.
- Estimar historias de usuario.
- Asegura que al final del Sprint la funcionalidad potencialmente sea entregada.

El ES se caracteriza por ser un equipo autoorganizado y multifuncional; Este rol tiene la responsabilidad del desarrollo del producto y la estimación de las historias de usuarios en las que cada integrante del equipo se compromete a desempeñarse durante cada Sprint (Lopez-Martinez et al. 2018).

Product Owner

Representa a quien tiene un interés en el proyecto y en el producto resultante, es el único responsable de definir los requerimientos del producto a desarrollar durante el proyecto y conseguir el financiamiento inicial y continuo para el proyecto. El PO también es responsable de planear cómo se va a liberar el producto, debe asegurarse que la funcionalidad más valiosa se

produce primero, priorizando los requerimientos de acuerdo al mercado o al área de negocio (Bass 2015). Las funciones principales del PO son (Edici n.d.):

- Redacta las historias de usuario
- Administra el BP.
- Es la representación del cliente.
- Asegurarse que el equipo entienda los requerimientos.
- Validar y aceptar el resultado del Sprint.

2.1.1.3 Historias de Usuario

Las historias de usuario son una notación concisa para expresar requisitos que se emplean cada vez más en la ingeniería de requisitos ágil y en el desarrollo ágil. De hecho, se han convertido en la notación de requisitos más utilizada en proyectos ágiles y su adopción ha sido fomentada por su uso en numerosos libros sobre desarrollo ágil. La literatura sugiere los mismos tres componentes básicos de una HU (Moreno and Yagüe 2012; Pandit and Tahiliani 2015):

- Breve texto que describe y representa la HU.
- Conversaciones entre las partes interesadas para intercambiar perspectivas sobre la HU.
- Criterios de aceptación. Estas son características de calidad, funcionalidades, escenarios, reglas de negocio, interfaces externas, restricciones y definiciones de datos.

Como se puede observar, en la Figura 4 se muestra un ejemplo de toda la información que puede contener una HU. El texto que representa la HU captura solo los elementos esenciales de un requisito: para quién es, qué se espera del sistema y, opcionalmente, por qué es importante (Lucassen et al. 2016). Las historias de usuario involucran 2 etapas:

1. Redacción: en esta participa el PO, describiendo a detalle los requerimientos del cliente en forma de historias de usuario.
2. Estimación: en esta participa el ES, a través de algún método se asigna una puntuación a cada HU.

ID:	Sprint:	Prioridad:	Puntos de Historia:
Título de la historia del usuario:	Un título corto		
Texto de la historia del usuario:	Como un <Usuario> [QUIEN], quiero <Tener> [QUÉ] para que <Beneficio> [POR QUÉ]		
Criterios de aceptación:	<Dado que> <Cuando> <Entonces>		
Criterios de prueba	Un control de alto nivel de los criterios de aceptación.		
Flujo principal	La secuencia de pasos para lograr el objetivo de la historia del usuario.		
Flujo alternativo:	Una secuencia diferente de pasos para lograr el objetivo de la historia del usuario.		
Flujo de excepción:	Una secuencia de eventos que no permiten alcanzar el objetivo de la historia del usuario.		

Figura 4. Formato detallado de una Historia de Usuario

Las HU ofrecen una serie de ventajas sobre la alternativa, algunas de las razones son:

- Enfatizan la comunicación verbal en lugar de la escrita.
- Son comprensibles tanto para usted como para los desarrolladores.
- Son del tamaño adecuado para la planificación.
- Funcionan para el desarrollo iterativo.

Una HU se puede estimar en PH, que representan la complejidad que tendrá en su implementación (Jennings and Wooldridge 2000; Mahnič and Hovelja 2012).

2.1.1.4 Puntos de historia.

Son una unidad de medida para expresar una estimación de la complejidad global que se necesita para implementar una HU (Dragicevic, Celar, and Turic 2017; Scott and Pfahl 2018). Los PH son un valor relativo y no pueden relacionarse con el tiempo (Ramirez-Noriega et al. 2016). De la misma manera, no se puede comparar la complejidad técnica y la incertidumbre de las diferentes historias de usuario, así como para medir la velocidad del proyecto (Dragicevic, Celar,

and Turic 2017). Una buena característica de los puntos de la historia es que cada equipo los define como mejor les parezca (Cohn 2004).

Como primer paso al utilizar esta unidad de medida, el equipo generalmente acuerda la cantidad de PH que se asignarán una HU de línea de base. A partir de ese momento, la estimación se basa en la comparación con esa línea de base, es decir se hace por analogía (Porru et al. 2020).

Los PH no se pueden comparar a horas de esfuerzo debido a que no se puede establecer relación directa entre complejidad y tiempo (Martinez 2017; Popli and Chauhan 2013). El número de horas que requiere para realizar una tarea depende de la capacidad técnica de la persona, la carga de trabajo del equipo, etc. Cómo se observa en la Figura 5 el proceso de asignación de PH depende de varios factores. Sin embargo, el número de factores y la priorización de los mismos es diferente para cada integrante del equipo de acuerdo a su experiencia y consideraciones. Es por esta razón, que el valor de un punto de historia difiere dependiendo de la situación en la que se realice la estimación. En Scrum hay diferentes técnicas para estimar, sin embargo, PP es la más común (López-Martínez et al. 2017).

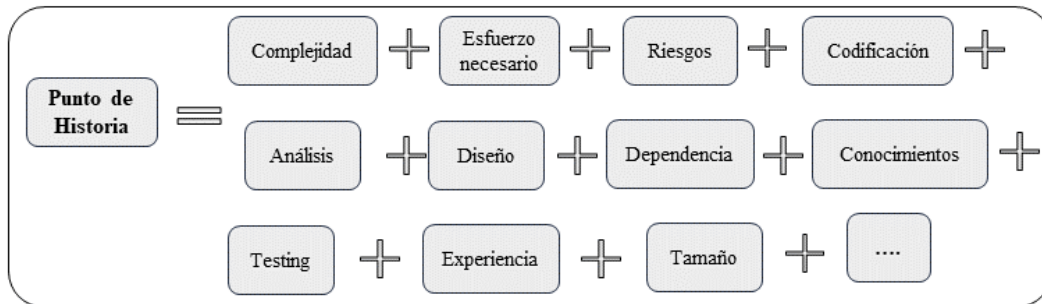


Figura 5. Representación del valor de un punto de historia

2.1.2 Planning Poker

PP es la técnica más utilizada en Scrum para estimar historias de usuario. PP utiliza cartas con números. Estas cartas se basan en la secuencia Fibonacci modificada (1, 2, 3, 5, 8, 13, 21, 40, 100) (Lopez-Martinez et al. 2018). Los PH están destinados a representar el consenso dentro de un equipo, de hecho, diferentes equipos pueden proporcionar estimaciones diferentes para la misma HU (Porru et al. 2020).

Como se muestra en la Figura 6, adicional a la secuencia Fibonacci, PP en ocasiones utiliza otras opciones. Se utiliza el 0 para elementos que no requieran esfuerzo y 1/2 para elementos demasiados pequeños. El infinito (∞) se utiliza cuando una estimación mayor debe de dividirse en partes menores. El signo de interrogación (?) significa que es imposible estimar por falta de información o detalle, y la taza de café significa que es hora de hacer un descanso. Sin embargo, estas opciones varían de equipo a equipo.

Lo que se busca al usar PP es asignar PH a las HU utilizando las cartas para determinar su complejidad (Mahnič and Hovelja 2012).

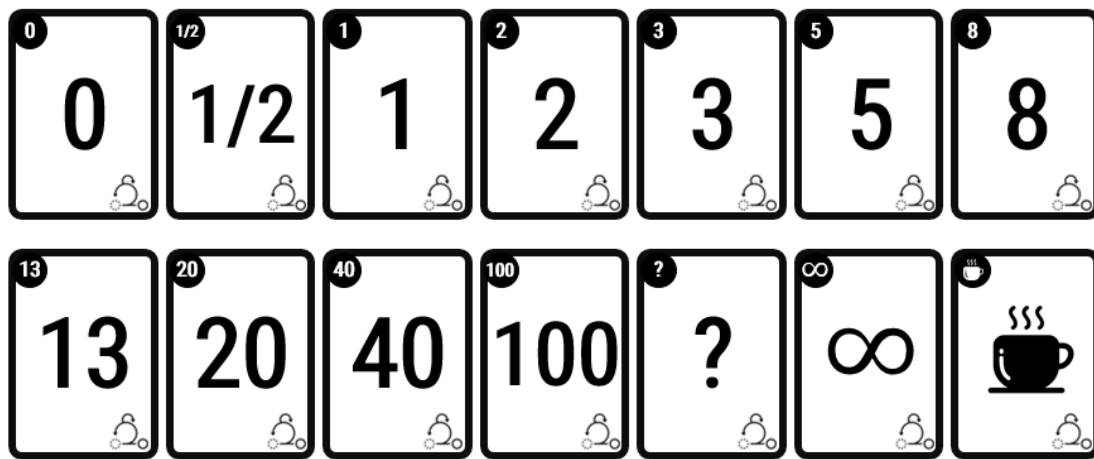


Figura 6. Cartas utilizadas en Planning Poker

El PO y el ES se reúnen para discutir los requerimientos del BP con el fin de alcanzar estimaciones basadas en el consenso (Lopez-Martinez et al. 2018). Dicha discusión debe centrarse en los diferentes elementos de la funcionalidad involucrada y sus aspectos de implementación para cada HU, teniendo en cuenta la breve descripción, los criterios de aceptación y los casos de prueba.

2.1.2.1 Procedimiento

Aunque en el proceso de estimación también participa el PO y el SM, las historias de usuario son estimadas solo por el ES (Edici n.d.; Jørgensen 2012; Mahnič and Hovelja 2012):

1. El PO inicia explicando las historias de usuario.

2. Posteriormente el ES discute el trabajo que se requerirá para realizarlas, hacen preguntas y cuestionamientos al PO acerca de las historias de usuario.
3. Cada miembro del equipo asigna de manera privada una puntuación (o carta) a cada HU contemplando los aspectos que el considere.
4. Las cartas seleccionadas por los miembros del ES son mostradas simultáneamente para asegurar la independencia de su estimación.
5. Si las estimaciones difieren mucho cada estimador debe explicar el por qué asignó ese valor, quien dio la estimación más alta y la más baja también deberán explicar el porqué de sus estimaciones.
6. El proceso se puede llegar a repetir hasta 3 veces si es necesario con el fin de llegar al consenso en la estimación.

PP provee buenas estimaciones, ya que asegura la participación de todo el equipo equitativamente. La participación de personas con diferente contextos ayuda a reducir el sobreoptimismo de las estimaciones basadas en juicio de experto identificando problemas que afectan su implementación (López-Martínez et al. 2017; Mahnič and Hovelja 2012).

2.1.2.2 Problemas en la estimación utilizando Planning Poker

PP tiene muchos beneficios, sin embargo, los expertos se han percatado que este método no siempre es eficiente, esto debido a que (Mahnič and Hovelja 2012; Raith et al. 2013):

- El resultado es siempre basado en la observación de expertos y su experiencia.
- El puntaje asignado a las historias de usuario es un valor que no puede ser convertido fácilmente a tiempo.
- La decisión de los miembros del equipo es poco clara, debido a que toman en cuenta la complejidad de las historias de usuario forma general, sin desglosar sub-atributos.

Para la estimación de historias de usuario existen diferentes atributos que deben considerarse como: la experiencia en el lenguaje de programación (Rath, Acharya, and Satapathy 2016; Zare, Khademi Zare, and Fallahnezhad 2016), experiencia en proyectos anteriores (Hsu et al. 2016; Malgonde and Chari 2018), experiencia con la tecnología (Mahnič and Hovelja 2012;

Zare, Khademi Zare, and Fallahnezhad 2016), familiaridad con el proyecto (Sholiq, Dewi, and Subriadi 2017), tamaño del equipo (Drahý and Pastor 2016)(Abdou, Yong, and Othman 2016), duración del proyecto (Abdou, Yong, and Othman 2016; Hsu et al. 2016), entre otros. Como se muestra en la Figura 7, los desarrolladores sin o con poca experiencia pueden llegar a tener problemas al realizar estimaciones, ya que cada uno tiene una perspectiva diferente de que atributos considerar para estimar, además no tienen la experiencia en estimación, ni en la consideración de los sub-atributos que se involucran en la realización de las historias de usuario. Por otro lado, las empresas pequeñas que no tienen datos históricos de proyectos anteriores, no emplean un método estandarizado para la estimación (Ramirez-Noriega et al. 2016).

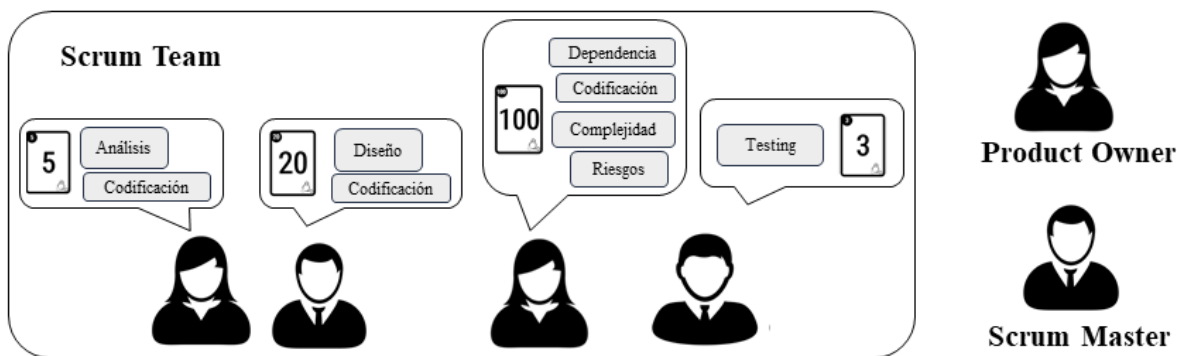


Figura 7. Proceso de estimación de Historias de Usuario

2.1.3 Complejidad

Es el número de elementos en un sistema y las posibles relaciones entre estos elementos (Cristóbal 2017). Si hablamos de complejidad en un contexto de software, podemos definirla como la dificultad del problema que se intentará resolver, la estructura del código o las relaciones entre los elementos que se manejarán en el software. En otras palabras, el término complejidad se puede utilizar para analizar la naturaleza compleja de los problemas, el código y los datos (López-Martínez et al. 2017). La complejidad del proyecto tiene un fuerte efecto sobre la incertidumbre inherente a la planificación. Sin embargo, la complejidad es una medida relativa que es afectada por la familiaridad o conocimiento de proyectos previos (Pressman n.d.).

En el caso de las historias de usuarios, la complejidad se centra en los diferentes elementos de la funcionalidad involucrada y sus aspectos de implementación. Determinar la complejidad al

estimar una HU no es un trabajo simple ya que involucra muchos aspectos y actualmente no existe un consenso sobre qué aspectos se deben considerar, es por eso que cada autor determina aspectos diferentes para estimar la complejidad (Dragicevic, Celar, and Turic 2017; López-Martínez et al. 2017). Hay estudios que trabajan con Redes Bayesianas para abordar los problemas que se presentan al estimar, ya que las Redes Bayesianas debido a su naturaleza estadística permiten manejar la incertidumbre (Dragicevic, Celar, and Turic 2017).

2.1.4 Redes Bayesianas

Una Red Bayesiana (RB) es un modelo probabilístico que relaciona un conjunto de variables aleatorias mediante un grafo dirigido, son redes graficas sin ciclos en el que se representan variables aleatorias y las relaciones de probabilidad que existan entre ellas que permiten conseguir soluciones a problemas de decisión en casos de incertidumbre (DOLDÁN TIÉ 2007).

Las RBs pertenecen a la familia de modelos gráficos probabilísticos. Estas estructuras gráficas se utilizan para representar el conocimiento sobre un dominio incierto. Cada nodo en el gráfico representa una variable aleatoria, mientras que los arcos entre nodos representan dependencias probabilísticas entre las variables aleatorias correspondientes. Como se puede observar en la Figura 8, el nodo hijo depende solo de sus nodos padres y cada nodo está asociado a una Tabla de Probabilidad Condicional (TPC) que definen la probabilidad de cada estado en el que una variable puede estar. Estas dependencias condicionales en el gráfico a menudo se estiman utilizando métodos estadísticos y computacionales conocidos. Por lo tanto, las redes bayesianas combinan principios de teoría de grafos, teoría de probabilidad, informática y estadística (DOLDÁN TIÉ 2007; Rodríguez and Dolado 2007).

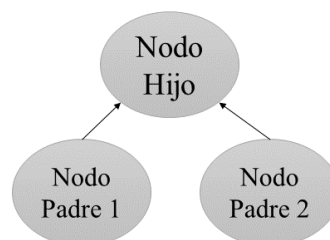


Figura 8. Relación entre nodos

Las redes Bayesianas juegan diversos papeles importantes dentro de la Inteligencia Artificial. Uno de ellos es su actuación dentro del manejo de incertidumbre en los sistemas expertos. Las redes Bayesianas permiten encontrar, de una manera consistente, relaciones probabilistas entre variables. El hecho de utilizar una representación gráfica para la explicación del modelo hace que las Redes Bayesianas sean una herramienta realmente muy atractiva en su uso, como representación del conocimiento. No sólo modelan de forma cualitativa el conocimiento, sino que además expresan de forma numérica la fuerza de las relaciones entre las variables. Esta parte cuantitativa del modelo suele especificarse mediante distribuciones de probabilidad como una medida de la creencia que tenemos sobre las relaciones entre variables de modelo (Césari 2006).

2.1.4.1 Teorema de Bayes

El teorema de Bayes es un procedimiento para obtener probabilidades condicionales (probabilidades de ocurrencia de acontecimientos condicionadas a la ocurrencia de otros acontecimientos). La probabilidad condicional de un evento es una probabilidad obtenida con la información adicional de que algún otro evento ya ha ocurrido. Usamos $P(B|A)$ para denotar la probabilidad condicional del evento B que ocurre, dado que el evento A ya ha ocurrido. La siguiente fórmula se utiliza para encontrar $P(B|A)$:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (2.1)$$

La explicación intuitiva para encontrar una probabilidad condicional a partir de la fórmula anterior sería: la probabilidad condicional de B dada A se puede encontrar asumiendo que el evento A ha ocurrido y, trabajando bajo esa suposición, calcular la probabilidad de que ocurra el evento B. Por lo tanto, la aplicación del teorema de Bayes es adecuada para revisar un valor de probabilidad basado en información adicional que se obtiene posteriormente. La clave para entender la esencia del teorema de Bayes es percibir que estamos tratando con sucesos secuenciales, donde se obtiene nueva información adicional para un evento posterior. Esta nueva información se utiliza para revisar la probabilidad del evento inicial. En este contexto, los términos probabilidad previa y probabilidad posterior son de uso común y por lo tanto es necesario definirlos. Una probabilidad previa es un valor de probabilidad inicial obtenido originalmente

antes de obtener cualquier información adicional mientras que una probabilidad posterior es un valor de probabilidad que ha sido revisado usando información adicional que se obtiene posteriormente (Joyce 2019; Martínez 2017; Rincón 2014).

2.1.4.2 Definición formal de las RBs

Las RBs o probabilísticas se fundamentan en la teoría de la probabilidad y combinan la potencia del Teorema de Bayes con la expresividad semántica de los grafos dirigidos; las mismas permiten representar un modelo causal por medio de una representación gráfica de las independencias/dependencias entre las variables que forman parte del dominio de aplicación (Felgaer 2004).

A continuación, se enlistan algunas definiciones y terminología de las Redes Bayesianas (se utilizarán letras mayúsculas para denotar los nodos (X) y las correspondientes letras minúsculas para designar sus posibles estados (x_i)) (Normandi Rocío Tirado Ríos, Freddy Enrique Triana Litardo 2016):

- **Nodo:** Un nodo X es una variable aleatoria que puede tener varios estados x_i . La probabilidad de que el nodo X este en el estado x se denotará como $P(x) = P(X = x)$.
- **Arco:** es la unión entre dos nodos y representa la dependencia entre dos variables del modelo. Un arco queda definido por un par ordenado de nodos (X, Y).
- **Nodo padre:** El nodo X es un padre del nodo Y , si existe un arco (Y, X) entre los dos nodos.
- **Nodo hijo:** El nodo Y es un hijo del nodo X , si existe un arco (Y, X) entre los dos nodos.
- **Probabilidad a priori:** es la probabilidad de una variable en ausencia de evidencia.
- **Probabilidad a posteriori:** es la probabilidad de una variable condicionada a la existencia de una determinada evidencia; la probabilidad a posteriori de X cuando se dispone de la evidencia e se calcula como $P(X|e)$.
- **Probabilidad conjunta:** Dado un conjunto de variables $\{X, Y, \dots, Z\}$ la probabilidad conjunta especifica la probabilidad de cada combinación posible de estados de cada variable $P(x_i, y_j, \dots, z_k) \forall j, \dots, k$, de manera que se obtiene la ecuación:

$$\sum_{i,j,\dots,k} P(x_i, y_j, \dots, z_k) = 1 \quad (2.2)$$

- **Probabilidad condicional:** Dadas dos variables X e Y, la probabilidad de que ocurra y_j dado que ocurrió el evento x_i es la probabilidad condicional de Y dado X y se denota como $P(y_j | x_i)$. La probabilidad condicional por definición es:

$$P(y_j | x_i) = \frac{P(y_j, x_i)}{P(x_i)}, \text{ dado } P(x_i) > 0 \quad (2.3)$$

Análogamente, si se intercambia el orden de las variables:

$$P(x_i | y_j) = \frac{P(y_j, x_i)}{P(y_j)} \quad (2.4)$$

A partir de las dos fórmulas anteriores se obtiene:

$$P(y_j | x_i) = \frac{P(y_j)}{P(x_i, y_j)} P(x_i) \quad (2.5)$$

Esta expresión se conoce como el Teorema de Bayes que en su forma general tenemos:

$$P(y_j | x_i) = \frac{P(y_j)P(x_i | y_j)}{\sum_j P(x_i | y_j)P(y_j)} \quad (2.6)$$

Al denominador se le conoce como el Teorema de la Probabilidad Total. En las redes Bayesianas el conjunto de valores que componen la probabilidad condicional de un hijo dados sus padres, se representa en las llamadas TPCs.

- **Independencia:** Dos variables X e Y son independientes si la ocurrencia de una no tiene que ver con la ocurrencia de la otra. Por definición se cumple que Y es independiente de X si y sólo si

$$P(y_j | x_i) = P(y_j)P(x_i) \forall i, j \quad (2.7)$$

Esto implica que:

$$P(y_j|x_i) = P(y_j)\forall_i,j \quad (2.8)$$

$$P(x_i|y_j) = P(x_i)\forall_i,j \quad (2.9)$$

- **Observación:** es la determinación del estado de un nodo ($X = x$) a partir de un dato obtenido en el exterior del modelo.
- **Evidencia:** es el conjunto de observaciones $e = \{X=x, Y=y, \dots Z=z\}$ en un momento dado.

2.1.4.3 Aplicación de RBs en la Ingeniería de Software

Las RBs son cada vez más populares en Ingeniería de Software, inteligencia artificial y estadística. Se han aplicado con éxito a dominios como medicina, evaluación de riesgos, visión, diagnósticos de sistemas y redes, etc. (Rodríguez and Dolado 2007).

Muchas de las actividades en la Ingeniería del Software utilizan valores inciertos o probabilísticos. Diversas técnicas estadísticas y la teoría de la probabilidad han sido aplicadas a la Ingeniería del Software desde sus inicios. Más recientemente, modelos gráficos, que combinan probabilidad y teoría de grafos. Las RBs son aplicadas a la toma de decisiones en dominios donde la incertidumbre representa un papel importante, como es el caso de la ingeniería del software.

En la ingeniería del software las RBs han sido usadas en diferentes áreas como (Rodríguez and Dolado 2007): estimación del esfuerzo y calidad, pruebas de software, fiabilidad, interfaces gráficos e interacción con el usuario, entre otros.

Las RBs tienen un número de características y ventajas que hacen que sean apropiadas para la ingeniería del software (Rodríguez and Dolado 2007):

- Utiliza una representación gráfica para la representación del modelo (López-Martínez et al. 2017).
- Están formadas por componentes cualitativos y cuantitativos (López-Martínez et al. 2017).
- Son utilizadas en proyectos de desarrollo de software para estimación de esfuerzo, evaluación de confiabilidad, predicción, entre otras aplicaciones (Dragicevic, Celar, and Turic 2017).

- Permiten encontrar de una manera consistente, relaciones probabilísticas entre variables (López-Martínez et al. 2017).
- Pueden modelar grados de certidumbre, en vez de valores exactos. Por tanto, permiten modelar la incertidumbre de manera efectiva y explícitamente, por lo que pueden realizar buenas predicciones con información incompleta. Aborda automáticamente la incertidumbre y el riesgo debido a su propia naturaleza estadística (Dragicevic, Celar, and Turic 2017).
- Pueden hacer inferencia en ambos sentidos, es decir, las variables de entrada pueden ser usadas para predecir las variables de salida y viceversa (Rodríguez and Dolado 2007).
- Dado un conjunto de evidencias, las redes Bayesianas permiten fácilmente calcular la sensibilidad de ciertas variables, simplemente modificando las evidencias (Rodríguez and Dolado 2007).
- La salida de una RB es una probabilidad de distribución en vez de valores únicos. Este tipo de información se puede usar para medir la confianza que podemos depositar en la salida de la RB (Rodríguez and Dolado 2007).

Las RBs son especialmente adecuadas para modelar la estimación del esfuerzo y pueden contribuir significativamente a la gestión de proyectos de software por lo que la aplicación de RBs en la estimación del esfuerzo en el desarrollo de software está ganando popularidad especialmente en los últimos tiempos. Las RBs se han utilizado ampliamente para predecir el esfuerzo de software. Las razones fundamentales para el uso de este enfoque es que permite a los analistas incorporar atributos causales del proceso, así como combinar medidas cualitativas y cuantitativas, superando así las conocidas limitaciones de los métodos tradicionales de métricas de software (Karna and Gotovac 2015).

2.2 Trabajos relacionados

Se han realizado muchas investigaciones y se han planteado muchas propuestas para la estimación del esfuerzo. Algunos de ellos se han centrado en la complejidad como uno de los atributos más importantes para la estimación. Se han utilizado diferentes técnicas y algoritmos matemáticos para obtener mejores estimaciones. Esta sección describe artículos relacionados con

la estimación de la complejidad. Algunos de los estudios tienen un enfoque similar al nuestro. En la Tabla 1 se muestran algunas características de los trabajos relacionados.

Tabla 1. Trabajos relacionados

Autores	Método utilizado	Atributos	Aplicación
(Dragicevic, Celar, and Turic 2017)	Modelo Bayesiano	<ul style="list-style-type: none"> ▪ Complejidad del formulario ▪ Complejidad de la función ▪ Complejidad de informes ▪ Especificación de calidad. ▪ Tipo de tarea ▪ Habilidades de desarrollador 	Proyectos de desarrollo de software
(Alostad, Abdullah, and Aali 2017)	Modelo basado en lógica difusa	<ul style="list-style-type: none"> ▪ Experiencia del equipo ▪ Complejidad de la tarea ▪ Tamaño de la tarea ▪ Precisión en la estimación 	Proyectos de desarrollo de software
(Zahraoui and Janati Idrissi 2015)	Método propio	<ul style="list-style-type: none"> ▪ Complejidad ▪ Tamaño ▪ Prioridad 	Proyectos de desarrollo de software
(López-Martínez et al. 2017)	Modelo Bayesiano	<ul style="list-style-type: none"> ▪ Tiempo ▪ Esfuerzo ▪ Experiencia ▪ Prioridad ▪ Valor de historia de usuario 	Historias de usuario

En (Dragicevic, Celar, and Turic 2017) se presenta una estimación de esfuerzo en proyectos de software. Los autores utilizaron una RB para predecir el esfuerzo en las tareas. El modelo incluye seis atributos: tipo de tarea, requisitos de complejidad (formulario de complejidad, función de complejidad, informe de complejidad), especificación de calidad y habilidades de desarrollador. El modelo fue validado usando una base de datos de 160 tareas de proyectos reales. El modelo propuesto por estos autores obtuvo el 90% de precisión en la predicción del esfuerzo en tareas. Los atributos utilizados por los autores se centran en aspectos de la tarea y técnicos que podrían ser difíciles de identificar por los desarrolladores novatos; además, los autores consideraron las habilidades del desarrollador en un solo atributo.

El modelo propuesto por los autores en (Alostad, Abdullah, and Aali 2017) está basado en lógica difusa para mejorar la estimación del esfuerzo. Los autores consideraron cuatro atributos:

la experiencia del equipo de desarrollo, la complejidad de la tarea, el tamaño de la tarea y la precisión de la estimación. Se obtuvo un nivel de precisión de hasta el 60%. Debido a la precisión mejorada en las estimaciones de los desarrolladores en cada sprint, el autor considera que el uso del modelo obtendría buenos resultados cuanto más se usa, ya que se vuelve más preciso con el tiempo y ofrece una mejor estimación del esfuerzo. Las categorías utilizadas para cada atributo fueron términos lingüísticos; sin embargo, este modelo examinó solo cuatro elementos y determinar la complejidad de una HU implica incluso más atributos que los mencionados por el autor.

El trabajo de (Zahraoui and Janati Idrissi 2015) utiliza un método propio para estimar esfuerzo en proyectos de Scrum. La investigación se centra en ajustar puntos de historia a una nueva medida de puntos de historia ajustada. Este trabajo utilizó solo 3 atributos: complejidad, tamaño y prioridad, tomando en cuenta muy pocos aspectos para realizar una estimación.

En (López-Martínez et al. 2017) se contemplaron 5 atributos: tiempo, esfuerzo, experiencia, prioridad y valor de la historia. Se utilizó una estructura de RB; cada atributo representa un nodo en la red, y el peso de sus relaciones se representa en los bordes. Este modelo se basa en el conocimiento de expertos para estimar historias de usuarios, que se utiliza para determinar los pesos de los bordes. Este modelo se utiliza a través de preguntas que representan variables en una RB; los valores se obtienen de los miembros del ES durante cada sesión de estimación. Después de una validación utilizando 19 HU este trabajo alcanzó un 87% de precisión. La principal desventaja de este modelo es que los autores consideraron los atributos como uno solo, sin descomponer, por ejemplo, la experiencia como un atributo único, mientras que cada uno de ellos podría dividirse en más sub-atributos.

Como podemos observar, la mayoría de los estudios reportados en la literatura considera aspectos importantes para estimar, sin embargo, presentan dos aspectos importantes que podrían estar en desventaja:

- 1) no hay consenso sobre el significado de cada atributo y el nivel de descomposición, por ejemplo, la complejidad.
- 2) Nadie se centra en los aspectos personales de los desarrolladores.

El presente estudio está inspirado en el modelo propuesto en (López-Martínez et al. 2017) y propone mejorar la descomposición de algunos atributos en sub-atributos, como la complejidad y los aspectos personales.

Capítulo 3

3. Modelo propuesto

Esta sección explica cómo se llevó a cabo la Revisión Sistemática de la Literatura (RSL) y la selección de los atributos considerados para el modelo utilizado en esta propuesta.

3.1 Revisión Sistemática de la Literatura

Previamente en (Durán et al. 2020) se realizó una RSL basada en la metodología B. Kitchenham (Kitchenham, Budgen, and Brereton 2018) con el objetivo de identificar los atributos considerados más importantes para la estimación de historias de usuario. A continuación, se describe brevemente el proceso que se realizó.

3.1.1 Formulación de preguntas de investigación

Se definieron dos preguntas de investigación, las cuales ayudaron a la identificación de los atributos más importantes. Las preguntas se muestran a continuación:

- ¿Qué atributos se consideran para estimar complejidad en las historias de usuario?
- ¿Cuál es el método utilizado para estimar la complejidad?

3.1.2 Definición de la estrategia de búsqueda

Se tomaron en cuenta diversas bibliotecas de investigación científica. Se utilizó una estructura de búsqueda basada en palabras claves y conectores lógicos: (Complexity AND Estimation AND (Scrum OR User Stories OR Project))

3.1.3 Definición de criterios de inclusión y exclusión

Se incluyeron trabajos publicados entre 2016 y 2018, publicaciones que responden directamente las preguntas de investigaciones, publicaciones que describen estudios empíricos y publicaciones que examinan la estimación de la complejidad, por otro lado se excluyeron

publicaciones no escritas en inglés, publicaciones no relacionadas con la complejidad en proyectos de software, publicaciones que no son revisiones de la literatura y artículos de opinión o puntos de vista.

3.1.4 Evaluación de la calidad

Se formularon algunas preguntas para evaluar la calidad de la estrategia de revisión. A continuación, se muestran las preguntas seleccionadas:

- QA1: ¿Describe el artículo los atributos utilizados para la estimación de complejidad? El estudio describe los atributos utilizados para estimar, y estos atributos puede ser importante en la estimación de las historias de usuario.
- QA2: ¿Los atributos utilizados en el estudio son válidos y confiables? Los atributos mencionados son apoyados por los autores, es decir, que son recomendados por los expertos en el área.

Cada trabajo se evaluó utilizando las mismas preguntas; solo se seleccionaron los trabajos que cumplían ambas preguntas calidad.

3.1.5 Resultados

Se encontró un total de 940 artículos en las bibliotecas de investigación científica. Estos artículos fueron revisados por título, resumen y vista rápida, reduciendo el resultado a 47 artículos. Los 47 artículos se revisaron uno por uno tomando en cuenta las preguntas de investigación, los criterios de inclusión y exclusión y la evaluación de calidad; como resultado se seleccionaron 26 artículos.

De los artículos seleccionados se extrajeron 51 atributos, los cuales se clasificaron y ordenaron para formar una taxonomía, esto con el objetivo de identificarlos de una manera más sencilla. Se establecieron 5 clasificaciones: aspectos personales (Hsu et al. 2016; Scott and Pfahl 2018; Zare, Khademi Zare, and Fallahnezhad 2016), complejidad técnica (Abdou, Yong, and Othman 2016; Bosch-Rekveltdt, Bakker, and Hertogh 2018; M., Yong, and Othman 2018; Qazi et al. 2016; Singh 2016), proyecto (Hsu et al. 2016; Zare, Khademi Zare, and Fallahnezhad 2016), equipo (Abdou, Yong, and Othman 2016) y complejidad de la organización (Bosch-Rekveltdt, Bakker, and Hertogh 2018; M., Yong, and Othman 2018; Qazi et al. 2016), ya que fueron las que más se encontraron en los artículos seleccionados; también se consideró lo sugerido por la

literatura y la opinión de algunos autores. Asimismo, se analizaron los métodos utilizados para estimar en los artículos seleccionados, se encontró que la mayoría de los autores utilizan métodos propios para realizar estimaciones.

3.2 Atributos seleccionados

Debido a que actualmente la mayoría de los métodos para estimar historias de usuarios se basan principalmente en aspectos técnicos o de proyecto (Ramirez-Noriega et al. 2016), se propone este modelo que considera principalmente los atributos personales de los desarrolladores, sin descuidar los aspectos técnicos. Se analizaron los atributos y clasificaciones de la revisión sistemática de la literatura (Durán et al. 2020) realizada previamente.

Debido a que el modelo propuesto se enfoca en los aspectos humanos, se analizaron los atributos contemplados en la categoría de aspectos personales. Se seleccionaron los más relevantes, incluso sugeridos por los expertos: experiencia con tecnología (Rath, Acharya, and Satapathy 2016; Zare, Khademi Zare, and Fallahnezhad 2016), experiencia con el lenguaje (Rath, Acharya, and Satapathy 2016; Sholiq, Dewi, and Subriadi 2017), experiencia en proyectos anteriores (Alostad, Abdullah, and Aali 2017; Hsu et al. 2016; López-Martínez et al. 2017), habilidades de desarrollador (Dragicevic, Celar, and Turic 2017; Zare, Khademi Zare, and Fallahnezhad 2016) y conocimiento del tema del proyecto (Sholiq, Dewi, and Subriadi 2017). Se tomaron en cuenta estos atributos ya que desde el punto de vista de la literatura estos están presentes con más frecuencia en el proceso de estimación. En Scrum, los aspectos personales son los más importantes debido a que se considera que son atributos muy influyentes en el proceso de estimación, en base a esto, un desarrollador con experiencia puede estimar de manera más fácil una HU que un desarrollador sin experiencia (Lopez-Martinez et al. 2018; López-Martínez et al. 2017). Debido a que 3 de los atributos seleccionados forman parte de la experiencia, estos se agruparon para formar un nodo de experiencia.

Aunque el enfoque del modelo propuesto son los aspectos humanos, se seleccionaron algunos atributos relacionados con la complejidad técnica, ya que los autores describen la importancia de incluir este tipo de aspectos para la estimación (Abdou, Yong, and Othman 2016; Dragicevic, Celar, and Turic 2017), estos atributos nos ayudan a considerar riesgos, aspectos

técnicos y de calidad para llevar a cabo una HU. Se selecciono la dependencia (Kayes, Sarker, and Chakareski 2016; Singh 2016; Tanveer, Guzmán, and Engel 2017) y el tamaño (Alostad, Abdullah, and Aali 2017; Malgonde and Chari 2018) de las HU, ya que estos atributos son importantes para conocer la complejidad técnica de una HU, además de ser atributos comúnmente utilizados y conocidos en el proceso de estimación. En el análisis, se observó que el nodo tamaño podía desglosarse más, debido a que existe diferentes formas de estimar el tamaño de una HU. Al nodo tamaño se le añadieron dos sub-nodos: funciones de conexión (Dragicevic, Celar, and Turic 2017) y funciones de datos (Dragicevic, Celar, and Turic 2017; Pandit and Tahiliani 2015).

Los atributos seleccionados para el primer nivel se describen en la Tabla 2. Los atributos seleccionados se agruparon en nodos para que estén organizados por su naturaleza. El modelo final se muestra en la Figura 9. Con este modelo pretendemos mejorar la precisión en la estimación de HU.

Tabla 2. Atributos seleccionados para el primer nivel

Atributo de primer nivel	Definición
Experiencia con la tecnología	El desarrollador domina las herramientas, métodos y tecnologías para el desarrollo de la Historia de usuario de forma efectiva. La experiencia con la tecnología aporta calidad y rapidez al término de una Historia de usuario.
Experiencia en proyectos previos	Conocimiento que un desarrollador posee en base a proyectos similares o anteriores. El desarrollo de trabajos previos similares aporta calidad y rapidez al término de una Historia de Usuario.
Experiencia en el lenguaje de programación	El desarrollador domina y es capaz de usar en forma efectiva los recursos de dicho lenguaje, y produce un resultado organizado y valioso. El desarrollador conoce las limitaciones de dicho lenguaje. La experiencia con el lenguaje de programación aporta calidad y rapidez al término de una Historia de usuario.
Habilidades del desarrollador	Conocimientos y capacidades que posee un desarrollador (capacidad de análisis, creatividad, etc.).
Conocimientos del tema del proyecto	El desarrollador conoce y domina la naturaleza y los conceptos más importantes del tema del que se realizara el proyecto.
Dependencia	Que tan importante es una Historia de usuario debido a que otras Historias de usuario dependen de esta.
Funciones de conexión	Son los requerimientos de procesamiento del usuario como entradas, salidas y consultas.
Funciones de datos	Son los requerimientos de almacenamiento del usuario, como archivos lógicos internos y archivos de interfaz externa.



Figura 9. Modelo propuesto

3.3 Técnicas para representar el modelo

Fue necesario seleccionar una técnica para representar el modelo propuesto. Buscamos una técnica que nos permitiera representar el conocimiento de los expertos a través de información cualitativa y cuantitativa y que se adaptara a nuestro modelo. Seleccionamos las RBs debido a que esta técnica requiere información cualitativa y cuantitativa y trabaja a través de nodos y las relaciones entre ellos.

Una RB es un modelo gráfico que describe relaciones probabilísticas entre variables relacionadas causalmente (Dragicevic, Celar, and Turic 2017). Se fundamentan en la teoría de la probabilidad y permiten representar un modelo causal por medio de una representación gráfica de las dependencias entre las variables (López-Martínez et al. 2017). Se componen de dos partes (Rodríguez and Dolado 2007):

- La parte cualitativa, es una estructura gráfica (grafo) que describe las posibles entidades (variables) y dependencias entre ellas.
- La parte cuantitativa está compuesta por probabilidades condicionadas que representan la incertidumbre del problema.

Nuestro modelo contaba con información cualitativa, es decir con una estructura jerárquica que representa variables y sus relaciones; y con información cuantitativa, la cual obtendríamos con el grado de dependencia entre cada nodo-relación. Dado nuestro modelo, se puede hacer inferencia

bayesiana; esto es, estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas. Las RBs manejan la incertidumbre con probabilidades, con esto nos brindan una idea de que tan probable es que ocurra un evento. (Sucar 2011).

Trabajos relacionados utilizaron esta técnica, lo cual respalda el hecho de que esa técnica ayudo a solucionar problemas similares. Autores como Martínez (López-Martínez et al. 2017) y Dragicevic (Dragicevic, Celar, and Turic 2017) consideraron la incertidumbre de la estimación a través de las RBs.

3.4 Construcción de la red Bayesiana

Esta sección muestra cómo se construyó la RB. La parte cualitativa representa la estructura formada por nodos y las relaciones y dependencias entre nodos, esta parte se desarrolla en función del grupo de variables seleccionadas del artículo (Durán et al. 2020). La parte cuantitativa se compone de TPCs, para crear esto, los profesionales dieron sus opiniones.

3.4.1 Parte cualitativa

La parte cualitativa de la RB está compuesta por nodos y la relación entre ellos. Las relaciones entre los nodos se muestran en la Figura 9. La RB está formada por una estructura jerárquica que está organizada por niveles, cada nivel representa un grupo de variables o nodos:

- Atributos de primer nivel: estos nodos son los que recopilan información para la estimación de la complejidad. Se definieron tres estados para estos atributos (poco, medio, alto).
- Atributos de segundo nivel: estas variables ayudan a organizar los nodos de primer nivel jerárquicamente.
- Atributo de complejidad de historias de usuario: esta variable muestra el valor final para medir la estimación de complejidad.

3.4.2 Parte cuantitativa

Para formar la parte cuantitativa de nuestra RB, es necesario cubrir tres aspectos importantes: valores entre las relaciones de los atributos, escalas de valores para los atributos de primer nivel y la construcción de las TPCs. Para obtener el valor entre las relaciones de los atributos fue necesario consultar la opinión de los expertos.

3.4.2.1 Valores de relación entre variables

Para obtener los valores de las relaciones entre las variables fue necesario consultar la opinión de los expertos de Scrum, para lo cual se diseñó y aplicó una encuesta que se describe en la siguiente sección.

3.4.2.1.1 Encuesta para determinar relación entre atributos

El objetivo de la encuesta fue validar la importancia y el valor de los atributos seleccionados dentro del modelo. Se preguntó a los encuestados el grado de aceptación de los atributos seleccionados. La encuesta se realizó digitalmente mediante formularios de Google. La encuesta se aplicó a 21 profesionales de la industria y 19 estudiantes de la carrera de ingeniería en computación de la Universidad Autónoma de Baja California. Desde el principio se contempló tener en cuenta solo la opinión de los profesionales para la construcción de la RB, sin embargo, también se aplicó a los estudiantes para saber cómo pensaban y si diferían de la opinión de los profesionales, sin embargo, para la creación de la RB solo se tomó en cuenta la opinión de los profesionales.

Preparamos 18 preguntas para conocer el grado de aceptación de los atributos propuestos y preguntas para recopilar información general sobre la experiencia de los encuestados como, experiencia de los encuestados tanto en proyectos de desarrollo como en el uso de Scrum, cantidad de proyectos en los que han trabajado, rol dentro de la empresa que labora, entre otras. Se utilizó una escala Likert de 5 puntos (Totalmente en desacuerdo, en desacuerdo, Neutral, de acuerdo, totalmente de acuerdo) para medir el grado de aceptación de cada atributo (Para más detalles ver el apéndice A.1). El diseño de la encuesta se validó teniendo en cuenta las recomendaciones enlistadas en (Linåker, Johan; Sulaman, Sardar Muhammad; Maiani de Mello, Rafael; Höst 2015).

Para validar los resultados de la encuesta aplicada se utilizó la técnica alfa de Cronbach (Celina and Campo 2016). Con un total de 11 elementos analizados obtuvimos un alfa de Cronbach de 0.912 para los estudiantes y 0.866 para los profesionales.

Los resultados obtenidos se muestran en la Tabla 3 (gris claro para profesionales y gris oscuro para estudiantes) la cual muestra los porcentajes que se obtuvo para cada atributo en cada escala Likert. Cada atributo fue validado considerando cuánto se está de acuerdo con el uso de ese atributo. Recibimos la respuesta de 19 estudiantes de la Universidad Autónoma de Baja California, de sexto y séptimo semestre de la carrera de Ingeniería en Computación, que tienen un promedio de 15 meses de experiencia desarrollando proyectos de software, 4 proyectos de desarrollo y aproximadamente 6 meses usando Scrum. Por otro lado, recibimos la respuesta de 21 profesionales de diversas empresas de software de la región, que tienen un promedio de 57 meses de experiencia desarrollando proyectos de software, 37 proyectos de desarrollo y aproximadamente 32 meses usando Scrum. Ambos grupos de encuestados mencionaron jugar el rol de desarrollador dentro de sus proyectos y el uso de PP como método para estimar las historias de los usuarios.

Tabla 3. Grado de aceptación para cada atributo

	Totalmente en desacuerdo		En desacuerdo		Neutral		De acuerdo		Totalmente de acuerdo	
	0	0	0	5.3	4.8	5.3	38.1	10.5	57.1	78.9
Experiencia con la tecnología	0	0	0	5.3	4.8	5.3	38.1	10.5	57.1	78.9
Experiencia con el lenguaje	0	0	0	0	4.8	10.5	42.9	15.5	52.4	73.7
Experiencia en proyectos previos	0	5.3	0	0	19.0	15.8	47.6	36.8	33.3	42.1
Habilidades del desarrollador	0	0	0	5.3	9.5	15.8	61.9	42.1	28.6	36.8
Conocimiento del tema del proyecto	0	0	0	0.	14.3	26.3	76.2	42.1	9.5	31.6
Dependencia	0	0	4.8	5.3	14.3	15.8	57.1	52.6	23.8	26.3
Funciones de conexión	0	0	4.8	0	14.3	21.1	42.9	36.8	38.1	42.1
Funciones de datos	0	0	4.8	0	23.8	15.8	28.6	52.6	42.9	31.6

Adicional a las preguntas para evaluar los atributos, se incluyeron preguntas para que los encuestados dieran sugerencias sobre otros atributos no contemplados en el modelo planteado. Por parte de los estudiantes se mencionaron los siguientes atributos: trabajo en equipo, experiencia en Scrum, tecnología a utilizar y claridad en los requisitos. Por otro lado, los profesionales mencionaron los siguientes atributos: trabajo en equipo, capacidad de aprendizaje, comunicación, tecnología involucrada, tipo de tarea y redacción de los requerimientos.

3.4.2.1.2 Aplicación de ecuaciones

Para calcular los pesos entre variables aplicamos las ecuaciones y definiciones utilizadas en (López-Martínez et al. 2017; Martinez 2017).

Definición 3.1 Sea $V = \{v_1, \dots, v_n\}$ el conjunto que denota la opinión de los expertos. Donde el elemento $v = (weight, frequency)$ representa el peso dado en un punto a la escala Likert (*weight*) y la frecuencia de la escala (*frequency*).

La escala Likert de 5 puntos define la variable n en 5 debido al número de puntos. Basado en la definición 3.1, la ecuación 3.1 establece la relación no normalizada de una variable p (Definición 3.2)

$$P_{p \in P} = \frac{\sum_{v \in V} (v_{weight} v_{frequency})}{\sum_{v \in V} v_{frequency}} \quad (3.1)$$

Definición 3.2 Sea $P = \{p_1, \dots, p_m\}$ el conjunto de valores que representan a los nodos padres, estos nodos tienen influencia causal sobre el mismo nodo hijo. La ecuación 3.2 obtiene las relaciones normalizadas considerando el conjunto de padres y sus valores calculados con la ecuación 3.1.

$$P_{norm_{p \in P}} = \frac{P}{\sum_{P \in P} P} \quad (3.2)$$

Definición 3.3 Sea $A = \{a_1, \dots, a_q\}$ el conjunto de variables de segundo nivel; experiencia, tamaño, complejidad técnica y complejidad en las historias de usuario, estas variables tienen dos estados: alto y bajo. Sus valores se calculan a través de la ecuación 3.3.

$$a_{a \in A} = \frac{\sum P^{EP} P}{|P|} \quad (3.3)$$

Cuando el conjunto P representa a los padres de la variable a , el total de variables que tienen influencia sobre otra se representa con $|P|$, y el valor no ponderado se representa con p . El valor normalizado se calcula con la ecuación 3.2, pero, teniendo en cuenta la variable a en lugar de p . La Tabla 4 muestra las frecuencias organizadas por atributo; peso asignado a la escala Likert (muy en desacuerdo = 1, en desacuerdo = 2, neutral = 3, de acuerdo = 4, totalmente de acuerdo = 5). La última fila muestra el peso de cada atributo. Tomaremos como ejemplo el nodo de experiencia para explicar cómo se obtuvieron los pesos.

Tabla 4. Pesos y frecuencias de los atributos

Peso	Escala Likert	Frecuencia							
		Exp. Con Tecnología	Exp. Con Lenguaje	Exp. en proy. previos	Habilidades del desarrollador	Con. tema. proyecto	Dependencia	Funciones de conexión	Funciones de datos
0	1	0	0	0	0	0	0	0	0
0.25	2	0	0	0	0	0	1	1	1
0.5	3	1	1	4	2	3	3	3	5
0.75	4	8	9	10	13	16	12	9	6
1	5	12	11	7	6	2	5	8	9
Peso		0.347	0.343	0.310	0.253	0.234	0.490	0.504	0.496

Aplicamos la ecuación 3.1 para obtener el peso no estandarizado de cada atributo de primer nivel: experiencia con tecnología, experiencia con lenguaje y experiencia en proyectos anteriores.

$$Exp. Tecnología = \frac{(0 \times 0) + (0.25 \times 0) + \dots + (1 \times 12)}{21} = 0.881 \quad (3.4)$$

$$Exp. Lenguaje = \frac{(0 \times 0) + (0.25 \times 0) + \dots + (1 \times 11)}{21} = 0.869 \quad (3.5)$$

$$Exp. Proyec. Previos = \frac{(0 \times 0) + (0.25 \times 0) + \dots + (1 \times 7)}{21} = 0.786 \quad (3.6)$$

Para obtener el peso normalizado de cada atributo usaremos la ecuación 3.2. Es necesario considerar los nodos que son padres del mismo nodo hijo, en este caso, el nodo de experiencia.

$$Exp. tec_{norm} = \frac{0.881}{0.881 + 0.869 + 0.786} = 0.347 \quad (3.7)$$

$$Exp. leng_{norm} = \frac{0.869}{0.881 + 0.869 + 0.786} = 0.343 \quad (3.8)$$

$$Exp. proy_{norm} = \frac{0.786}{0.881 + 0.869 + 0.786} = 0.310 \quad (3.9)$$

Para obtener el peso del atributo de experiencia y los otros atributos de segundo nivel, aplicaremos la ecuación 3.3.

$$Experiencia = \frac{0.881 + 0.869 + 0.786}{3} = 0.845 \quad (3.10)$$

Aplicaremos la ecuación 3.2 para obtener el peso estandarizado del atributo de experiencia. También debemos considerar a los otros padres que influyen en el mismo atributo y sus pesos no estandarizados. En este caso: habilidades del desarrollador (0.798), conocimiento del tema del Proyecto (0.738) y complejidad técnica (0.770).

$$Experiencia_{norm} = \frac{0.845}{0.845 + 0.798 + 0.738 + 0.770} = 0.268 \quad (3.11)$$

3.4.2.2 Valores a priori para atributos de primer nivel

Se eligieron escalas de valores para los nodos recolectores; es decir, qué valor o estado puede tener cada atributo de primer nivel. Se utilizaron tres escalas: baja, regular y alta. Se hizo una pregunta para cada nodo recolector (atributo de primer nivel), que debería responderse con una de las tres escalas propuestas (baja, regular, alta). En el apéndice A.2 se puede ver a detalle el cuestionario utilizado para estimar a través de la red, las preguntas formuladas para cada atributo, las escalas de valor utilizadas para cada atributo y su interpretación.

3.4.2.3 Construcción de las TPCs

Las TPCs muestran la probabilidad de que ocurra un evento basado en la combinación de las variables y el valor de sus estados (Lopez-Martinez 2017). En este paso, se define una TPC para cada nodo secundario de la estructura. Para obtener las TPCs aplicaremos las ecuaciones y definiciones utilizadas en (López-Martínez et al. 2017).

Definición 4 sea $S = \{s_1, s_2, \dots, s_{|S|}\}$ el conjunto de estados que puede tener una variable. Los valores ponderados para las variables fueron 0.170 (bajo), 0.330 (regular) y 0.500 (alto). La matriz en la ecuación 3.12 muestra el proceso para crear la TPC. El valor de la variable se multiplica por el valor de la escala para encontrar una ponderación $(p_1s_1, p \in P, s \in S)$. En segundo lugar, los resultados en el primer paso se combinan, cada combinación obtiene un valor único que agrega cada elemento de la combinación $((p_1s_1)+(p_2s_1)+ \dots + (p_ns_1))$.

$$W = \begin{pmatrix} w_{11} & \dots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nm} \end{pmatrix} \quad (3.12)$$

Donde:

$$\begin{aligned} w_{11} &= ((p_1s_1 + (p_2s_1) + \dots + (p_ns_1) \\ w_{12} &= ((p_1s_1 + (p_2s_1) + \dots + (p_ns_2) \\ w_{1m} &= ((p_1s_1 + (p_2s_1) + \dots + (p_ns_m) \end{aligned}$$

...

$$\begin{aligned} w_{n1} &= ((p_1s_m + (p_2s_m) + \dots + (p_ns_1) \\ w_{n2} &= ((p_1s_m + (p_2s_m) + \dots + (p_ns_2) \\ w_{nm} &= ((p_1s_m + (p_2s_m) + \dots + (p_ns_m) \end{aligned}$$

Después, la variable *max* representa el valor máximo de la matriz, esta variable se usa para obtener los valores finales de forma proporcional. El valor máximo divide la matriz en la ecuación 3.12, en la forma $W=W/max$, para obtener los valores finales. Estos valores representan el estado alto de las variables de segundo nivel y la variable complejidad en historias de usuario, el estado bajo es el valor complementario del estado alto. El proceso se repite para obtener la TPC de cada atributo de segundo nivel.

$$W = \frac{\begin{pmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{pmatrix}}{max} \quad (3.13)$$

Tomaremos como ejemplo el atributo tamaño para explicar cómo se obtuvo la TPC para este nodo. Las escalas de valores que se utilizaron fueron baja (0.17), regular (0.33) y alta (0.5). Los pesos para los atributos Funciones de conexión y funciones de datos son 0.504 y 0.496 respectivamente. Para obtener la matriz de la ecuación 3.12, primero es necesario multiplicar el peso del atributo (P) por cada valor de la escala (S), el resultado se muestra en la Tabla 5.

Tabla 5. Combinaciones de escalas y pesos para el nodo tamaño.

	P1 Funciones de conexión (0.504)	P2 Funciones de datos (0.496)
S1 Alto (0.5)	0.252	0.248
S2 Regular (0.33)	0.166	0.164
S3 Bajo (0.17)	0.086	0.084

Para formar la TPC es necesario contemplar todas las combinaciones posibles $((p1s1)+(p2s1)+ \dots +(pns1))$ entre los atributos que conforman el nodo tamaño. El resultado es sumado por cada combinación y debe dividirse por el valor máximo de la matriz para obtener los valores finales. Estos valores representan el estado alto de los atributos de segundo nivel, el estado bajo es el valor complementario del estado alto. La Tabla 6 muestra las posibles combinaciones que pueden existir en el nodo de tamaño y los resultados obtenidos.

Se repitió el mismo proceso para obtener la TPC de cada atributo de segundo nivel. Se utilizó el software GeNIe 2.4 academic para representar la RB, en el cual se capturaron las TPC de cada nodo. La Figura 10 muestra los resultados obtenidos para los pesos de cada relación entre nodos en la red y la RB resultante.

Tabla 6. Tabla de probabilidades condicionales para el nodo tamaño.

Combinaciones			Valor para nodo tamaño	
Funciones de conexión	Funciones de datos	Suma	Alto	Bajo
P1*S1	P2*S1	0.252 + 0.248 = 0,500	1,00	0,00
P1*S1	P2*S2	0.252 + 0.164 = 0,416	0,83	0,17
P1*S1	P2*S3	0.252 + 0.084 = 0,336	0,67	0,33
P1*S2	P2*S1	0.166 + 0.248 = 0,414	0,83	0,17
P1*S2	P2*S2	0.166 + 0.164 = 0,330	0,66	0,34
P1*S2	P2*S3	0.166 + 0.084 = 0,251	0,50	0,50
P1*S3	P2*S1	0.086 + 0.248 = 0,334	0,67	0,33
P1*S3	P2*S2	0.086 + 0.164 = 0,249	0,50	0,50
P1*S3	P2*S3	0.086 + 0.084 = 0,170	0,34	0,66

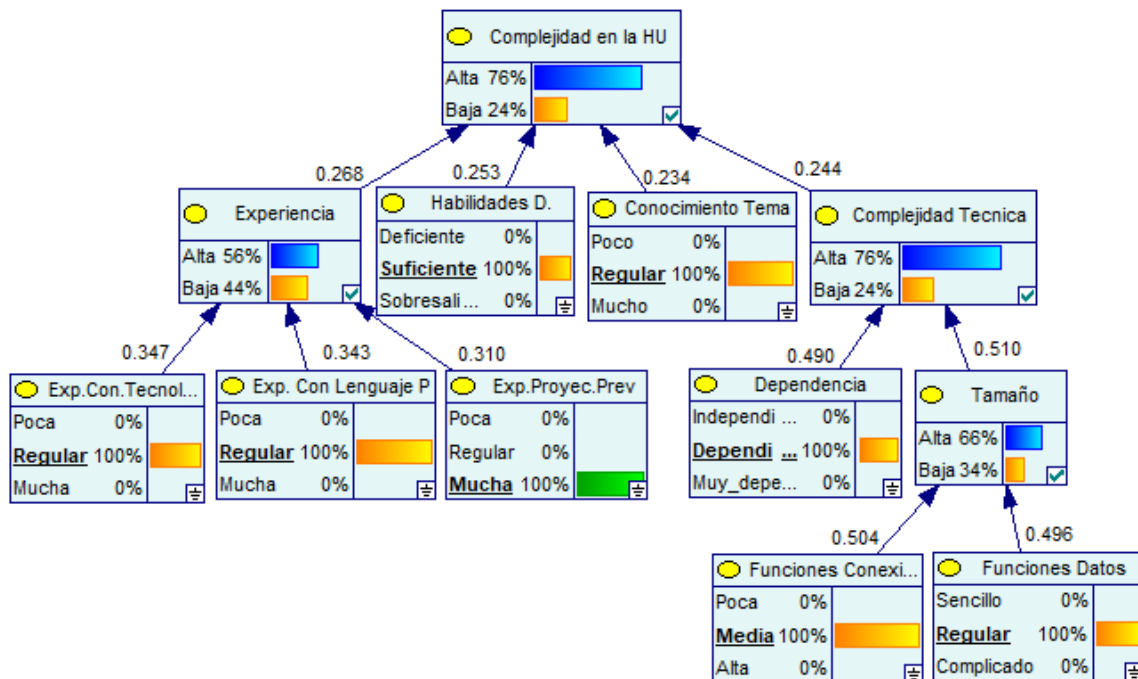


Figura 10. Red Bayesiana resultante

Capítulo 4

En esta sección se presenta los diferentes experimentos aplicados, así como los resultados obtenidos y la discusión sobre estos.

4. Pruebas del modelo

Para validar el modelo creado, se diseñaron dos experimentos diferentes, uno para estudiantes y otro para profesionales.

4.1 Experimentos

4.1.1 Estudiantes

El objetivo de este experimento fue obtener las estimaciones de las historias de usuario individuales utilizando el método tradicional (PP) y la red creada en nuestra propuesta. Tuvimos una muestra de 19 estudiantes del séptimo semestre de ingeniería en computación de la Universidad Autónoma de Baja California. Seleccionamos 19 porque fue la cantidad de alumnos inscritos en la asignatura de Ingeniería de software. También consideramos que con 19 estudiantes podríamos obtener una buena cantidad de datos, tomando como referencia la muestra de López que requirió solo 6 estudiantes (López-Martínez et al. 2017). Los estudiantes tienen conocimiento básico en Scrum y PP, un promedio de 15 meses de experiencia desarrollando proyectos de software y 6 meses utilizando Scrum. Previo a nuestro experimento el único método para estimar conocido por nuestro grupo muestra era PP. Analizamos un total de 112 historias de usuario definidas por los estudiantes. Para recopilar los datos, se crearon dos archivos de Excel en los que los estudiantes registraron sus estimaciones individualmente. Uno para realizar un seguimiento de las estimaciones convencionales y otro para las estimaciones realizadas con la red. El experimento se dividió en 3 partes:

- Parte 1: Una vez que se han descrito las historias de los usuarios, cada miembro del Equipo Scrum estima cada HU de forma individual y convencional (usando PP).
- Parte 2: para la segunda parte, pedimos responder individualmente las preguntas mostradas en el Apéndice A.2 para cada HU. Esta información fue introducida a nuestra red.

- Parte 3: Como datos adicionales, se solicitó registrar una tercera estimación de manera convencional. Esto consistió en volver a estimar la historia del usuario utilizando PP una vez que se realizó.

4.1.2 Profesionales

El objetivo de este experimento fue obtener las estimaciones de historias de usuarios individuales utilizando el método tradicional (PP) y la red creada en este trabajo. Tuvimos una muestra de 6 profesionales de diferentes empresas de la región con más de 5 años de experiencia desarrollando software y más de 2 años utilizando Scrum. Consideramos una muestra pequeña tomando como referencia la muestra utilizada en (López-Martínez et al. 2017) que requirió solo 4 profesionales.

Previo a nuestro experimento el único método para estimar conocido por nuestro grupo muestra era PP. Analizamos un total de 10 historias de usuario. Este experimento se realizó utilizando la herramienta de encuesta digital (formularios de Google) la cual fue enviada a cada experto de manera individual. El experimento se dividió en 3 partes:

- Parte 1: Se seleccionó un proyecto desarrollado por estudiantes y 10 historias de usuarios de este proyecto.
- Parte 2: Se proporcionó información importante sobre el proyecto a los profesionales (Descripción del proyecto, idiomas que se utilizarán, tecnologías, herramientas, etc.) y se pidió al experto que estimara, utilizando PP cada HU, aparentando que se desarrollaría, teniendo en cuenta la información proporcionada y su experiencia.
- Parte 3: Una vez realizada la estimación tradicional, se le pidió al experto que respondiera las 8 preguntas enumeradas en el Apéndice A.2 para cada HU considerando la información proporcionada y su experiencia.

4.2 Resultados

En el caso de los estudiantes, los resultados obtenidos fueron organizados por equipo (se registraron 4 equipos participantes), una pequeña parte de los resultados obtenidos por uno de los equipos se muestra en la Tabla 8. La columna PH (inicial) representa la estimación realizada por

los estudiantes antes de la realización de la historia del usuario de forma tradicional utilizando PP. La columna de PH (final) representa las estimaciones de los estudiantes después de finalizar la historia del usuario de forma tradicional (PP). La columna de complejidad (con RB) representa el valor obtenido por la RB según las respuestas dadas por los estudiantes para cada HU.

Debido a que en Scrum la escala de medida son los PH fue necesario trasladar nuestro resultado obtenido a través de la red (Complejidad) a PH, esto también con el fin de poder comparar nuestros resultados con los resultados obtenidos por el método tradicional (PP). Para eso seleccionamos los 9 puntajes y cartas más utilizadas en PP (1,2,3,5,8,13,20,40,100). Asignamos un número de carta de PP (del 1 al 9) a cada uno de los puntajes más utilizados, por lo que cada carta sería equivalente a un número de PH como se muestra en la Tabla 7.

Tabla 7. Equivalencia entra cartas y puntos de historia

# de Carta en Planning Poker	Puntos de historia
1	1
2	2
3	3
4	5
5	8
6	13
7	20
8	40
9	100

Convertimos la complejidad obtenida por la red a un número de carta en PP usando la ecuación 3.14. Con esta ecuación buscamos encontrar un valor desconocido en base a los valores que ya conocemos. En la cual, Y es el valor que estamos buscando y representamos el número de la carta en PP, X representa la complejidad obtenida por la RB, $X1$ representa el valor más bajo que se puede obtener en la RB, $X2$ representa el valor más alto que se puede obtener en la RB, $Y1$ representa la posición más pequeña de las cartas e $Y2$ la posición más grande de las cartas. El resultado se muestra en la columna #Carta en PP (con RB).

$$Y = Y1 + \left[\left(\frac{X - X_1}{X_2 - X_1} \right) (Y_2 - Y_1) \right] \quad (3.14)$$

La columna de PH (con RB) representa el valor equivalente (que se muestra en la Tabla 7) de acuerdo con el número de carta obtenida como resultado de la aplicación de la ecuación 3.14.

Los resultados obtenidos por los profesionales se muestran en la Tabla 9. La columna de PH (inicial) representa la estimación en una forma tradicional (PP) realizada antes de que se inicie la historia del usuario. La columna de complejidad (con RB) representa el valor obtenido por la RB de acuerdo con respuestas obtenidas para cada HU. Al igual que con los estudiantes, se aplicó la ecuación 3.14 y se consideraron las equivalencias que se muestran en la Tabla 7 para obtener la columna PH (con RB)

Tabla 8. Resultado de estudiantes

Caso	Sprint	Integrante	HU	PH (Inicial)	PH (Final)	Complejidad (con RB)	#Carta en PP (con RB)	PH (con RB)
1	3	Integrante 1	1	5	5	75	4	5
2	3	Integrante 2		3	5	67	3	3
3	3	Integrante 3		8	5	75	4	5
4	3	Integrante 4		5	5	82	5	8
5	3	Integrante 5		5	5	75	4	5
6	3	Integrante 1	2	5	8	73	4	5
7	3	Integrante 2		5	8	78	5	8
8	3	Integrante 3		5	8	78	5	8
9	3	Integrante 4		8	5	82	5	8
10	3	Integrante 5		13	5	76	4	5
11	5	Integrante 1	3	13	8	78	5	8
12	5	Integrante 2		20	13	84	6	13
13	5	Integrante 3		20	20	84	6	13
14	5	Integrante 4		13	20	87	6	13
15	5	Integrante 5		40	20	89	7	20
16	5	Integrante 1	4	20	40	89	7	20
17	5	Integrante 2		20	40	90	7	20
18	5	Integrante 3		40	40	89	7	20
19	5	Integrante 4		20	20	89	7	20
20	5	Integrante 5		40	40	93	7	20

Tabla 9. Resultados de profesionales

Caso	Nombre	HU	PH (Inicial)	Complejidad (con RB)	#Carta en PP (con RB)	PH (con RB)
1	Person 1	1	5	77	4	5
2	Person 2		3	67	3	3
3	Person 3		5	77	4	5
4	Person 4		1	56	1	1
5	Person 5		8	75	4	5
6	Person 6		8	81	5	8
7	Person 1	2	3	67	3	3
8	Person 2		2	62	2	2
9	Person 3		5	83	5	8
10	Person 4		2	63	2	2
11	Person 5		5	75	4	5
12	Person 6		8	79	5	8
13	Person 1	3	8	83	5	8
14	Person 2		1	60	1	1
15	Person 3		13	84	6	13
16	Person 4		1	61	1	1
17	Person 5		5	78	5	8
18	Person 6		8	77	4	5
19	Person 1	4	13	78	5	8
20	Person 2		1	60	1	1
21	Person 3		20	94	7	20
22	Person 4		1	60	1	1
23	Person 5		5	75	4	5
24	Person 6		5	73	4	5

Para validar los resultados obtenidos por ambos métodos, aplicamos pruebas de correlación de Spearman [45] utilizando SPSS. En el caso de los estudiantes, se analizaron las variables:

- EstimaciónIni representa la estimación inicial antes de hacer la HU.
- EstimaciónFin representa la estimación final, una vez que se realizó la HU.
- EstimaciónRB representa el resultado obtenido por la RB.

Las pruebas se aplicaron teniendo en cuenta las relaciones: *estimaciónIni-estimaciónRB* que determina la correlación entre la estimación inicial y la obtenida por la red y *estimaciónFin-estimaciónRB* que determina la correlación entre la estimación final y la obtenida por la red.

La prueba de Spearman reportó que la relación *estimaciónIni-estimaciónRB*, tiene una correlación moderada positiva ($r=0.659$, $p\text{-valor}=0$), mientras que para la relación *estimaciónFin-estimaciónRB* nos proporcionó una correlación fuerte positiva ($r= 0.799$, $p\text{-valor}=0$). Como se puede ver, la correlación entre la estimación final y la obtenida en la red es mayor, esto es bueno porque indica que lo obtenido por la red está muy relacionado con lo que el alumno estima una vez que se realizó la HU.

En el caso de los profesionales, se analizaron las variables:

- EstimaciónIni representa la estimación inicial antes de hacer la historia del usuario.
- EstimaciónRB representa el resultado obtenido por la RB.

La prueba se aplicó teniendo en cuenta las variables: *estimaciónIni-estimaciónRB* que determina la correlación entre la estimación inicial y la obtenida por la red. Las pruebas de Spearman reportó que la relación *estimaciónIni-estimaciónRB* tiene una correlación fuerte positiva ($r=0.887$, $p\text{-valor}=0$).

Si comparamos los resultados de estudiantes y profesionales, podemos deducir que la correlación en ambos casos fue buena, sin embargo, la correlación de los profesionales es mayor, esto porque los profesionales tienen más experiencia tanto desarrollando como estimando.

4.3 Discusión

Este estudio propone un modelo para estimar la complejidad de las historias de usuario considerando mayormente los aspectos personales de los desarrolladores a través del desglose del atributo experiencia, con el propósito de proporcionar a los equipos de desarrollo un método que favorezca positivamente la precisión. Este proceso es incierto y la poca o nula experiencia de los desarrolladores lo vuelve más complicado (López-Martínez et al. 2017). Se propone la hipótesis de que considerando más atributos correspondientes a la experiencia de los participantes en Scrum se contribuye a incrementar la precisión en la estimación de las historias de usuario. Los resultados de este estudio apoyan esta hipótesis.

Después de analizar los resultados obtenidos en las estimaciones de los desarrolladores tanto profesionales como estudiantes realizadas con el método tradicional (PP) pudimos percatarnos que ambas tenían un buen grado de correlación con el resultado obtenido a través de la RB.

Nuestro modelo estima alrededor del 88% de precisión como lo haría un desarrollador profesional. La precisión de nuestro modelo aumenta para los desarrolladores con mayor experiencia, lo cual es un resultado viable porque podemos confiar en nuestro modelo como confiaríamos en un desarrollador profesional.

La correlación entre la estimación de nuestro modelo y la estimación inicial del estudiante es del 65% de precisión; y la correlación entre la estimación de nuestro modelo y la estimación final del alumno (la cual denominamos estimación real, porque se realizó después de que los programadores estudiantes desarrollaran la HU) fue del 79% de precisión.

El modelo propuesto en este documento es más preciso (considerando la correlación obtenida por profesionales) que el modelo propuesto por Martínez (López-Martínez et al. 2017) (87%) y Alostad (Alostad, Abdullah, and Aali 2017) (60%). Nuestro modelo descompuso la experiencia del desarrollador en tres subcategorías: experiencia con la tecnología, con el lenguaje y experiencia con proyectos previos. Como lo dice Scrum la experiencia y los aspectos humanos son unos de los atributos más importantes no solo en el proceso de estimación, sino en todo el ciclo de Scrum (Mahnič and Hovelja 2012) y con nuestros resultados podemos respaldar esa percepción.

Por otro lado, Dragicevic (Dragicevic, Celar, and Turic 2017) alcanzo un 90% de precisión en la estimación, pero su trabajo se centró en la estimación del esfuerzo. Su modelo se centró en descomponer la complejidad en otras subcategorías. El modelo no considera la experiencia del desarrollador, se centró en recopilar las habilidades de los programadores, por lo tanto, no es viable compararnos, aun así, nuestra estimación estuvo cerca de la de Dragicevic.

En comparación con nuestro modelo, en (López-Martínez et al. 2017), Alostad (Alostad, Abdullah, and Aali 2017) y Dragicevic (Dragicevic, Celar, and Turic 2017) utilizaron atributos más genéricos para la estimación, a pesar de que estos atributos pueden desglosarse. Pudimos notar que si desglosamos los atributos en sub-atributos, podemos cubrir más detalles en la estimación.

El modelo propuesto en el presente documento será útil para nuevos desarrolladores y para desarrolladores sin experiencia. Podemos confiar en que los resultados del modelo serán similares a los de un desarrollador profesional.

Capítulo 5

5. Conclusiones, limitaciones, aportaciones y trabajo futuro

5.1 Conclusiones

En este estudio, se propuso un método para estimar las historias de los usuarios a través de un modelo compuesto por atributos relacionados con los aspectos personales de los desarrolladores. Con este modelo, buscamos mejorar la precisión en la estimación de historias de usuarios. Tomamos como base nuestro artículo anterior (Durán et al. 2020) del cual analizamos los atributos para formar nuestro modelo propuesto. Debido a que nuestro modelo se enfoca en aspectos personales, analizamos los atributos que estaban dentro de esta categoría y seleccionamos los más relevantes: experiencia con tecnología, experiencia con lenguaje, experiencia en proyectos anteriores, habilidades de desarrollador y conocimiento del tema del proyecto. Decidimos agregar algunos atributos importantes relacionados con la complejidad técnica de las historias de los usuarios: dependencia y tamaño. Notamos que el atributo de tamaño podría dividirse en 2 atributos más: funciones de conexión y funciones de datos. Dando con esto un modelo centrado en los aspectos personales de los desarrolladores, considerando también algunos aspectos técnicos importantes.

Como técnica para representar el modelo, elegimos RBs, ya que nos permiten describir relaciones probabilísticas entre variables relacionadas. Para la construcción de la RB, fue necesario obtener los valores entre las relaciones de cada atributo, para obtener estos datos consultamos a profesionales a través de encuestas. Con los valores obtenidos y la aplicación de fórmulas, obtuvimos la TPC para cada nodo secundario.

Para probar nuestro modelo, utilizamos dos tipos de muestras, estudiantes y profesionales. El primer experimento se aplicó a los estudiantes y consistió en hacer 3 estimaciones, la primera de manera tradicional con PP antes de desarrollar la historia del usuario, la segunda estimación utilizando la RB antes de desarrollar la historia del usuario y la tercera estimación de manera

tradicional una vez que se realizó la HU. En el caso de los profesionales, el experimento consistió en seleccionar 10 HU definidas por los estudiantes; estas fueron estimadas por profesionales, de manera tradicional, y con la RB.

Para validar los resultados, aplicamos pruebas de correlación de Spearman a las estimaciones obtenidas. Para los estudiantes, obtuvimos una correlación de 0.659 entre la estimación inicial y la estimación obtenida a través de la red, y una correlación de 0.799 entre la estimación final y la estimación obtenida a través de la red. Estos resultados nos muestran que existe una mayor correlación entre la estimación final y la estimación obtenida a través de la red, lo que indica que los datos obtenidos a través de la red son más similares a lo que el estudiante estimó una vez que se desarrolló la historia del usuario. Para los profesionales, obtuvimos una correlación de 0.887 entre la estimación inicial y la estimación obtenida a través de la red. Podemos notar que la correlación de los profesionales es superior a la de los estudiantes, esto se debe a que los profesionales tienen más experiencia. A través de los resultados, también podemos destacar que nuestro modelo estima alrededor del 88% como lo haría un desarrollador profesional, lo cual es un resultado favorable y confiable.

Si comparamos nuestros resultados con algunos trabajos relacionados como Martínez (López-Martínez et al. 2017) quien utilizó la misma metodología que se presentó en este estudio y Alostad (Alostad, Abdullah, and Aali 2017), quien incluso utilizó una metodología diferente pero con el mismo enfoque que nuestra propuesta, podemos ver que nuestro modelo es más preciso, esto se debe al hecho de que nuestro modelo se compone principalmente de atributos personales, que tienen un gran impacto en el proceso de la estimación. También porque en nuestro trabajo desglosamos algunos atributos en sub-atributos, lo que nos permitió obtener más detalles e información de la HU y del desarrollador en el momento de la estimación.

En nuestro modelo contemplamos la experiencia, pero no como un atributo único. Al desglosar un atributo tan importante como la experiencia, pudimos aumentar la precisión en nuestras estimaciones, en este caso, pudimos obtener más información relacionada con la experiencia del desarrollador. Esto demuestra que, si se desglosan los atributos correctos, podemos lograr mejores estimaciones. Los resultados obtenidos a través de la red nos brindan seguridad, ya que son similares a los de un desarrollador profesional; Esto se logra porque el peso de cada

atributo dentro de la red representa la opinión de los expertos, lo que ayuda a gestionar la incertidumbre e inexperiencia de los desarrolladores y mejorar la precisión de la estimación.

Trabajar con este método nos dará confianza y seguridad en nuestras estimaciones. Este método puede ser utilizado en cualquier equipo de desarrollo sin importar su experiencia, incluso puede funcionar en proyectos no ágiles, pues al ser un método que consiste en contestar preguntas en base a un requerimiento de software, se puede adaptar a cualquier metodología de desarrollo. Sin embargo, este método ayudará especialmente si estamos trabajando con desarrolladores sin experiencia o con poca experiencia, o en equipos sin datos históricos.

5.2 Limitaciones

A continuación, se presentan algunas limitantes que surgieron en esta investigación:

- Contemplamos en nuestro modelo solo atributos cualitativos lo que conlleva subjetividades.
- No consideramos atributos relacionados a la evaluación de la HU.
- Se logró superar el porcentaje de precisión de otros modelos, sin embargo, el aumento fue muy poco.

5.3 Aportaciones

A continuación, se presentan los aspectos que podemos resaltar de esta investigación:

- Método para estimar historias de usuario que puede ser utilizado por desarrolladores con y sin experiencia o sin datos históricos.
- Aumento al porcentaje de precisión en comparación con otros métodos utilizados.
- Revisión de la literatura sobre los atributos que influyen en la estimación de la complejidad en las historias de usuario.

5.4 Trabajo futuro

Como trabajo futuro, se planea extender el modelo siguiendo la idea principal de un modelo centrado en los atributos personales tomando como enfoque el uso de atributos cuantitativos con el fin de incrementar la precisión y disminuir incertidumbre en las estimaciones. Implementación de este método de estimación en más equipos para tener una muestra más grande en los experimentos a fin de obtener datos más confiables. También se plantea probar el método en proyectos de software no ágiles.

Bibliografía

- Abdou, Saed M, Kuan Yong, and Mohammed Othman. 2016. "Project Complexity Influence on Project Management Performance – The Malaysian Perspective." *MATEC Web of Conferences* 66: 00065.
- Alostad, Jasem M, Laila R A Abdullah, and Lamya Sulaiman Aali. 2017. "A Fuzzy Based Model for Effort Estimation in Scrum Projects." *International Journal of Advanced Computer Science and Applications* 8(9): 270–77. www.ijacsa.thesai.org.
- Bass, Julian M. 2015. "How Product Owner Teams Scale Agile Methods to Large Distributed Enterprises." : 1525–57.
- Bosch-Rekveltdt, Marian, Hans Bakker, and Marcel Hertogh. 2018. "Comparing Project Complexity across Different Industry Sectors." *Complexity* 2018.
- Celina, Heidi;, and Adalberto Campo. 2016. "Aproximación Al Uso Del Coeficiente Alfa de Cronbach." *Revista Colombiana de Psiquiatría* 45(1): 19–25. <http://www.redalyc.org/pdf/806/80634409.pdf><http://www.redalyc.org/pdf/806/80650839004.pdf>.
- Césari, Matilde Inés. 2006. "Trabajo Final de Especialidad En Tecnologías de Explotación de Información DE UNA RED BAYESIANA." <http://laboratorios.fi.uba.ar/lsi/rgm/tesistas/cesari-trabajofinaldeespecialidad.pdf>.
- Cohn, M. 2004. *User Stories Applied: For Agile Software Development*. Pearson Education. https://books.google.com.mx/books?id=DHZZP_YL3FxyC.
- Cristóbal, J. R.San. 2017. "Complexity in Project Management." *Procedia Computer Science* 121: 762–66. <https://doi.org/10.1016/j.procs.2017.11.098>.
- DOLDÁN TIÉ, FÉLIX. 2007. "REDES BAYESIANAS Y RIESGO OPERACIONAL." *Revista Galega de Economía* 16.
- Dragicevic, Srdjana, Stipe Celar, and Mili Turic. 2017. "Bayesian Network Model for Task Effort

- Estimation in Agile Software Development.” *Journal of Systems and Software* 127: 109–19.
<http://dx.doi.org/10.1016/j.jss.2017.01.027>.
- Drahý, Cyril, and Otto Pastor. 2016. “Relationship Between Project Complexity and Risk Kinds Identified on Projects.” *EMI - Economics Management Innovation* 8(1): 42–50.
http://emijournal.cz/wp-content/uploads/2016/05/04_Relationship-Between-Project-Complexity-and-Types-of-Risk-Identified-in-the-Projects.pdf.
- Durán, Mayra, Reyes Juárez-ramírez, Samantha Jiménez, and Claudia Tona. 2020. “Taxonomy for Complexity Estimation in Agile Methodologies : A Systematic Literature Review.” : 1–10.
- Edici, Tercera. *Cuerpo de Conocimiento de Scrum™ (Guía Sbok)*.
- Felgaer, Pablo. 2004. “Optimización de Redes Bayesianas Basado En Técnicas de Aprendizaje Por Inducción.” *Reportes Técnicos en Ingeniería del Software* 6(2): 64–69.
<http://laboratorios.fi.uba.ar/lsi/felgaer-tesisingenieriainformatica.pdf>.
- Georgsson, Anna. 2011. “Introducing Story Points and User Stories to Perform Estimations in a Software Development Organisation - A Case Study at Swedbank IT.”
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.4909&rep=rep1&type=pdf%5Cnhttp://www.diva-portal.org/smash/record.jsf?pid=diva2:400313>.
- Gutiérrez, Allan Balam Rueda. 2010. “Propuesta de Una Guía Para Interpretar Los Procesos de Moprosoft de La Categoría de Operación Usando Una Combinación de Métodos Ágiles.” : 217.
- Hsu, Shu Chien, Kai Wei Weng, Qingbin Cui, and William Rand. 2016. “Understanding the Complexity of Project Team Member Selection through Agent-Based Modeling.” *International Journal of Project Management* 34(1): 82–93.
<http://dx.doi.org/10.1016/j.ijproman.2015.10.001>.
- Jeff Sutherland, Neil Harrison, and Joel Riddle. 2014. “Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams.” In *2014 47th Hawaii International Conference on System Sciences*,.

- Jennings, N R, and M Wooldridge. 2000. "Agent-Oriented Software Engineering." *Artificial Intelligence* 117: 277–96.
- Jørgensen, M. 2012. "Does the Use of Fibonacci Numbers in Planning Poker Affect Effort Estimates?" *IET Conference Proceedings*: 228-232(4). <http://digital-library.theiet.org/content/conferences/10.1049/ic.2012.0030>.
- Joyce, James. 2019. "Bayes' Theorem." *The Stanford Encyclopedia of Philosophy (Spring 2019 Edition)*.
- Karna, H., and S. Gotovac. 2015. "Estimating Software Development Effort Using Bayesian Networks." *23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*: 229–33. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7314091.
- Kayes, Imrul, Mithun Sarker, and Jacob Chakareski. 2016. "Product Backlog Rating: A Case Study on Measuring Test Quality in Scrum." *Innovations in Systems and Software Engineering* 12(4): 303–17.
- Kitchenham, Barbara, David Budgen, and Pearl Brereton. 2018. "Evidence-Based Software Engineering and Systematic Reviews." *Evidence-Based Software Engineering and Systematic Reviews*.
- Lenarduzzi, Valentina. 2015. "Could Social Factors Influence the Effort Software Estimation?" *Proceedings of the 7th International Workshop on Social Software Engineering - SSE 2015*: 21–24. <http://dl.acm.org/citation.cfm?doid=2804381.2804385>.
- Linåker, Johan; Sulaman, Sardar Muhammad; Maiani de Mello, Rafael; Höst, Martin. 2015. "Guidelines for Conducting Surveys in Software Engineering v. 1.1." : 0–63.
- Lopez-Martinez, Janeth et al. 2018. "Analysis of Planning Poker Factors between University and Enterprise." *Proceedings - 2017 5th International Conference in Software Engineering Research and Innovation, CONISOFT 2017* 2018-Janua: 54–60.
- López-Martínez, Janeth et al. 2017. "User Stories Complexity Estimation Using Bayesian

- Networks for Inexperienced Developers.” *Cluster Computing*: 1–14.
- Lopez-Martinez, Janeth. 2017. “Estimating User Stories Complexity and Importance.” *Advances in Intelligent Systems and Computing* 571. <http://link.springer.com/10.1007/978-3-319-56541-5>.
- Lucassen, Garm, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf, and Sjaak Brinkkemper. 2016. “Improving Agile Requirements: The Quality User Story Framework and Tool.” *Requirements Engineering* 21(3): 383–403.
- M., Abdou Saed, Kuan Yong, and Mohamed Othman. 2018. “Categorization of Project Complexity Factors In Malaysian Construction Industry.” *International Journal of Property Sciences* 7(1): 1–15.
- Mahnič, Viljan, and Tomaž Hovelja. 2012. “On Using Planning Poker for Estimating User Stories.” *Journal of Systems and Software* 85(9): 2086–95.
- Malgonde, Onkar, and Kaushal Chari. 2018. Empirical Software Engineering *An Ensemble-Based Model for Predicting Agile Software Development Effort*. Empirical Software Engineering.
- Martinez, J. 2017. “Estimación de Historias de Usuario En Scrum a Través de Un Modelo Basado En Descomposición de La Complejidad.” : 87.
- Mendes, Emilia. 2011. “Knowledge Representation Using Bayesian Networks A Case Study in Web Effort Estimation.” : 612–17.
- Moreno, Ana M., and Agustín Yagüe. 2012. “Agile User Stories Enriched with Usability.” *Lecture Notes in Business Information Processing* 111 LNBIP: 168–76.
- Normandi Rocío Tirado Ríos, Freddy Enrique Triana Litardo, Jorge Wilson Saa Saltos. 2016. “Optimización de Redes Bayesianas Basado En Técnicas de Aprendizaje Por Inducción.” 3(9): 41–60.
- Pandit, Pallavi, and Swati Tahiliani. 2015. “AgileUAT: A Framework for User Acceptance Testing Based on User Stories and Acceptance Criteria.” *International Journal of Computer*

Applications 120(10): 16–21.

Pauly, Daniel, Bjoern Michalik, and Dirk Basten. 2015. “Do Daily Scrums Have to Take Place Each Day? A Case Study of Customized Scrum Principles at an E-Commerce Company.” *Proceedings of the Annual Hawaii International Conference on System Sciences* 2015-March: 5074–83.

Peischl, Bernhard, Markus Zanker, Mihai Nica, and Wolfgang Schmid. 2010. “Constraint-Based Recommendation for Software Project Effort Estimation.” *Journal of Emerging Technologies in Web Intelligence* 2(4): 282–90.

Popli, R, and N Chauhan. 2013. “A Sprint-Point Based Estimation Technique in Scrum.” In *2013 International Conference on Information Systems and Computer Networks*, , 98–103.

———. 2014. “Cost and Effort Estimation in Agile Software Development.” *Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on*: 57–61. <http://www.scopus.com/inward/record.url?eid=2-s2.0-84900313137&partnerID=40&md5=56e260b4c9414344f28884432815991a>.

Porru, Simone, Alessandro Murgia, Serge Demeyer, and Michele Marchesi. 2020. “Estimating Story Points from Issue Reports.”

Pressman, Roger S. *Ingeniería Del Software. Un Enfoque Práctico*.

Qazi, Abroon, John Quigley, Alex Dickson, and Konstantinos Kirytopoulos. 2016. “Project Complexity and Risk Management (ProCRiM): Towards Modelling Project Complexity Driven Risk Paths in Construction Projects.” *International Journal of Project Management* 34(7): 1183–98.

Raith, Florian, Ingo Richter, Robert Lindermeier, and Gudrun Klinker. 2013. “Identification of Inaccurate Effort Estimates in Agile Software Development.” *Proceedings - Asia-Pacific Software Engineering Conference, APSEC 2*: 67–72.

Ramirez-Noriega, Alan, Reyes Juarez-Ramirez, Raul Navarro, and Janeth Lopez-Martinez. 2016. “Using Bayesian Networks to Obtain the Task’s Parameters for Schedule Planning in Scrum.”

Proceedings - 2016 4th International Conference in Software Engineering Research and Innovation, CONISOFT 2016 (1): 167–74.

Rath, Santanu Kumar, Barada Prasanna Acharya, and Shashank Mouli Satapathy. 2016. “Early Stage Software Effort Estimation Using Random Forest Technique Based on Use Case Points.” *IET Software* 10(1): 10–17. <http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2014.0122>.

Raza, Ali et al. 2017. “Impact of Story Point Estimation on Product Using Metrics in Scrum Development Process.” *International Journal of Advanced Computer Science and Applications* 8(4). <http://thesai.org/Publications/ViewPaper?Volume=8&Issue=4&Code=ijacsa&SerialNo=52>.

Rincón, Luis. 2014. “Introducción a La Probabilidad.” : 530. <http://lya.ciencias.unam.mx/lars/Publicaciones/Prob1-2016.pdf>.

Rodríguez, D, and Javier Dolado. 2007. “Redes Bayesianas En La Ingeniería Del Software.” *Cc.Uah.Es*: 1–21. <http://www.cc.uah.es/drg/b/RodriguezDolado.BBN.2007.pdf>.

RUBIN, K. S. 2013. “ESSENTIAL SCRUM Framework, A Practical Guide to the Most Popular Agile Process.”

Schwaber, Ken, and Jeff Sutherland. 2013. “La Guía de Scrum.” *Scrumguides.Org* 1: 21. <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>.

———. 2016. “La Guía de Scrum.” *Scrum.org*: 5. <http://creativecommons.org/licenses/by-sa/4.0/legalcode%0Ahttp://creativecommons.org/licenses/by-sa/4.0/legalcode>.

Scott, E., and D. Pfahl. 2018. “Using Developers’ Features to Estimate Story Points.” *ACM International Conference Proceeding Series* (106): 106–10.

SCRUMstudy. 2016. *A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ Guide)*. USA: VMEdU, Inc.

Sholiq, Renny Sari Dewi, and Apol Pribadi Subriadi. 2017. “A Comparative Study of Software

Development Size Estimation Method: UCPabc vs Function Points.” *Procedia Computer Science* 124: 470–77. <https://doi.org/10.1016/j.procs.2017.12.179>.

Singh, R.K. 2016. “A MCDM Approach for Measuring Project Complexity.” (1): 294–302.

Srivastava, Apoorva, India ASET, Amity University, Uttar Pradesh, Noida Campus, and ; Sukriti Bhardwaj ; Shipra Saraswat. 2017. “SCRUM Model for Agile Methodology.” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*.

Sucar, Luis Enrique. 2011. “Introduction to Bayesian Networks and Influence Diagrams.” *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*: 9–32.

Tanveer, Binish, Liliana Guzmán, and Ulf Martin Engel. 2017. “Effort Estimation in Agile Software Development: Case Study and Improvement Framework.” *Journal of Software: Evolution and Process* 29(11): 1–14.

Usman, Muhammad, Jürgen Börstler, and Kai Petersen. 2017. “An Effort Estimation Taxonomy for Agile Software Development.” *International Journal of Software Engineering and Knowledge Engineering* 27(04): 641–74. <http://www.worldscientific.com/doi/abs/10.1142/S0218194017500243>.

Zahraoui, Hind, and Mohammed Abdou Janati Idrissi. 2015. “Adjusting Story Points Calculation in Scrum Effort & Time Estimation.” In *2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015*,.

Zare, Fatemeh, Hasan Khademi Zare, and Mohammad Saber Fallahnezhad. 2016. “Software Effort Estimation Based on the Optimal Bayesian Belief Network.” *Applied Soft Computing Journal* 49: 968–80. <http://dx.doi.org/10.1016/j.asoc.2016.08.004>.

Ziauddin, Shahid Kamal Tipu, Shahrukh Zia. 2012. “An Effort Estimation Model for Agile Software Development.” *Advances in Computer Science and its Applications (ACSA)* 2(1): 2166–2924. <http://www.worldscientific.com/doi/abs/10.1142/S0218194017500243>.

Apéndices

Apéndice A Encuestas

A.1 Encuesta para validar atributos

- ¿Qué tan de acuerdo está usted con que la *experiencia con la tecnología* es un atributo importante para determinar la *experiencia del desarrollador*?

Experiencia con la tecnología: El desarrollador domina las herramientas, métodos y tecnologías para el desarrollo de la HU de forma efectiva.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que la *experiencia con el lenguaje de programación* es un atributo importante para determinar la *experiencia del desarrollador*?

Experiencia con el lenguaje: El desarrollador domina y es capaz de usar en forma efectiva los recursos de dicho lenguaje, y produce un resultado organizado y valioso. El desarrollador conoce las limitaciones de dicho lenguaje.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que la *experiencia en proyectos previos* es un atributo importante para determinar la *experiencia del desarrollador*?

Experiencia en proyectos previos: conocimiento que un desarrollador posee en base a proyectos similares o anteriores.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que la **experiencia** del desarrollador es un atributo importante para determinar la **complejidad de una HU**?

Experiencia: es la acumulación de conocimientos y habilidades adquiridos.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que las **habilidades del desarrollador** son un atributo importante para determinar la **complejidad de una HU**?

Habilidades del desarrollador: Conocimientos y capacidades que posee un desarrollador (capacidad de análisis, creatividad, etc.).

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que el **conocimiento del tema del proyecto** es un atributo importante para determinar la **complejidad de una HU**?

Conocimiento del tema del proyecto: El desarrollador conoce y domina la naturaleza y los conceptos más importantes del tema del que se realizara el proyecto.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que la **dependencia** entre una HU y otra es un atributo importante para determinar la **complejidad técnica de una HU**?

Dependencia: que tan importante es una HU debido a que otras Historias de usuario dependen de esta.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que las **funciones de conexión** de una HU es un atributo importante para determinar el **tamaño** de una HU?

Funciones de conexión: son los requerimientos de procesamiento del usuario como entradas, salidas y consultas.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que las **funciones de datos** de una HU es un atributo importante para determinar el **tamaño** de una HU?

Funciones de datos: son los requerimientos de almacenamiento del usuario, como archivos lógicos internos y archivos de interfaz externa.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que el **tamaño** de una HU es un atributo importante para determinar la **complejidad técnica** de una HU?

Tamaño: es la dimensión que tiene la HU en relación a la cantidad de trabajo que requiere considerando los aspectos técnicos que se requieren para desarrollarla.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

- ¿Qué tan de acuerdo está usted con que la **complejidad técnica** es un atributo importante para determinar la **complejidad de una HU**?

Complejidad técnica: es una cualidad de la HU que está compuesta de diversos elementos técnicos.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

A.2 Encuesta para estimar a través de la red

- ¿Cuánta *experiencia tienes con la tecnología* a utilizar en la creación de la HU?

Poco	Nunca he utilizado la tecnología con la que se desarrollara la HU.
Regular	Conozco, pero aún no domino la tecnología con la que se desarrollara la HU.
Mucho	Domino la tecnología con la que se desarrollara la HU.

- ¿Cuánta *experiencia tienes con el lenguaje de programación* a utilizar en la creación de la HU?

Poco	Nunca he utilizado el lenguaje con el que se desarrollara la HU.
Regular	Conozco, pero aún no domino el lenguaje con el que se desarrollara la HU.
Mucho	Domino el lenguaje con el que se desarrollara la HU.

- ¿Cuánta *experiencia tienes en proyectos previos* desarrollando historias de usuario similares?

Poco	Nunca he desarrollado una HU similar.
Regular	He desarrollado una HU similar alguna vez.
Mucho	He desarrollado diversas historias de usuario similares.

- ¿Cómo consideras que son tus *habilidades como desarrollador* para llevar a cabo la HU?

Deficiente	No logro identificar mis habilidades y me cuesta trabajo solucionar problemas.
Suficiente	Soy consciente de las habilidades que tengo y en ocasiones las utilizo para resolver problemas.
Sobresaliente	Conozco mis habilidades y se cuándo aplicarlas a la solución de problemas.

- ¿Cuánto es tu *conocimiento acerca del tema del proyecto* para desarrollar la HU?

Poco	Desconozco el tema del que se desarrollara la HU
Regular	Conozco los conceptos básicos del tema del que se desarrollara la HU.
Mucho	Domino el tema del que se desarrollara la HU.

- ¿Qué tan *dependiente* es la HU a desarrollarse con respecto a otras?

Independiente	La HU es independiente a otras historias de usuario.
Dependiente	La HU depende un poco de otras historias de usuario
Muy dependiente	La HU es la base para el desarrollo de más de varias historias de usuario

- ¿Cuánta dificultad tienen las *funciones de conexión* para el desarrollo de la HU?

Poca	La HU contiene funciones sencillas de entrada, salida y consultas
Media	La HU contiene funciones de complejidad mediana de entrada, salida y consultas
Alta	La HU contiene funciones complejas de entrada, salida y consultas

- ¿Cuánta dificultad tienen las *funciones de datos* para el desarrollo de la HU?

Sencillo	La HU contiene funciones sencillas de lógica interna y externa.
Regular	La HU contiene funciones de complejidad mediana de de lógica interna y externa.
Complicado	La HU contiene funciones complejas de de lógica interna y externa.

Apéndice B Historias de usuario

B.1 Historias de usuario

A continuación, se enlistan algunas historias de usuario que fueron estimadas por los estudiantes y profesionales.

ID	Descripción
US1	Como usuario quiero ir a la página de inicio del congreso para ver la información general del congreso.
US2	Como usuario quiero ir a ver la información de las conferencias que estarán en el congreso.
US3	Como usuario quiero ver la información de los alojamientos para conocer la ubicación de los hoteles dónde se realizará el congreso y qué puedo hacer en la ciudad donde se llevará a cabo.
US4	Como usuario quiero ver qué personas están involucradas en el congreso para saber a quién contactar.
US5	Como usuario quiero registrarme en la página del congreso para poder acceder al sistema.
US6	Como usuario quiero ver la información general del congreso para estar informado
US7	Como usuario quiero iniciar sesión con mi correo electrónico y contraseña para poder acceder a mi cuenta del sistema.
US8	Como usuario quiero recuperar mi contraseña para poder recuperar el acceso a mi cuenta del sistema.
US9	Como administrador quiero tener una página solo para administradores, de modo que pueda acceder a toda la información del congreso.
US10	Como administrador quiero crear, editar o eliminar la información de los talleres para poder actualizar esta información en la base de datos del sistema del congreso.

Apéndice C Publicaciones

C.1 Publicaciones en JCR

- User Story Estimation based on the Complexity Decomposition using Bayesian Networks.
Autores: Mayra Durán, J. Reyes Juárez Ramírez, Samantha Jiménez, Claudia Tona.
Journal (JRC): Programming and Computer Software. Factor de impacto: 0.75, Q4
(Scopus: Q3)

C.2 Publicaciones en conferencia

- A Taxonomy for Complexity Estimation in Agile Methodologies: A Systematic Literature Review.
Autores: Mayra Durán, J. Reyes Juárez Ramírez, Samantha Jiménez, Claudia Tona.
Conferencia: 7th International Conference in Software Engineering Research and Innovation CONISOFT 2019
- Towards a Set of Factors to Identify the Success in Scrum Project Delivery: A Systematic Literature Review.
Autores: Claudia Tona, J. Reyes Juárez Ramírez, Samantha Jiménez, Mayra Durán.
Conferencia: 7th International Conference in Software Engineering Research and Innovation CONISOFT 2019