

Mejoramiento del proceso de gestión de proyectos de desarrollo de software mediante el diseño y desarrollo de una herramienta en versión alfa en la empresa SCITECH S.A.S. que use simulaciones Montecarlo

Erik Maurieth Sánchez Vargas

Trabajo de grado para optar al título de Ingeniero Industrial

Director

Néstor Fabian Santos Nova

Maestro en Administración de Negocios

Universidad Industrial de Santander

Facultad de Ingenierías Físico-mecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2021

### **Dedicatoria**

*A mi mamá por ser el motor de mi vida y ser todos los días un ejemplo de amor y sacrificio.*

*A mi papá por motivarme cada día a ser mejor y apoyarme desde siempre.*

*A mi tía Martha por ser una segunda mamá para mí.*

### **Agradecimientos**

A mi familia que me apoyó y me brindó la mano siempre en los momentos que lo necesité.  
A mi Rayito de Sol por iluminar mi vida con su amor, cariño, sacrificio y apoyo. No sé qué haría  
sin ti.

A mis mejores amigos de Empaguestas que gracias a ellos no hay día que me sienta mal y no me  
saquen una sonrisa.

A una de mis mejores amigas Natis que amo con todo mi corazón, con la que empecé este  
camino y siempre ha sido mi referente como persona.

A una de mis mejores amigas Diana porque siempre ha sido la voz de la razón y me ha brindado  
apoyo incondicional a pesar de los problemas.

A mi mejor amigo Daniel y Krishan que me acogieron en el colegio cuando no tenía a nadie y  
han sido desde entonces mi apoyo incondicional.

A la Pastoral Universitaria UIS porque desde que los conocí se convirtieron en mi familia y mi  
segundo hogar.

A mi director Néstor Santos el cual admiro muchísimo y es un referente para mí como  
profesional, docente y persona. Nada de esto sería posible sin todo lo que me enseñó y ayudó.

A Science & Technology por enseñarme muchísimo y abrirme las puertas, especialmente al Ing.  
Carlos por brindarme la confianza para dirigir un proyecto y convertir de la empresa una escuela  
de proyectos para mí.

A mis profesores de la Universidad Industrial de Santander que hicieron posible todo esto, cuya  
labor admiro y espero ejercer algún día.

## Tabla de contenido

Introducción .....	13
1.Generalidades del proyecto el proyecto .....	16
1.1.Objetivos .....	16
1.1.1.General .....	16
1.1.2.Objetivos Específicos: .....	16
1.2.Planteamiento del problema.....	17
1.3.Metodología del proyecto .....	18
1.3.1.Reconocimiento y diagnóstico de la técnica actual de estimación de costos y tiempos.....	19
1.3.2.Diseño del plan de mejora basado en la metodología de clasificación y asignación de tareas e implementación del mismo .....	19
1.3.3.Determinación de los costos por hora de los involucrados en un proyecto con base a las variaciones identificadas.....	19
1.3.4.Diseño y desarrollo de la herramienta en Microsoft VBA en Excel.....	20
2.Generalidades de la empresa.....	20
2.1.Descripción de la empresa .....	20
2.1.1.Información general de la empresa .....	21
2.1.2.Misión. ....	21
2.1.3.Visión.....	21
2.1.4.Organigrama .....	22
2.1.5.Mapa de procesos.....	23
3.Marco de referencia .....	23
3.1.Marco de antecedentes .....	23
3.2.Marco conceptual.....	25
3.3.Marco teórico .....	26
3.3.1.Gestión de proyectos de software .....	26
3.3.2.Modelos y técnicas de estimación de tiempos y costos de desarrollo de software.....	34
3.3.3.Diseño y desarrollo de herramientas de software .....	42
3.3.4.Técnicas de simulación .....	45
3.3.5.Mejoramiento de procesos .....	47

4.Diagnóstico inicial .....	47
4.1.Reconocimiento de los procesos .....	47
4.1.1.Dirección de proyectos de desarrollo de software. ....	47
4.1.2.Desarrollo de software y QA. ....	49
4.1.3.Proceso de estimación de tiempos .....	49
4.1.4.Proceso de estimación de costos .....	50
4.1.5.Proceso de asignación de tareas de desarrollo .....	53
4.1.6.Análisis del proceso de estimación de tiempos y costos actual .....	53
4.1.7.Análisis del proceso de asignación de tareas de desarrollo .....	57
5.Plan de mejora.....	58
5.1.Mejora proceso de estimación de tiempos y costos .....	58
5.1.1.Problema a solucionar.....	58
5.1.2.Propuesta de mejora para el proceso de estimación de tiempos y costos .....	58
5.1.3.Plan de implementación de la mejora propuesta.....	59
5.1.4.Implementación del plan de mejora para el proceso de estimación de tiempos y costos ....	60
5.2.Plan de mejora para el proceso de asignación de tareas .....	97
5.2.1.Problema a solucionar.....	97
5.2.2.Propuesta de mejora para el proceso de estimación de asignación de tareas.....	97
5.2.3.Plan de implementación de la mejora propuesta.....	98
5.2.4.Implementación del plan de mejora para el proceso de asignación de tareas.....	99
6.Conclusiones .....	107
7.Recomendaciones .....	108
Referencias.....	110

### Lista de tablas

<b>Tabla 1.</b> Cumplimiento de objetivos .....	15
<b>Tabla 2.</b> Información general de la empresa .....	21
<b>Tabla 3.</b> Comparación metodologías tradicionales vs ágiles .....	28
<b>Tabla 4.</b> Peso de los actores .....	38
<b>Tabla 5.</b> Peso de los casos de uso.....	38
<b>Tabla 6.</b> Factores técnicos .....	39
<b>Tabla 7.</b> Escalas de estimación TCF .....	40
<b>Tabla 8.</b> Factores ambientales .....	40
<b>Tabla 9.</b> Escalas de estimación de EF .....	41
<b>Tabla 10.</b> % de distribución de tiempo .....	50
<b>Tabla 11.</b> Definición del presupuesto .....	51
<b>Tabla 12.</b> Tiempos ajustados.....	56
<b>Tabla 13.</b> Costos ajustados.....	57
<b>Tabla 14.</b> Plan de implementación para la mejora propuesta para estimación de tiempos y costos .....	59
<b>Tabla 15.</b> Matriz de evaluación de técnicas de estimación .....	61
<b>Tabla 16.</b> Matriz de definición de criterios mínimos de un requerimiento .....	62
<b>Tabla 17.</b> Estimación de complejidad de las tareas .....	64
<b>Tabla 18.</b> Instrucciones para los puntajes de cada tarea .....	65
<b>Tabla 19.</b> Clasificación de las complejidades de las tareas.....	66
<b>Tabla 20.</b> Clasificación de requerimientos antiguos .....	73
<b>Tabla 21.</b> Diseño de datos de la tabla Requerimientos .....	84
<b>Tabla 22</b> Diseño de datos de la tabla Movimientos .....	85
<b>Tabla 23.</b> Diseño de datos de la tabla de Roles.....	86
<b>Tabla 24.</b> Diseño de datos de la tabla UsuariosJIRA.....	86
<b>Tabla 25.</b> Diseño de datos de la tabla Proyectos.....	87
<b>Tabla 26.</b> Diseño de datos de la tabla Complejidades.....	87
<b>Tabla 27.</b> Diseño de datos de la tabla AsignacionesSemanales .....	88
<b>Tabla 28.</b> Diseño de datos de la tabla de DetalleAsignacionSemanal .....	89

<b>Tabla 29.</b> Tabla de costos unitarios por hora por rol .....	91
<b>Tabla 30.</b> Plan de implementación para la mejora propuesta para la asignación de tareas .....	98
<b>Tabla 31.</b> Matriz de calificación desempeño .....	99
<b>Tabla 32.</b> Puntajes para el criterio de año de experiencia con Genexus .....	100
<b>Tabla 33.</b> Puntaje para el criterio de años de experiencia con SQL.....	100
<b>Tabla 34.</b> Puntaje para el criterio de cantidad de lenguajes .....	100
<b>Tabla 35.</b> Clasificaciones de los desarrolladores .....	101

### Lista de figuras

Figura 1.. Triple restricción en los proyectos.....	17
Figura 2. Organigrama actual de la empresa.....	22
Figura 3. Mapa de procesos .....	23
Figura 4. Diagrama del proceso de desarrollo de software.....	48
Figura 5. Regresión lineal de horas empleadas vs horas estimadas por desarrollador 24 .....	54
Figura 6. Regresión lineal de horas empleadas vs horas estimadas por desarrollador 25 .....	55
Figura 7. Regresión lineal de horas empleadas vs horas estimadas por analista 15 .....	55
Figura 8. Backlog de un proyecto de desarrollo de software en JIRA .....	68
Figura 9. Dashboard de un requerimiento.....	69
Figura 10. Lista de tareas de un requerimiento .....	69
Figura 11. Dashboard de una tarea .....	70
Figura 12. Proceso iterativo que se realiza para procesar la información .....	71
Figura 13. Función creada en Python para hacer consultas a la API REST .....	72
Figura 14. Análisis de la distribución de los datos de complejidad baja .....	74
Figura 15. Simulación con distribución gamma de los datos de complejidad baja.....	74
Figura 16. Análisis de la distribución de los datos de complejidad media .....	76
Figura 17. Simulación con distribución normal de los datos de complejidad media .....	77
Figura 18. Análisis de la distribución de los datos de complejidad alta .....	78
Figura 19. Simulación con distribución Gamma de los datos de complejidad alta.....	79
Figura 20. Análisis de la distribución de los datos de complejidad muy alta .....	80
Figura 21. Simulación con distribución Gamma de los datos de complejidad muy alta.....	81
Figura 22. Diagrama de entidad relación.....	82
Figura 23. Interfaz principal.....	90
Figura 24. Verificación de tareas a iterar.....	92
Figura 25. Tabla de simulación de las tareas .....	93
Figura 26. Hoja de informe de simulación del tiempo total estimado.....	93
Figura 27. Reporte de simulación del tiempo total.....	94
Figura 28. Hoja de informe de simulación del costo total estimado .....	95
Figura 29. Informe de simulación del costo total .....	96

Figura 30. Nuevo campo en las tareas de desarrollo .....	97
Figura 31. Botón Ver última asignación .....	103
Figura 32. Botón Nueva asignación .....	103
Figura 33. Botón Modificar asignación .....	104
Figura 34. Botón Automatizar asignación .....	105
Figura 35 Tablero de gestión.....	106

## **Lista de apéndices**

**Ver apéndices adjuntos y pueden ser consultados en la base de datos de la Biblioteca UIS**

Apéndice A. Código fuente código Python

Apéndice B. Script SQL para la creación de las tablas

Apéndice C. Procedimientos almacenados

Apéndice D. Herramienta de Excel

## Resumen

**Título:** Mejoramiento del proceso de gestión de proyectos de desarrollo de software mediante el diseño y desarrollo de una herramienta en versión alfa en la empresa SCITECH S.A.S. que use simulaciones Montecarlo. \*

**Autor:** Erik Maurieth Sánchez Vargas\*\*

**Palabras clave:** Simulación Montecarlo, Gestión de proyectos, Método estocástico, Reducción de costos, Programación.

**Descripción:** El presente documento es un proyecto de grado enfocado en el mejoramiento del proceso de gestión de proyectos de desarrollo de software mediante el diseño y desarrollo de una herramienta en versión alfa en la empresa SCITECH S.A.S que use simulaciones Montecarlo.

El proyecto se divide en 5 etapas que constan del diagnóstico, plan de mejora, identificación de costos por desarrollador y programación de la herramienta. Para esto, mediante la revisión de documentos internos, reuniones informales con diferentes líderes de proyecto y la líder del departamento de proyectos se realizó un diagnóstico inicial del estado actual de los procesos para posteriormente generar una propuesta de mejora a partir de los procesos identificados, donde se incluye, una mejora a la metodología de asignación de estimación de los requerimientos para los proyectos de desarrollo de software y otra mejora a la metodología de asignación de tareas de desarrollo de los proyectos en ejecución. Finalmente, todo converge en una herramienta programada en MS Excel a base de macros, que permite determinar mediante el método de Montecarlo los posibles costos y tiempos que se estiman en un proyecto además de escribir en una base de datos con el objetivo de permitir la creación de un tablero de gestión en Tableau para supervisar la operación completa día a día en tiempo real.

---

\* Trabajo de Pregrado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: MBA. Néstor Fabián Santos-Nova. Tutor: Carlos Alfonso Zafra Gerena

### Abstract

**Title:** Improvement of the software development project management process through the design and development of an alpha version tool in the company SCITECH S.A.S. using Monte Carlo simulations. \*

**Author:** Erik Maurieth Sánchez Vargas\*\*

**Keywords:** Montecarlo simulation, Project management, stochastic methods, cost reduction, programming.

**Description:** This document is a degree project focused on the improvement of the software development project management process through the design and development of an alpha version tool in the company SCITECH S.A.S that uses Montecarlo simulations.

The project is divided into 5 stages that consist of the diagnosis, improvement plan, identification of costs by developer and the design and develop of the tool. For this, by reviewing internal documents and processes, informal meetings with different project leaders, developers and the project department leader, an initial diagnosis of the current state of the processes was made to later generate an improvement proposal based on the identified processes, where it includes an improvement to the methodology for assigning the estimation of requirements for software development projects and another improvement to the methodology for assigning development tasks for projects in progress. Finally, everything converges in a tool developed in MS Excel based on macros, which allows to determine by means of the Montecarlo method the possible costs and times that are estimated in a project in addition to writing in a database with the aim of allowing the creation of a dashboard in Tableau to monitor the entire day-to-day operation in real time.

---

\* Degree work

\*\* Faculty of Physical-Mechanical Engineering. School of Industrial and Business Studies. Director: MBA. Néstor Fabián Santos-Nova. Tutor: Carlos Alfonso Zafra Gerena

## Introducción

Actualmente, las exigencias del entorno empresarial han ido aumentando con el pasar del tiempo, por lo tanto, la antigüedad o el posicionamiento de una compañía en un mercado no son garantías de que al siguiente año estas condiciones permanecerán igual. De tal manera, la toma rápida de decisiones acertadas se convierte en una capacidad que todos los líderes de las compañías deben desarrollar para estimar el futuro y pivotar si es necesario. Sin embargo, este trabajo sólo es posible de realizar correctamente mediante un análisis amplio e inmediato de la información que brindan los procesos que cada empresa ejecuta, por lo tanto, la tecnología juega un rol fundamental en esta tarea dado que permite automatizar el tratamiento de los datos, presentando información valiosa para tomar decisiones en segundos.

Science and Technology S.A.S. es una empresa que se dedica desde hace más de 19 años a la ejecución de proyectos de desarrollo de software y outsourcing tecnológico, siendo la primera su principal actividad comercial, de tal manera, la correcta gestión de proyectos se vuelve un elemento fundamental en el crecimiento económico de la compañía. Sin embargo, esta tarea de gestión se convierte en un elemento complejo cuando se llega a tener una cantidad considerable de proyectos en simultáneo. Adicionalmente, como afirma (Salazar, 2009) en la mayoría de empresas de desarrollo de software los procesos de estimación y planificación son muy débiles, donde generalmente, el administrador realiza el costeo y planificación de la duración al proyecto utilizando solamente juicio de experto, lo que genera cronogramas y presupuesto poco acertados, lo cual se ve agravado dado que las características de cada proyecto de desarrollo de software cambian de cliente en cliente, generando una dificultad en la estimación de la duración de tareas y, por lo tanto, en las estimaciones de costos.

Consecuentemente, la utilización de un modelo estocástico se vuelve necesario para realizar una mejora al proceso de estimación de tiempos, costos y asignaciones de tareas, ya que este permite tener en cuenta la variabilidad de la duración de las tareas, y con ello, el tiempo y costo pronosticado para cada proyecto. De esta forma, empresas de diferentes industrias que han aplicado este tipo de modelos han visto reducidos sus gastos y tiempos hasta un 30%, mejorando los flujos de caja y planeación de sus procesos.

De esta forma, el presente documento consiste en la descripción de la ejecución del proyecto de mejoramiento del proceso de gestión de proyectos de desarrollo de software mediante el diseño y desarrollo de una herramienta en versión alfa en la empresa SCITECH S.A.S. que use simulaciones Montecarlo teniendo en cuenta la variabilidad del tiempo empleado por cada desarrollador en los requerimientos de cada proyecto para determinar su duración total, los costos y adicionalmente, mejorar la asignación de tareas a los desarrolladores delegando cada tarea al más eficiente, reduciendo de esta manera costos y tiempo, mediante el método de Montecarlo. En primer lugar, se expone la problemática a solucionar mediante un diagnóstico de la situación actual. Después, se presentan los resultados esperados del proyecto. Luego, se plantea un marco de referencia que contiene un marco de antecedentes donde se revisan proyectos que hayan abordado la misma problemática, y un marco teórico que contiene información útil para la realización de este trabajo y finalmente, se detallan las etapas metodológicas del proyecto, la estructura y el desempeño de las actividades realizadas.

### Cumplimiento de objetivos

**Tabla 1.**

*Cumplimiento de objetivos*

<b>Objetivo</b>	<b>Cumplimiento</b>
<ul style="list-style-type: none"> <li>Realizar un diagnóstico del estado actual de los procesos de gestión de proyectos de la empresa SCITECH S.A.S. comparados con los estándares del PMBOK.</li> </ul>	Capítulo 4
<ul style="list-style-type: none"> <li>Diseñar un plan de mejora para la clasificación y asignación de tareas a los desarrolladores con el objetivo de mejorar el tiempo de ejecución y costo de la tarea según la dificultad de la misma.</li> </ul>	Capítulo 5
<ul style="list-style-type: none"> <li>Determinar los costos por hora de los involucrados en un proyecto de SCITECH S.A.S. para estimar el costo aproximado de un proyecto con base a las variaciones identificadas.</li> </ul>	Capítulo 5
<ul style="list-style-type: none"> <li>Desarrollar una herramienta usando el lenguaje de macros Microsoft Visual Basic para Aplicaciones (Microsoft VBA) en Excel para determinar los tiempos, asignaciones de trabajo, costos y realizar seguimiento a los proyectos mediante simulaciones Montecarlo.</li> </ul>	Apéndice A Apéndice B Apéndice C Apéndice D Capítulo 5

## **1. Generalidades del proyecto el proyecto**

### **1.1. Objetivos**

#### **1.1.1. General**

Diseñar y desarrollar una herramienta en versión alfa que permita mejorar el proceso de gestión de proyectos de desarrollo de software en la empresa SCITECH S.A.S. mediante simulaciones Montecarlo

#### **1.1.2. Objetivos Específicos:**

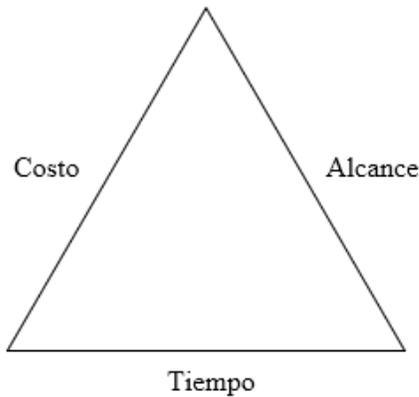
- Realizar un diagnóstico del estado actual de los procesos de gestión de proyectos de la empresa SCITECH S.A.S. comparados con los estándares del PMBOK.
- Diseñar un plan de mejora para la clasificación y asignación de tareas a los desarrolladores con el objetivo de mejorar el tiempo de ejecución y costo de la tarea según la dificultad de la misma.
- Determinar los costos por hora de los involucrados en un proyecto de SCITECH S.A.S. para estimar el costo aproximado de un proyecto con base a las variaciones identificadas.
- Desarrollar una herramienta usando el lenguaje de macros Microsoft Visual Basic para Aplicaciones (Microsoft VBA) en Excel para determinar los tiempos, asignaciones de trabajo, costos y realizar seguimiento a los proyectos mediante simulaciones Montecarlo.

## 1.2. Planteamiento del problema

En la gestión de proyectos hay 3 elementos fundamentales conocidos ampliamente como la triple restricción, los cuales son el costo, tiempo y alcance. La restricción del costo se refiere a la cantidad de dinero presupuestado al inicio de un proyecto para cumplir con una serie de objetivos propuestos. La restricción del tiempo se define como el tiempo dispuesto para llevar a cabo el proyecto y la restricción del alcance se refiere a la totalidad de los productos, servicios o resultados a ser proporcionados. (Project Management Institute, 2017)

### Figura 1.

*Triple restricción en los proyectos*



Como se observa en la figura 1, debido a la alta relación entre sí, la triple restricción se representa comúnmente como un triángulo, donde gráficamente se puede inferir que cada elemento es inherente a los demás y que obligatoriamente la alteración de cada uno de ellos afecta a los otros dos. Con base en esto, se plantea la premisa que, para desarrollar un proyecto de manera exitosa se debe tener un constante control de estas tres variables.

Science & Technology S.A.S. es una empresa colombiana que lleva operando en el país por más de 19 años, brindando soluciones tecnológicas a la medida de los requerimientos del cliente, desde la ejecución de proyectos de desarrollo de software hasta el outsourcing tecnológico. Dentro de la empresa se llevan a cabo una serie de procesos que inician desde el contacto con el cliente por el área comercial, pasan a la planeación del proyecto desde el área de proyectos, continúan con la ejecución de tareas y la revisión de calidad desde las áreas de desarrollo y calidad

o QA (aseguramiento de calidad por sus siglas en inglés) respectivamente y terminan con la puesta en producción del producto creado para el cliente. Estos procesos son supervisados por cada project manager, dirigidos por los procesos de gestión estratégica desde la gerencia y apoyados por las áreas administrativas como recursos humanos, contabilidad e infraestructura.

A pesar que desde el área comercial y el área de proyectos se realiza una estimación inicial sobre la duración de los requerimientos que manifiesta el cliente, estas estimaciones a menudo presentan una variabilidad con respecto a los tiempos ejecutados dado que no se tiene en cuenta las posibles desviaciones generadas por los desarrolladores designados para el proyecto, generando de esta manera, una incorrecta estimación del costo presupuestado y una mala proyección sobre el tiempo requerido. Adicional a esto, la asignación de tareas de desarrollo se realiza solo de acuerdo a la demanda de tareas que se envían para su asignación al coordinador de desarrollo, elementos que juntos provocan atrasos en los avances del proyecto y aumento en el tiempo de ejecución.

Por estas razones, el uso de un modelo matemático determinista, el cual es usado comúnmente en las compañías de desarrollo de software, parte de la suposición de que las variables de entrada se conocen y serán inmutables, por tanto, siempre se obtendrá el mismo resultado (Anderson, Sweeny, Williams, Camm, & Martin, 2011). El uso de este modelo representa un factor de error a la hora de realizar presupuestos y estimar tiempos, por tanto, la implementación de una herramienta que permita simular las variaciones en los tiempos de ejecución de las tareas y asignarlas, según su prioridad y características, al programador más cualificado, mejora el proceso de gestión de los proyectos de desarrollo en su fase precontractual y de ejecución.

### **1.3. Metodología del proyecto**

En esta sección se exponen los pasos que se realizaron para llevar a cabo el proyecto y alcanzar los objetivos planteados.

### ***1.3.1. Reconocimiento y diagnóstico de la técnica actual de estimación de costos y tiempos***

Con el objetivo de identificar los procesos que requieren mejoras, se realizó un reconocimiento de los procesos que se llevan a cabo en la empresa correspondientes a la gestión de los proyectos de desarrollo de software. Posteriormente, se analizaron los procesos identificados y se diagnosticaron los puntos débiles que afectan las estimaciones de tiempo y costos de los proyectos, los cuales fueron objeto de mejora.

Para su desarrollo se realizaron reuniones con distintos gerentes de proyecto, la líder del departamento y el gerente general, además de revisar la documentación interna de la compañía al respecto.

### ***1.3.2. Diseño del plan de mejora basado en la metodología de clasificación y asignación de tareas e implementación del mismo***

Con base en el diagnóstico generado en el primer punto de la metodología, se analizó la manera actual que se realizan las estimaciones y se comparó con los distintos tipos de estimaciones que se realizan en el sector, para posteriormente mediante un análisis basado en varios criterios seleccionar y adaptar una técnica que se ajuste a las características de la empresa.

Con la técnica ya seleccionada, se procedió a diseñar y documentar el plan de mejora que se divide en dos partes: el plan de mejora de la manera de estimar los tiempos y costos de un proyecto y plan de mejora donde se utilice un criterio que facilite y mejore la asignación de las tareas de desarrollo basado en el desempeño de los desarrolladores y la dificultad de la tarea.

Una vez diseñados y documentados los planes de mejora, se procedió a realizar la implementación de las mejoras identificadas.

### ***1.3.3. Determinación de los costos por hora de los involucrados en un proyecto con base a las variaciones identificadas***

A partir de la técnica seleccionada para estimar los tiempos y costos del proyecto, se realizó una tabla que incluía una los costos por hora de los roles involucrados en un proyecto teniendo en cuenta las posibles variaciones de tiempo que pueden existir en un proyecto.

#### ***1.3.4. Diseño y desarrollo de la herramienta en Microsoft VBA en Excel.***

Se diseñó y desarrolló una herramienta en Excel con *Microsoft Visual Basic for Applications*, basada en la metodología planteada y en las necesidades de la empresa. Para esto se necesitó las siguientes etapas:

Diseño y creación de una base de datos que permita almacenar toda la información resultante de los procesos de gestión de software en JIRA.

Integración de la base de datos con JIRA mediante la programación de un script de Python solicitudes a la REST API de JIRA y procedimientos almacenados que limpian y procesan los datos para almacenarlos de manera correcta.

Programación para la lectura de los datos almacenados en la base de datos y posterior, simulación de la duración de las tareas de un proyecto.

## **2. Generalidades de la empresa**

### **2.1. Descripción de la empresa**

Science and Technology S.A.S es una empresa santandereana que lleva operando en el país más de 19 años con experiencia especializada en el desarrollo de soluciones tecnológicas, siendo Solution Silver Partner de Genexus. Su operación se basa en la prestación de servicios de desarrollo de software hecho a la medida con la herramienta low-code Genexus y comercialización productos propios ya desarrollados. La metodología que se utiliza en la empresa para la gestión de proyectos se basa en SCRUM con algunos elementos del PMBok.

### 2.1.1. Información general de la empresa

**Tabla 2.**

*Información general de la empresa*

Item	Descripción
Razón social:	Science and Technology S.A.S
Sigla:	SCITECH S.A.S
NIT:	804010488 – 1
Número de empleados:	45
Dirección:	Carrera 37 # 52 - 43 VIP Center, Ofc. 901

Nota: Información general de la empresa

### 2.1.2. Misión

Science & Technology es una Fábrica de Software dedicada a la prestación de Servicios de Gestión Tecnológica. Fundamentados en nuestros valores institucionales y en una cultura innovadora y de trabajo en equipo, buscamos establecer relaciones estratégicas con nuestros clientes y superar sus expectativas, al igual que las de accionistas y colaboradores, ofreciendo siempre calidad y confiabilidad. (SCITECH)

### 2.1.3. Visión

En 2022 Science & Technology habrá consolidado su proceso de madurez como Fábrica de Software, logrando un nivel satisfactorio de innovación en su gestión tecnológica y un alto estándar de un sistema autogestionado de calidad y de seguridad de la información. Igualmente, logrando la conformación de un número significativo de clientes y aliados

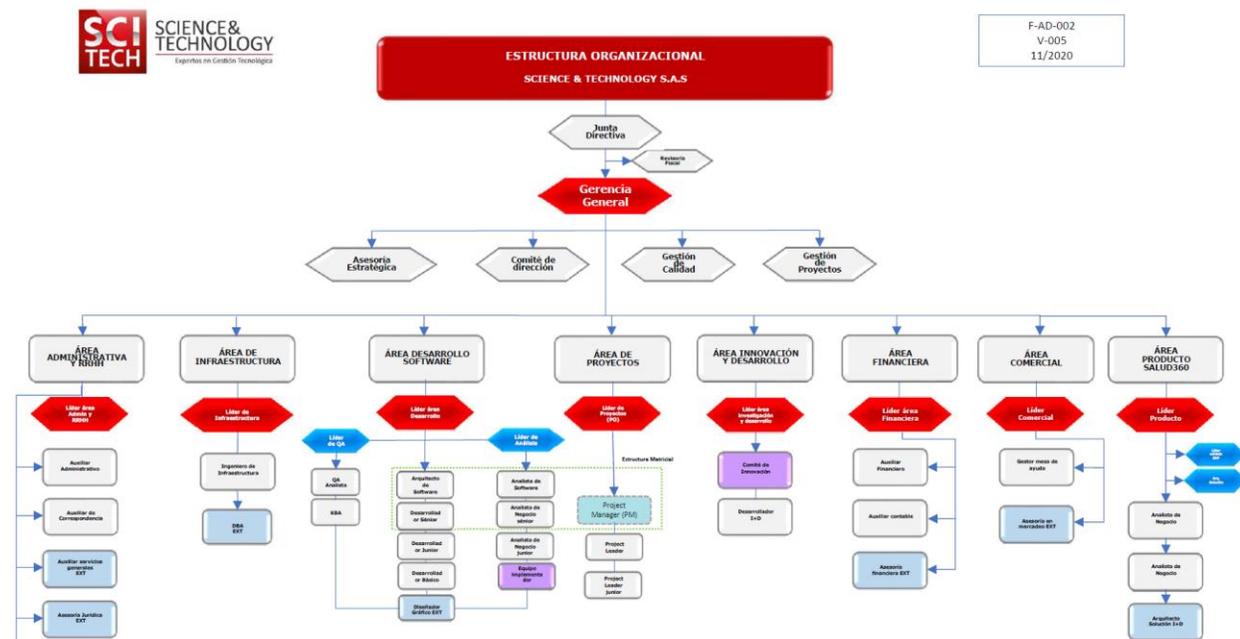
estratégicos a largo plazo que garanticen la proyección y sostenibilidad de la compañía. (SCITECH)

2.1.4. Organigrama

La compañía está compuesta por 8 áreas: área de innovación y desarrollo, área de infraestructura, área de producto Salud360, área de desarrollo, área de proyectos, área administrativa y RRHH, área comercial y área financiera.

Figura 2

Organigrama actual de la empresa

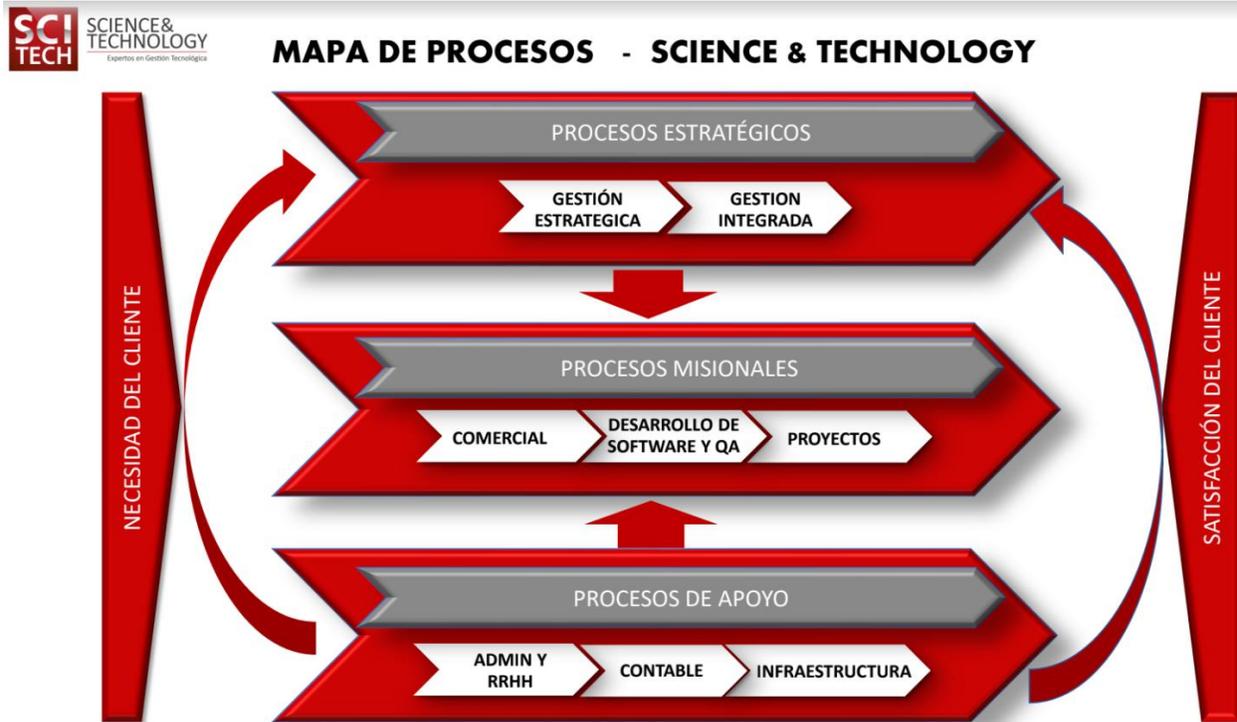


Nota: Imagen tomada de los documentos oficiales de la empresa

### 2.1.5. Mapa de procesos

**Figura 3**

*Mapa de procesos*



Nota: Imagen tomada de los documentos oficiales de la empresa

## 3. Marco de referencia

### 3.1. Marco de antecedentes

El criterio de selección para los trabajos que sirvieron como soporte para el marco de referencia del proyecto, fue que se analice el problema de asignación de tareas o la estimación de tiempos y costos en proyectos de desarrollo de software. Con el objetivo de identificar aportes que sean valiosos para la solución de un problema semejante.

En el trabajo de Margarita André Ampuero de “un modelo para la asignación de recursos humanos a equipos de desarrollo de software” abordó la problemática identificada en las empresas

que se dedican al desarrollo de software de asignación de personal. Este trabajo menciona que, de una investigación realizada a 45 empleados de 12 empresas del sector en Cuba, se podía confirmar que la planeación y las asignaciones de personas representan siempre un problema a la hora de ejecutar los proyectos. (Ampuero, 2015)

Como parte de su investigación, menciona que a pesar de que en las compañías efectivamente se realiza una identificación del personal más apto para las tareas que se asignan, este proceso se realiza de manera empírica y no se tienen evaluaciones de las capacidades de los programadores, de tal manera, se toman constantemente decisiones erróneas dado que no se basan en criterios objetivos. Adicionalmente, resalta que a pesar de que la experiencia y la perspicacia de los gerentes de proyecto es importante y aporta en este proceso, se debe realizar estas asignaciones en conjunto con sistemas o modelos que permitan optimizar la toma de decisiones.

Por otra parte, Nohora Mercado Caruso, Edwin Puerta del Castillo y Katherine Salas Navarro en su trabajo de investigación “Mejora de los procesos de estimación de costos de software. Caso del sector de software de Barranquilla” con base en el método Delphi, recolectaron las opiniones de 16 expertos sobre los distintos escenarios de cómo las empresas pueden llevar a cabo sus procesos de estimación de costos. Los escenarios que se tuvieron en cuenta fueron: juicio de expertos, estimación por analogía, precio para ganar, estimaciones por medio del uso de aplicaciones sistematizadas.

El resultado de este trabajo de investigación, arrojó que, dentro del contexto local, las empresas no usaban los complejos y abstractos modelos propuestos por numerosas fuentes bibliográficas, en cambio, usaban modelos mucho más prácticos y rápidos que se ajustaban a sus necesidades como empresa. Sin embargo, resaltó que “la precisión de las estimaciones es afectada en ocasiones por agentes subjetivos y las malas prácticas en la planificación del proyecto.” (Caruso, Castillo, & Navarro, 2015). De tal manera, ellos proponen una metodología que descarta la utilización de modelos complejos y abstractos para estimar los esfuerzos del desarrollo, e integra las técnicas más empíricas con una revisión sistemática de cada parte del proceso para darle más peso al modelo.

Rahman Ali y otros, en el trabajo “Software Cost Estimation – A Comparative Study of COCOMO-II and Bailey-Basili Models” exponen que en su contexto local los métodos de estimación de costos se realizan de manera empírica y con baja implementación de modelos algoritmos. De tal manera, se realiza una comparación entre datos proveídos por la NASA y Turkis Industry de los modelos COCOMO-II y BaileyBasili. (Rahman, y otros, 2020)

De este trabajo se concluye que la utilización de algunos métodos algoritmos no son tan efectivos. De tal manera, se resalta la importancia de realizar un análisis de las condiciones y características de la compañía y adaptar la técnica o modelo a la misma.

### 3.2. Marco conceptual

- **Alcance del proyecto:** “Trabajo realizado para entregar un producto, servicio o resultado con las funciones y características especificadas.” (Project Management Institute, 2017)
- **Calidad:** “Grado en el que un conjunto de características inherentes satisface los requisitos.” (Project Management Institute, 2017)
- **Ciclo de Vida del Proyecto:** “Serie de fases que atraviesa un proyecto desde su inicio hasta su conclusión.” (Project Management Institute, 2017)
- **Contrato:** “Un contrato es un acuerdo vinculante para las partes en virtud del cual el vendedor se obliga a proveer el producto, servicio o resultado especificado y el comprador a pagar por él.” (Project Management Institute, 2017)
- **Controlar los costos:** “Proceso de monitorear el estado del proyecto para actualizar los costos del proyecto y gestionar cambios a la línea base de costos.” (Project Management Institute, 2017)
- **Costo:** “Efectivo o valor de efectivo equivalente sacrificado por los bienes y servicios que se espera que proporcionen un beneficio presente o futuro a la organización” (Hansen & Mowen, 2006)
- **Datos atípicos:** “Observaciones generadas de forma distinta al resto de los datos.” (Peña, 2014)
- **Datos:** “Mediciones discretas, no organizadas, sin procesar u observaciones crudas.” (Project Management Institute, 2017)

- **Método de Montecarlo:** Es un método estocástico que se usa principalmente para realizar aproximaciones numéricas a expresiones que poseen una alta complejidad o que el costo de realizar pruebas es muy alto. Este método consiste en realizar numerosas veces una generación de datos aleatorios utilizando como intermediaria la función de distribución del sistema con el objetivo de evaluar posibles resultados.
- **Modelo determinístico:** Este modelo predice el mismo resultado desde un punto de partida determinado, ignorando la posible variación aleatoria. (Lawson & Marion, 2008)
- **Modelo estocástico:** “Con este modelo es posible usar distribuciones de probabilidad continuas, lo que permite obtener estimaciones de los valores que puede tomar la variable de respuesta y su desviación estándar.” (Leitón & Abrego, 2009)
- **Presupuesto:** “Estimación aprobada para el proyecto o cualquier componente de la estructura de desglose del trabajo o cualquier actividad del cronograma.” (Project Management Institute, 2017)
- **Riesgo:** “Evento o condición incierta que, si se produce, tiene un efecto positivo o negativo en uno o más de los objetivos de un proyecto.” (Project Management Institute, 2017)
- **Rol:** “Función definida a ser realizada por un miembro del equipo del proyecto, como probar, archivar, inspeccionar o codificar.” (Project Management Institute, 2017)
- **Variable aleatoria:** Es una variable numérica cuyos valores están determinados por el azar. (Peña, 2014)

### 3.3. Marco teórico

#### 3.3.1. Gestión de proyectos de software

Con el objetivo de hablar sobre la gestión de proyectos de software se tiene que comenzar definiendo un proyecto.

Un proyecto es una secuencia de actividades únicas, interconectadas y complejas que tienen un objetivo o propósito que debe ser cumplido dentro de un tiempo específico, dentro de un presupuesto y acorde a una especificación. (Wysocki, 2019)

Por lo tanto, con el objetivo de realizar una buena ejecución y control del proyecto se requiere de seguir una metodología, la cual, según se define como: “La aplicación del conocimiento, habilidades, herramientas y técnicas sobre las actividades para cumplir los requerimientos del proyecto.” (Project Management Institute, 2017)

De tal manera, óptimamente todos los proyectos deben manejar una metodología la cual les permita tener una guía sobre la cual realizar una correcta planeación, ejecución, seguimiento y finalización del mismo. Sin embargo, dependiendo de la industria, la complejidad del proyecto, el equipo de trabajo, la cultura organizacional, entre otras, se pueden utilizar una o distintas metodologías que vayan más acordes a las necesidades del proyecto.

Asimismo, en la industria de desarrollo de software, se usan principalmente dos tipos de metodologías: tradicionales y ágiles. Las metodologías tradicionales, se basan en una descripción de requerimientos donde posteriormente se realiza un análisis y diseño de los mismos, asegurando de esta manera la calidad de los mismos. Sin embargo, esta metodología sigue un proceso secuencial y rígido que no da vuelta atrás, además de que se realiza una gran planeación para una ejecución larga con poca comunicación con el cliente. (Cadavid, Martínez, & Vélez, 2013).

Por otro lado, las metodologías ágiles se caracterizan por ser flexibles y de fácil modificación. Estas metodologías subdividen los proyectos grandes y complejos en unidades más pequeñas donde la comunicación con el cliente es constante y la retroalimentación por su parte es esperada y deseada, ya que se espera que el producto sea mejorado constantemente a medida que las condiciones del mercado van cambiando.

Sin embargo, es importante aclarar que cada vez menos empresas usan las metodologías tradicionales para esta industria, dado que las metodologías ágiles poseen unas características que le brindan ventajas sobre las tradicionales.

**Tabla 3.***Comparación metodologías tradicionales vs ágiles*

<b>Metodologías tradicionales</b>	<b>Metodologías ágiles</b>
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Nota: Tomada de (Cadavid, Martínez, &amp; Vélez, 2013)

**3.3.1.1. Metodologías tradicionales**

De las principales metodologías tradicionales se pueden mencionar las más usadas que son:

**3.3.1.1.1. Métrica V3**

Esta metodología se define como una metodología de planificación, desarrollo y mantenimiento de sistemas de información, la cual fue realizada por el Ministerio de Administraciones Públicas de España. Esta metodología tiene un enfoque hacia los procesos donde se descompone cada uno en actividades y posteriormente en tareas. (Sevilla)

La estructura principal de esta metodología se divide en 3:

- Planificación de Sistemas de Información (PSI)
- Desarrollo de Sistemas de Información, la cual se subdivide en el Estudio de Viabilidad del Sistema (EVS), Análisis del Sistema de Información (ASI), Diseño del Sistema de Información (DSI), Construcción del Sistema de Información (CSI), y la Implantación y Aceptación del Sistema de Información (IAS)
- Mantenimiento del Sistema de Información (MSI)

Adicionalmente, también incluye “un conjunto de interfaces que definen una serie de actividades de tipo organizativo o de soporte al proceso de desarrollo y a los productos.” (Sevilla)

- Gestión de Proyectos (GP)
- Seguridad (SEG)
- Gestión de la configuración (GC)
- Aseguramiento de la calidad (CAL)

#### **3.3.1.1.2. PRINCE2**

PRINCE2 es una metodología para la gestión de proyectos desarrollada por el gobierno británico, la cual según el manual oficial de PRINCE2, se divide en 4 partes principales llamados Elementos: siete Principios, siete Temáticas, siete Procesos, y la Adaptación al entorno del proyecto.

Los siete principios constan de:

- **Justificación comercial continua:** el cual hace referencia a que cada proyecto debe tener como objetivo el retorno de la inversión.
- **Roles y responsabilidades definidos:** el cual hace referencia a que todas las personas deben tener claro qué hacer y qué harán los demás.
- **Enfoque en los productos:** el cual hace referencia a que se debe establecer un criterio único y bien descrito de qué es lo que se desea.

- **Aprender de la experiencia:** el cual hace referencia a que todos los equipos de trabajo deben aprender de las experiencias de los proyectos anteriormente realizados.
- **Gestión por excepción:** el cual hace referencia a que, según el nivel jerárquico de decisión, el inferior no debe informar al superior a menos que se presente alguna Excepción dentro del proyecto, las cuales son afectaciones a la calidad, el alcance, el riesgo o los beneficios.
- **Gestión por fases:** el cual hace referencia a que cada proyecto gestionado con PRINCE2 debe ser dividido y manejado en fases de manera que sea controlado fácilmente.
- **Adaptación al entorno del proyecto:** el cual hace referencia a la necesidad de adaptar PRINCE2 al tamaño y a las necesidades de cada proyecto, de manera que no se realicen procesos innecesarios.

Las siete temáticas constan de:

- **Business Case:** el cual hace referencia a que se debe establecer un criterio para determinar si el proyecto es y se mantiene deseable, viable y alcanzable.
- **Cambio:** el cual hace referencia a la identificación, evaluación y control de cualquier cambio en los requerimientos que se hayan aprobado y definido inicialmente.
- **Organización:** el cual hace referencia a la identificación de la estructura de responsabilidades y roles del proyecto.
- **Planes:** el cual hace referencia al marco de referencia para diseñar, desarrollar y mantener los planes de proyectos que definen dentro del proyecto cómo, cuándo y por quién serán desarrolladas las actividades.
- **Progreso:** el cual hace referencia a la evaluación y verificación del progreso comparado con el plan y cronograma identificado inicialmente.
- **Calidad:** el cual hace referencia a la definición e implementación de un criterio para crear y verificar que los productos cumplen con su uso.

- **Riesgo:** el cual hace referencia a la identificación y control de los riesgos e incertidumbres dentro del proyecto.

Los siete procesos constan de:

- **Puesta en Marcha:** el cual hace referencia al proceso bajo el cual la junta determina si el proyecto vale la pena de realizar.
- **Inicio de un Proyecto:** el cual hace referencia al proceso donde se entiende qué se necesita realizar para cumplir con los productos requeridos.
- **Dirección de un Proyecto:** el cual hace referencia al proceso donde la junta del proyecto toma decisiones clave y ejerce el control del proyecto.
- **Gestión de los Límites de Fase:** el cual hace referencia al proceso donde la junta determina si se puede continuar con la siguiente fase o detener el proyecto.
- **Control de una Fase:** el cual hace referencia al proceso diario que ejecuta el líder del proyecto donde asigna trabajo y le hace seguimiento al mismo.
- **Gestión de la Entrega de Productos:** el cual hace referencia al proceso donde el líder de proyecto y líder del equipo establecen los criterios formales de aceptación, ejecución y entrega de los productos.
- **Cierre de un proyecto:** el cual hace referencia al proceso donde la junta decide si ya efectivamente se realizó todo el trabajo necesario o se solicita un cierre prematuro del proyecto.

### 3.3.1.2. Metodologías ágiles

De las principales metodologías ágiles se mencionan las siguientes más usadas:

#### 3.3.1.2.1. SCRUM

*SCRUM* es un marco de trabajo el cual está basado en el pensamiento *lean* y utiliza un enfoque iterativo e incremental donde cada proyecto se subdivide en unidades más pequeñas llamadas *Sprints* que constan de la lista de requerimientos llamada *Backlog*, especificados por el

*Product Owner*, que es el encargado del proyecto por parte del cliente. Adicionalmente, este marco de trabajo establece unos pilares que fundamental la operación de un proyecto, además una serie de roles y ceremonias cuyo objetivo es evaluar y generar retroalimentación constante sobre el desarrollo del proyecto. (Schwaber & Sutherland, 2020)

Los pilares bajo los cuales se basa *SCRUM* son Transparencia, Inspección y Adaptación. El cual el primero habla sobre que cada tarea realizada debe ser percibida y visible para todas las personas involucradas en el proyecto, de lo contrario, aumenta el riesgo y disminuye el valor del mismo. De esta manera, se permite realizar una inspección a cada etapa y tarea del proyecto. (Schwaber & Sutherland, 2020)

El segundo pilar habla sobre la Inspección, el cual explica que cada objetivo acordado debe ser revisado frecuentemente con el objetivo de detectar posibles desviaciones o potenciales problemas en la ejecución, de tal manera que se permita adaptar el curso del proyecto a las situaciones del entorno.

Y, por último, el tercer pilar Adaptación menciona que en el momento que se identifique alguna desviación fuera de lo aceptable, se deben tomar acciones correctivas.

De igual manera, *SCRUM* establece que la interiorización de los valores Compromiso, Foco, Franqueza, Respeto y Coraje permite que el *SCRUM Team*, sea competitivo y permitan el desarrollo de los *Sprint* de manera óptima. Por lo tanto, dentro de la guía oficial de *SCRUM* se establece la siguiente filosofía:

El *Scrum Team* se compromete a lograr sus objetivos y a apoyarse mutuamente. Su foco principal está en el trabajo del *Sprint* para lograr el mejor progreso posible hacia estos objetivos. El *Scrum Team* y sus interesados son francos sobre el trabajo y los desafíos. Los miembros del *Scrum Team* se respetan entre sí para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan. Los miembros del *Scrum Team* tienen el coraje de hacer lo correcto, para trabajar en problemas difíciles. (Schwaber & Sutherland, 2020)

Por otra parte, los roles que *SCRUM* plantea son *Product Owner*, *SCRUM Master* y *Developers* que en conjunto dan lugar al *SCRUM Team* que debe ser menor a 10 personas donde cada una de ellas debe ser capaz de autogestionarse, es decir, tener la capacidad de decidir quién hace qué, cuándo y cómo.

- **Product Owner:** Este rol es responsable por la comunicación clara y explícita de los objetivos del producto, decidir la organización del *Backlog* y asegurarse que el mismo sea transparente, visible y que se entienda. Adicionalmente, este tiene la decisión final sobre el proyecto.
- **Scrum Master:** Este rol es el responsable de la efectividad del *Scrum Team*, donde él se encarga de ser un facilitador para que todo dentro de la ejecución del proyecto sea efectiva y cada *sprint* represente un incremento al producto final.
- **Developers:** Son las personas encargadas de realizar los incrementos o desarrollos en cada *sprint* del proyecto.

En cada *sprint* se realizan 4 eventos que son: *Sprint Planning*, *Daily Scrum*, *Sprint Review* y *Sprint Retrospective*, diseñados específicamente para favorecer la transparencia durante todo el proyecto, además de generar oportunidades de inspeccionar y adaptar el mismo a las condiciones del entorno. (Schwaber & Sutherland, 2020).

- **Sprint Planning:** Este evento se realiza al inicio de cada *Sprint* y se realiza con el objetivo de planear, definir al equipo porqué es importante el *Sprint* y cómo se realizará el trabajo.
- **Daily Scrum:** Es una reunión diaria de no más de 15 minutos donde los *Developers* discuten los resultados del trabajo y definen un plan para el día siguiente, con el objetivo de eliminar obstáculos de manera temprana y retroalimentar el proceso.
- **Sprint Review:** En esta reunión se lleva a cabo al final de cada *sprint* y realiza una revisión del producto generado a partir del trabajo hecho en el *sprint*, de manera que se pueda evaluar y adaptar para el siguiente.
- **Sprint Retrospective:** Es una sesión de trabajo donde se evalúa el desempeño de las personas, herramientas, interacciones, los procesos, y la definición de cuándo

un producto está listo para ser liberado al público, de manera que se aumente la calidad y la efectividad del equipo para el siguiente *sprint*. (Schwaber & Sutherland, 2020)

#### **3.3.1.2.2. Kanban**

Kanban es una metodología que se basa en la aplicación de los procesos de producción *Just in Time* ideados por Toyota. (Paciencia & Maida, 2015). Esta metodología se fundamenta en que las tareas deberían limitarse, por lo tanto, sólo se debe empezar nuevas cuando los bloques de trabajo anteriores estén terminados. Para realizar el seguimiento, facilitar el control, seguimiento y transparencia de la ejecución del proyecto se utiliza unas tarjetas en un tablero que se ubican en columnas dependiendo de la etapa que se haya definido como necesaria para terminar con la tarea, es decir, si al inicio del proyecto se determinó que para terminar una tarea se necesitaba pasar por un proceso de 6 etapas, el tablero debe contener 6 columnas donde cada tarea según su estado se debe ubicar en una columna de manera que visualmente se aprecien dónde se encuentra cada tarea y qué le falta para ser terminada.

Este método tiene como principal objetivo realizar un control del flujo de trabajo, de manera que se pueda apreciar rápidamente cuellos de botella o problemas de recursos con el fin de tomar acciones tempranas en el proyecto. Adicionalmente, su sencilla estructura facilita y permite integrarse con distintas metodologías donde la ejecución del proyecto requiera pasar por un proceso secuencial o se tengan etapas definidas.

#### **3.3.2. Modelos y técnicas de estimación de tiempos y costos de desarrollo de software**

La de estimación de costos de un software se puede definir como el proceso mediante el cual se intenta predecir el esfuerzo que se requiere para construir y desarrollar un sistema de software, el cual, tiene como principal componente el factor humano. (Mercado-Caruso, Castillo, & Oliveira, 2017) La clave principal para desarrollar una buena estimación, se basa en el análisis y retroalimentación de todos los datos que se obtienen en el proceso de desarrollo. (Walston & Felix, 1997).

Esta estimación de costos se puede realizar de con distintas técnicas o métodos que se pueden ajustar según las necesidades de la compañía que esté realizando este proceso. Se pueden clasificar de acuerdo a las siguientes categorías y se explican los más usados:

**3.3.2.1. Modelos paramétricos/de algoritmos.** Son aquellos que generan las estimaciones del costo del proyecto mediante una función que está compuesta de distintos factores. Donde se tiene la ecuación donde E es el esfuerzo total y CF hace referencia a los factores de costos. **(Rashid, Nisar, Mahmood, Rehman, & Arafat, 2019)**

$$E = (CF_1, CF_2, CF_3, \dots, CF_n) \quad (1)$$

**3.3.2.1.1. Modelo Putnam o SLIM (Software Life-Cycle Model).** Se basa en una serie de fórmulas de estimación software, las cuales se extraen del análisis de bases de datos de proyectos anteriores, observando cómo se comportaron las estimaciones software y distribuciones de esfuerzo.

**3.3.2.1.2. COCOMO.** Fue desarrollado en 1981 y se encarga de calcular las horas de trabajo y costos como una función de distintas variables como el tamaño del programa, funcionalidades, líneas de código, entre otras.

**3.3.2.2. Métodos basados en juicio de expertos/consenso.** Estas técnicas están basadas en la capacidad de un grupo de expertos, de reunirse y trabajar para desarrollar una estimación en conjunto del esfuerzo que puede requerir un software.

**3.3.2.2.1. Método Delphi.** Este método se basa en una serie de pasos donde se les pide a expertos que califiquen individualmente unos requerimientos específicos, luego, se realiza una retroalimentación sobre las calificaciones que realizaron sus pares, y se vuelve a realizar el proceso hasta que se logre llegar a un consenso.

**3.3.2.2.2. Bottom-up.** Se basa en la descomposición de los requerimientos en elementos mucho más pequeños para que de esta manera sea mucho más sencillo obtener una estimación global de todo el proyecto.

**3.3.2.2.3. Planning Poker.** Esta técnica se basa en una serie de rondas donde el cliente le explica los requerimientos a un grupo de expertos, los cuales poseen cartas que poseen unos puntos de historia basados en la serie de Fibonacci que representan las dificultades de los requerimientos y al finalizar cada ronda, los expertos muestran al tiempo la carta que representa la dificultad que cada uno considera. Si los resultados son muy dispares, cada uno expone su opinión y se realiza el mismo proceso hasta que se llegue a un consenso.

**3.3.2.3. Modelos de aprendizaje.** Se apoya en la tecnología con enfoque de aprendizaje al igual que de información actual para desarrollar modelos de estimación. Estas técnicas aprenden de experiencias pasadas y construyen un modelo para automatizar el proceso de estimación. **(Rashid, Nisar, Mahmood, Rehman, & Arafat, 2019)**

**3.3.2.3.1. Razonamiento basado en casos.** Este modelo asume que casos similares tienen soluciones similares. Por lo tanto, todas las historias y casos son almacenados en una base de datos, en la cual, el sistema se encarga de buscar coincidencias a los casos que se necesitan actualmente y les brinda la misma solución.

**3.3.2.3.2. Redes neuronales.** Estos modelos se encargan de usar datos históricos y ajustar automáticamente los parámetros de sus algoritmos para reducir el error. **(Rashid, Nisar, Mahmood, Rehman, & Arafat, 2019)**

**3.3.2.4. Modelos basados en regresiones.** Su uso es muy frecuente debido a su facilidad. Estos constan de algoritmos matemáticos que estiman el costo de un software como una función de otros factores.

**3.3.2.4.1. Regresión estándar.** Se basa en el método de los mínimos cuadrados. Es muy popular debido a que es muy simple y se puede realizar de manera sencilla por cualquier software

de manejo estadístico o matemático. Se puede escribir como la ecuación (2) y representa el tiempo que toma realizar determinada labor con una serie de factores que toman papel.

$$y_t = \beta_1 + \beta_2 X_2 + \dots + \beta_k X_k + e_t \quad (2)$$

**3.3.2.4.2. Regresión robusta.** Es una versión mejorada de la regresión estándar la cual lidia de manera más sencilla con los datos atípicos debido a que se utiliza en softwares especializados.

### **3.3.2.5. Técnicas basadas en la estimación del tamaño de software.**

**3.3.2.5.1. Puntos de función.** “Su propósito es medir el software cualificando la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema.” (*Portillo & Paredes, 2015*). Su ventaja radica en que proporcionan una medida objetiva, cuantitativa y auditable del tamaño de lo que se desee desarrollar, desde el punto de vista de los requerimientos comunicados por el usuario final del desarrollo. (*Rodríguez, 1999*). Esta técnica consta de 8 pasos los cuales son:

1. Planificación del conteo de Puntos Función.
2. Recogida de información.
3. Cálculo del Factor de Ajuste (VAF).
4. Inventariado de transacciones lógicas y ficheros lógicos.
5. Clasificación de componentes.
6. Revisión de las 14 características generales del sistema (GSC's).
7. Tabulación de resultados.
8. Validación de resultados.

**3.3.2.5.2. Puntos de casos de uso.** Su objetivo consiste en estimar las horas que se necesitarán para llevar a cabo el desarrollo del software y cuántas personas se necesitan para hacerlo. Se realiza mediante 3 pasos, los cuales constan de: primero, calcular los puntos de casos de uso no ajustados (UUCP) mediante el peso de los actores (AUW) y de los casos de uso (UUCW), es decir, la cantidad de actividades que se desarrollarán en el requerimiento y su complejidad según el actor

que las desarrolle. El resultado de estos pesos se debe sumar una vez que hayan determinado. De tal manera, empezando por el peso de los actores (AUW) se tiene la tabla 5.

**Tabla 4.**

*Peso de los actores*

<b>Tipo de actor</b>	<b>Descripción</b>	<b>Factor</b>
Simple	Otro sistema con una API definida	1
Medio	Otro sistema interactuando con algún protocolo (TCP) o una persona interactuando a través de una interfaz en modo texto	2
Complejo	Una persona interactuando a través de una interfaz gráfica de usuario	3

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso.

Una vez llenada esta tabla, se debe contar la cantidad de usuarios o actores en el sistema, multiplicar cada tipo por su factor de peso y sumar esos valores para obtener el (AUW). (Orea, 2010). Luego, se procede a estimar el peso de los casos de uso con la tabla 6.

**Tabla 5.**

*Peso de los casos de uso*

<b>Tipo de caso de uso</b>	<b>Descripción</b>	<b>Factor</b>
Simple	3 transacciones o menos	5
Medio	4 a 7 transacciones	10
Complejo	Más de 7 transacciones	15

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso.

De igual manera, se realiza la misma operación para hallar el peso de los casos de uso (UUCW).

Segundo, para calcular los puntos de casos de uso (UCP), se utiliza como guía la tabla 7 y la tabla 8 donde se determina el peso de los factores técnicos (TCF) de los módulos que se vayan a desarrollar y las escalas de estimación de TCF respectivamente.

**Tabla 6.**

*Factores técnicos*

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>
T1	Sistema distribuido	2
T2	Objetivos de performance o tiempo de respuesta	1
T3	Eficiencia del usuario final	1
T4	Procesamiento interno complejo	1
T5	El código debe ser reutilizable	1
T6	Facilidad de instalación	0,5
T7	Facilidad de uso	0,5
T8	Portabilidad	2
T9	Facilidad de cambio	1
T10	Concurrencia	1
T11	Objetivos especiales de seguridad	1
T12	Acceso directo a terceras partes	1
T13	Facilidades especiales de entrenamiento a usuarios	1

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso.

**Tabla 7.**

*Escalas de estimación TCF*

Descripción	Factor
Irrelevante	De 0 a 2
Medio	De 3 a 4
Esencial	5

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso

Una vez lista la tabla, se realiza una multiplicación de la estimación de cada factor por su peso y se suma esos productos para obtener el total (TFactor). Luego, para obtener el TCF, se utiliza la fórmula (3) propuesta por el creador del método, obteniendo de esta manera el peso de los factores técnicos. (Orea, 2010)

$$TCF = 0.6 + (0.01 * TFactor) \quad (3)$$

Posteriormente, para determinar el peso de los factores ambientales se usa la tabla 9, la cual hace referencia a las habilidades y experiencias del grupo de desarrollo usando las escalas de la tabla 10.

**Tabla 8.**

*Factores ambientales*

Factor	Descripción	Factor
E1	Familiaridad con el modelo del proyecto utilizado	1,5
E2	Experiencia en la aplicación	0,5

E3	Experiencia en orientación a objetos	1
E4	Capacidad del analista líder	0,5
E5	Motivación	1
E6	Estabilidad en los requerimientos	2
E7	Personal de medio tiempo	-1
E8	Dificultad en el lenguaje de programación	-1

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso

### Tabla 9.

*Escalas de estimación de EF*

Descripción	Factor
Sin experiencia, sin motivación, estabilidad	De 0 a 2
Promedio	3
Amplia experiencia, motivación, estabilidad	De 3 a 5

Nota. Adaptada de Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso

La suma de cada multiplicación de los factores por su peso da como resultado el (EFactor), el cual se usa para hallar el peso de los factores ambientales (EF) mediante la fórmula (4) recomendada por el autor.

$$EF = 1.4 + (-0.03 * EFactor) \quad (4)$$

Tercero, la estimación de las horas hombre se realiza mediante la fórmula 5. Donde el 20 hace referencia la sugerencia de usar 20 horas hombre por cada UCP. “Por ejemplo, para un

sistema de 60 UCP \* 20 hrs/hombre nos da un total de 1200 hrs/hombre. Lo que equivale a 30 semanas (40 hrs por semana para una persona), de esta forma, un equipo de 5 personas desarrollaría el sistema en 6 semanas.” (Orea, 2010)

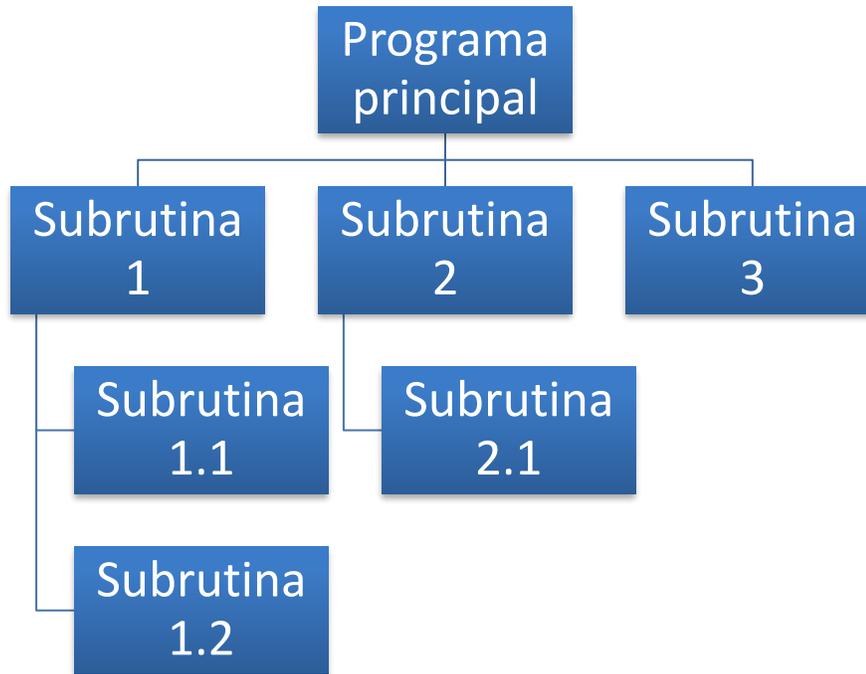
$$HH = UCP * 20 \quad (5)$$

### ***3.3.3. Diseño y desarrollo de herramientas de software***

Para la construcción y diseño de los sistemas de software generalmente se realiza una conceptualización a grandes rasgos del enfoque bajo el cual se resolverá la serie de requerimientos realizados por el cliente. De tal manera, para realizar este proceso, se tienen distintos patrones bajo los cuales cada arquitecto de software se puede basar para diseñar el software, unos ejemplos de ellos son la arquitectura del *Main program – Subroutines*, *Pipe-and-Filter*, Orientado a objetos, cliente - servidor, entre otros. Finalmente, dependiendo del patrón seleccionado, se procede a programar el software.

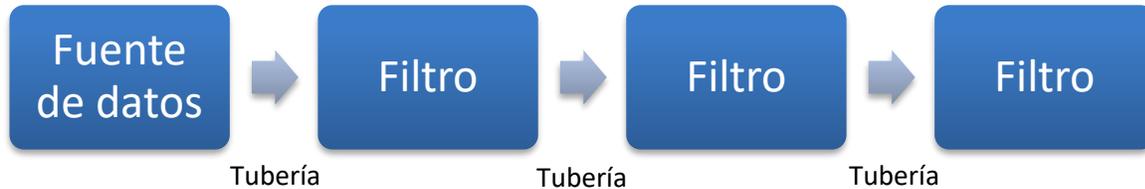
#### **3.3.3.1. Main program – Subroutine**

Este tipo de arquitectura es el más tradicional y viejo patrón de arquitectura en el cual se empieza con un problema grande, y se descompone en tareas más pequeñas e independientes. Una vez se descompone las tareas se procede nuevamente a descomponer esas tareas en tareas aún más pequeñas y así sucesivamente hasta el punto que se tenga un problema pequeño y fácil de resolver.

**Figura 4.***Arquitectura Main program - Subrutina*

### 3.3.3.2. Pipe-and-filter

Este patrón de arquitectura consiste en que cada componente es una entidad independiente llamada filtro, la cual reciba datos los transforma y da como resultado datos, ya sea transformados o producidos. Este tipo de arquitectura tiene varias ventajas como la reusabilidad dado que cada filtro puede ser llamado miles de veces y la flexibilidad dado que cada filtro puede ser cambiado sin necesidad de modificar los demás. Sin embargo, su principal desventaja es que cuando un sistema es bastante grande su rendimiento puede reducir dado que su gasto computacional puede ser alto.

**Figura 5.***Arquitectura Pipe-and-filter*

### 3.3.3.3. Orientada a objetos

El tipo de arquitectura orientado a objetos se basa principalmente en un sistema de objetos que interactúan entre sí. De tal manera, para crear el sistema de objetos se requiere identificar cada uno de los actores y elementos que forman parte del problema que se pretende resolver, por ejemplo, si se requiere un sistema que los empleados administrativos puedan consultar facturas, los cajeros puedan crear y añadir productos a las mismas, haciendo el análisis y diseño para el sistema se identifican los actores administrativos, los cajeros, las facturas y los productos, por lo tanto, en el diseño del sistema se tienen que crear 4 clases, las cuales corresponden a cada actor dentro del proceso. La primera hace referencia a los administrativos donde ellos sólo pueden consultar las facturas, es decir sólo tienen un método, siendo cada método las acciones vinculadas a cada clase, y unos atributos como el nombre, edad, entre otros, que hacen referencia a las cualidades de cada clase, propios de cada usuario diferente que se tenga o, mejor dicho, propios de cada instancia. De igual manera, se realiza el diseño para los cajeros que tienen unos atributos como el nombre, edad, puesto, salario, etcétera, y unos métodos como el crear facturas. Aplicando la misma lógica, se haría lo mismo para las facturas y los productos.

Las ventajas de este tipo de arquitectura, son numerosas donde se destaca principalmente que es fácil de realizar mantenimiento y expansión al modelo general, adicionalmente, en algunos casos puede haber una relación directa entre la realidad y el modelo, además de que cada objeto es reutilizable.

#### **3.3.3.4. Cliente-Servidor**

En este tipo de arquitectura, los programas se dividen en dos partes donde cada una corre en un computador diferente. Una de ellas corre en un servidor en cual cuenta con un poder de procesamiento alto y que provee los servicios a través de todo el internet. Por otra parte, el Cliente el cual maneja la parte sólo los datos de entrada, despliega los datos y sirve de comunicación con el servidor. (Dooley, 2017) En el servidor se evalúan todas las solicitudes recibidas del cliente, este realiza todo lo necesario para cumplir con ella y devuelve los datos procesados.

#### **3.3.4. Técnicas de simulación**

La simulación es una técnica donde mediante la construcción de modelos se pretende recrear la realidad y tomar decisiones a partir de los resultados de la misma. Estas técnicas de simulación se pueden dividir según las características de los modelos dado que según las condiciones que se quiera evaluar, estas toman nombres diferentes.

##### **3.3.4.1. Según las variables de estado**

###### **3.3.4.1.1. Simulación discreta**

La simulación discreta se caracteriza porque los cambios solamente pueden ocurrir en momentos separados en el tiempo, es decir, cada cierto tiempo las variables toman un valor distinto. Un ejemplo de esto se puede ver el nivel de agua de una bañera que está siendo llenada mediante botes de agua.

###### **3.3.4.1.2. Simulación continua**

La simulación continua se refiere a que los cambios en las variables se están generando en cada momento del experimento, de tal manera los cambios ocurren de manera continua durante el tiempo que ocurre el experimento. Un ejemplo de esto se puede ver el nivel de agua de una bañera que está siendo llenada mediante una llave.

### **3.3.4.2. Según la evolución en el tiempo**

#### **3.3.4.2.1. *Simulación estática***

En este tipo de simulaciones el tiempo no juega ninguna importancia, por lo tanto, para realizar el análisis de este tipo de simulación no se realiza teniendo en cuenta el tiempo. Un ejemplo de esto sería el cálculo del peso total que puede cargar un barco, donde solamente se ingresan variables que el valor de ellas en el tiempo no es relevante.

#### **3.3.4.2.2. *Simulación dinámica***

Las simulaciones dinámicas se usan principalmente en los estudios donde se requiere analizar el comportamiento de algo a través del tiempo. Por ejemplo, la velocidad de un barco donde según la época y momento del día aumentará o disminuirá,

### **3.3.4.3. Según aleatoriedad**

#### **3.3.4.3.1. *Simulación determinista***

Las variables en las simulaciones deterministas no corresponden a un evento azaroso, de tal manera que las variables que se ingresan al modelo en ningún momento cambian y si se tienen las mismas condiciones iniciales el resultado siempre será el mismo.

#### **3.3.4.3.2. *Simulación estocástica***

En este tipo de simulación las variables se comportan de manera aleatoria donde así se tengan las mismas condiciones iniciales el resultado será diferente.

### **3.3.5. Mejoramiento de procesos**

El mejoramiento de procesos es parte fundamental del avance y crecimiento de una empresa, dado que a medida que esta crece, se vuelve necesario revisar todos los procesos que se llevan a cabo e identificar oportunidades de mejora con el objetivo de eliminar los errores, minimizar las demoras y maximizar el uso de activos (Harrington, 1998). De tal manera, se define que el mejoramiento de procesos es la herramienta fundamental para el aumento de la calidad de los servicios y/o productos, satisfacer requerimientos del cliente y/o reducir costos (Rojas & Caballero, 2016) con la condición de que la solución propuesta sea más fácil de emplear, adaptable con las necesidades del cliente y proporcione a la organización una ventaja competitiva. (Harrington, 1998)

## **4. Diagnóstico inicial**

La recolección de información para el diagnóstico de los procesos de gestión de proyectos se realizó mediante la revisión de documentos internos y reuniones informales con diferentes líderes de proyecto y la líder del departamento de proyectos, con el objeto de determinar los inconvenientes que genera la actual forma de gestionar los proyectos y asignar las tareas.

### **4.1. Reconocimiento de los procesos**

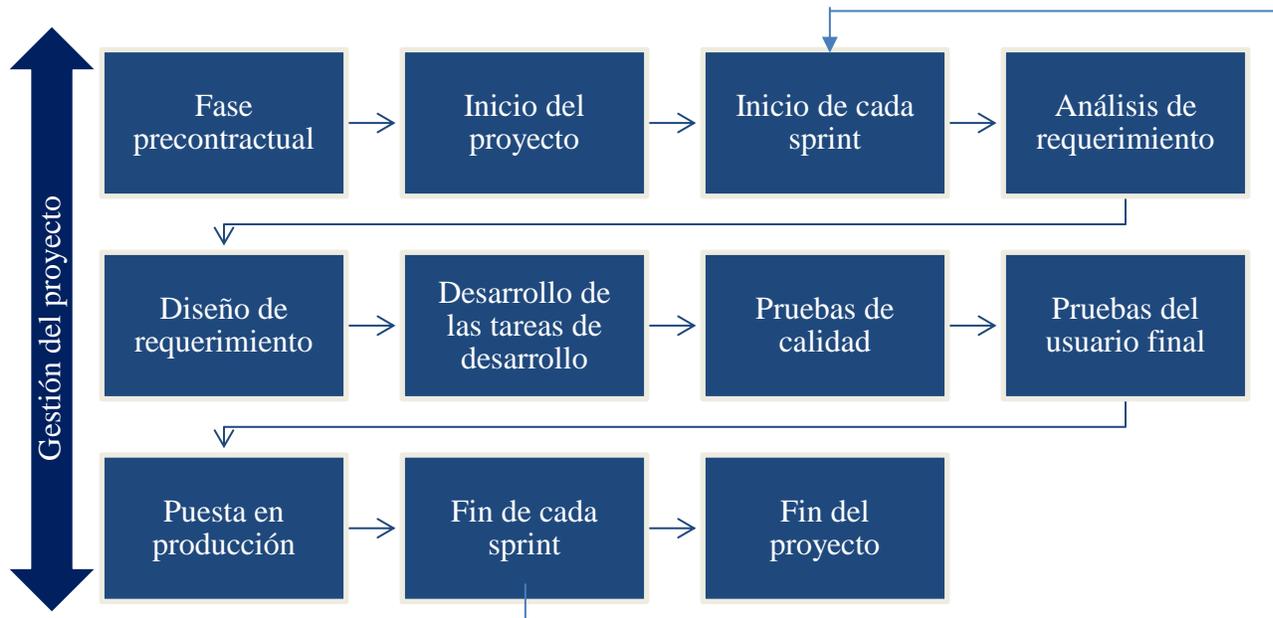
De acuerdo con el mapa de procesos de la empresa Science and Technology S.A.S., la compañía cuenta con tres procesos: la comercialización de productos de software ya elaborados y contacto con el cliente, la dirección de proyectos de desarrollo de software y el desarrollo de software y QA (*quality assurance* por sus siglas en inglés). A continuación, se dará una breve explicación de los dos últimos, que están entrelazados, dado que estos son el objeto de análisis y mejora del proyecto.

#### **4.1.1. Dirección de proyectos de desarrollo de software.**

Para dar una explicación mucho más clara del proceso se propone una ilustración gráfica (figura 3), donde se expone el proceso iterativo de la ejecución de los proyectos de desarrollo de software, que se enmarca dentro del marco de trabajo SCRUM. Se creó esta ilustración con base en la documentación mencionada anteriormente.

**Figura 6.**

*Diagrama del proceso de desarrollo de software*



Durante el inicio del proceso de gestión de proyectos, en la fase precontractual se lleva a cabo la estimación del tiempo requerido para todo el proyecto, donde el cliente realiza una lista de requerimientos y posteriormente la empresa mediante juicio de expertos realiza una evaluación de la posible duración de los requerimientos especificados. Posteriormente, se interactúa con el proceso de desarrollo de software y se inician las actividades correspondientes al análisis, diseño, desarrollo, pruebas y despliegue.

Para la primera actividad se realiza análisis más detallado del requerimiento junto con el cliente donde se documenta con alto detalle lo que se necesita. Luego de esto va el diseño donde se estructura cómo las bases de datos y los requerimientos interactúan. Consecutivamente se envían a asignación las tareas de desarrollo a cada programador con base en la estructura planteada

por el diseñador y arquitecto en la actividad anterior. Inmediatamente, va el aseguramiento de calidad o QA (quality assurance por sus siglas en inglés) donde se realizan pruebas que aseguran el correcto funcionamiento de los desarrollos y la documentación de los manuales y documentos para el cliente. Después, la puesta en producción que hace referencia al despliegue del producto desarrollado de manera pública. Estas actividades se repiten una cantidad de veces definidas según los sprints o bloques de requerimientos que se hayan definido en la fase precontractual donde cada uno es supervisado por el project manager o gerente de proyecto en conjunto al gerente general de la compañía.

#### **4.1.2. Desarrollo de software y QA.**

Este proceso se dedica a asignar, realizar, probar y hacer seguimiento a las tareas de desarrollo que son enviadas por los gerentes de proyecto. Cada semana estos envían las tareas al coordinador de desarrollo, las cuales son resultado del diseño basado en el análisis realizado junto con el cliente. Para esta asignación se tiene en cuenta la demanda de horas que se envían de cada proyecto y se distribuye según la capacidad disponible en el momento. Una vez realizada la tarea, se marca como lista para QA y el líder del área se encarga de asignar a un QA Tester o asegurador de calidad para que este revise el total cumplimiento y funcionalidad de la asignación. En caso de que esta no cumpla con los estándares, se marca como devuelta y el programador encargado corrige para nuevamente realizar el ciclo de pruebas.

#### **4.1.3. Proceso de estimación de tiempos**

En la fase precontractual donde se lleva a cabo la estimación del tiempo proyectado para cada requerimiento, se realizan una serie de reuniones donde el cliente expone a groso modo las necesidades que presenta y plantea la solución que requiere. Después, la empresa mediante juicio de expertos realiza una evaluación de la posible duración de la construcción de cada requerimiento especificado. Esta evaluación de tiempos inicialmente contempla únicamente el tiempo necesario para programar el requerimiento.

Luego, con base en este valor, se destina la duración de las demás actividades en el proceso de ejecución del proyecto y el proceso desarrollo y QA, es decir, análisis, diseño, pruebas, despliegue y documentación, como la multiplicación del tiempo de programación por el porcentaje correspondiente a cada actividad como se puede observar en la tabla 1. Finalmente, se distribuyen los requerimientos y los tiempos en distintos sprints, que constan de conjuntos de requerimientos relacionados y finalmente se tiene como resultado el tiempo total para el proyecto.

**Tabla 10.***% de distribución de tiempo*

<b>Concepto</b>	<b>Valor en horas</b>	<b>% del total</b>
Análisis	Horas de construcción * 10%	10%
Diseño	Horas de construcción * 20%	20%
Desarrollo	Horas de construcción	55%
Pruebas	Horas de construcción * 10%	10%
Despliegue	Horas de construcción * 5%	5%
<b>TOTALES HORAS PRESUPUESTADAS</b>		<b>100%</b>

#### **4.1.4. Proceso de estimación de costos**

Con base en esta estimación, para calcular el presupuesto del proyecto se tiene una lista de precios base por hora donde se discrimina el costo de cada rol que se asigna para la ejecución del proyecto como, por ejemplo: desarrollador junior, desarrollador senior, analista de requerimientos, analista de QA o *QA Tester*, diseñador y/o arquitecto y el gerente de proyecto, y se pasa a multiplicar estos valores por los porcentajes que se pueden observar en la tabla 2. Adicionalmente, se considera qué tanto se involucrará a un desarrollador junior y a un desarrollador senior, dado

que dependiente de la dificultad de los requerimientos, se necesitará involucrar a alguien con más experiencia y formación. Finalmente, se suma los valores de licencias, servidores, capacitación a los usuarios finales si da lugar y los costos indirectos, para hallar el presupuesto total del proyecto.

**Tabla 11.***Definición del presupuesto*

<b>Descripción</b>	<b>Horas/Cantidad</b>	<b>Costo unitario</b>	<b>Subtotal</b>
<b>Equipo Desarrollo</b>			
Gerencia de proyecto	Horas totales presupuestadas * 15%	Costo salarial / 160 h al mes	Costo parcial total
Analista de software	Horas totales presupuestadas * 10%	Costo salarial / 160 h al mes	Costo parcial total
Diseñador o Arquitecto	Horas totales presupuestadas * 20%	Costo salarial / 160 h al mes	Costo parcial total
Desarrollador senior	Horas totales presupuestadas * 10% * % Peso de desarrollo senior	Costo salarial / 160 h al mes	Costo parcial total
Desarrollador junior	Horas totales presupuestadas * 10% * % Peso de desarrollo junior	Costo salarial / 160 h al mes	Costo parcial total
Analista de QA	Horas totales presupuestadas * 10%	Costo salarial / 160 h al mes	Costo parcial total
<b>Administrativo</b>			
Área Financiera	10 horas	Costo salarial / 160 h al mes	Costo parcial total

Área Comercial	10 horas	Costo salarial / 160 h al mes	Costo parcial total
<b>Licencias de desarrollo</b>			
Licencias Genexus	Cantidad licencias	Valor licencia	Costo parcial total
Licencias K2B TOOLS	Cantidad licencias	Valor licencia	Costo parcial total
Licencias Gx Server	Cantidad licencias	Valor licencia	Costo parcial total
Licencias Jira	Cantidad licencias	Valor licencia	Costo parcial total
Licencia Zoom	1	Valor licencia	Costo parcial total
<b>Hardware y Comunicaciones</b>			
Servidor de desarrollo	Meses de duración del proyecto	Costo prorrateado en todos los proyectos	Costo parcial total
Servidor de Base datos	Meses de duración del proyecto	Costo prorrateado en todos los proyectos	Costo parcial total
Servidor de pruebas	Meses de duración del proyecto	Costo prorrateado en todos los proyectos	Costo parcial total
Horas de capacitación	Horas programadas	Valor	Costo parcial total
<b>Otros Costos</b>			
Puesto de trabajo	Meses de duración del proyecto		Costo parcial total
<b>TOTAL PRESUPUESTO</b>			

#### **4.1.5. Proceso de asignación de tareas de desarrollo**

Una vez aceptado por el cliente el valor comercial establecido para el proyecto, se procede a asignar un gerente de proyecto y se da inicio al mismo mediante un *kickoff* o reunión de inicio. Posteriormente, después del análisis y el diseño de las tareas de desarrollo, estas pasan a ser asignadas mediante un proceso que involucra al coordinador de desarrollo y el líder del proyecto, el cual, mediante un correo envía al primero una lista de subtareas donde se especifica las horas estimadas para cada una de ellas, para que el coordinador las asigne al desarrollador según la disponibilidad de los ingenieros asignados al proyecto y la demanda de todos los proyectos

#### **4.1.6. Análisis del proceso de estimación de tiempos y costos actual**

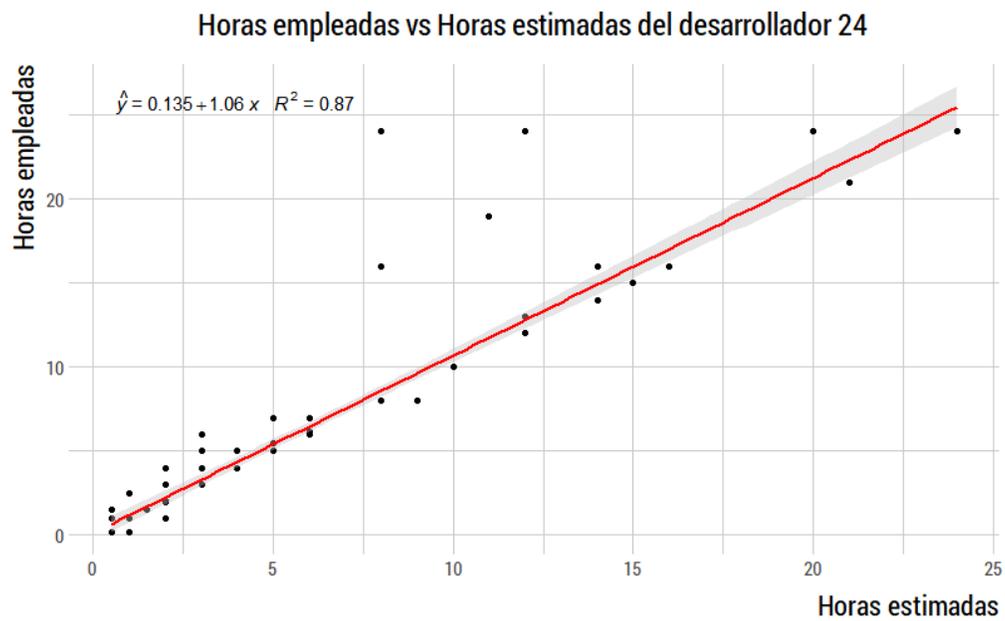
De acuerdo a la información recolectada, se puede observar que en las estimaciones de tiempos y costos existe la posibilidad de numerosas desviaciones del presupuesto dado que no se tienen en cuenta aspectos como la variabilidad asociada a cada desarrollador, analista y QA tester. Esta variabilidad se debe a que, dependiendo de la complejidad de los requerimientos, la experiencia y otros factores únicos de cada persona, se tendrá un rendimiento único y, por lo tanto, el tiempo de duración de cada tarea está sujeto a la persona que se asigne desde el inicio del proyecto.

De tal manera, estas estimaciones realizadas mediante un enfoque determinístico, como manifestó el gerente de la compañía, a menudo varían de los datos que se obtienen durante las ejecuciones de los proyectos, evidenciando retrasos y desajustes en los presupuestos de los proyectos en ejecución. Esto es demostrable en la información que se encuentra en el Apéndice G, donde cada dato representa una tarea asignada anteriormente a algún desarrollador o informador, el eje x la duración estimada de la tarea y el eje y la duración real. Por lo tanto, en los casos donde se tiene información suficiente, se logra evidenciar un comportamiento único para cada desarrollador e informador donde en la mayoría de los casos se tiene una pendiente positiva mayor a 1 con un correlación media o alto, indicando que los tiempos estimados son menores a los tiempos realmente ejecutados.

Las desviaciones en costos y tiempos se pueden ver ejemplificadas en el siguiente caso. Si se emplea en un proyecto muy pequeño 2 desarrolladores senior y 1 analista y usando para este ejemplo los modelos de regresiones lineales de las figuras 4, 5 y 6 tendríamos los siguientes datos.

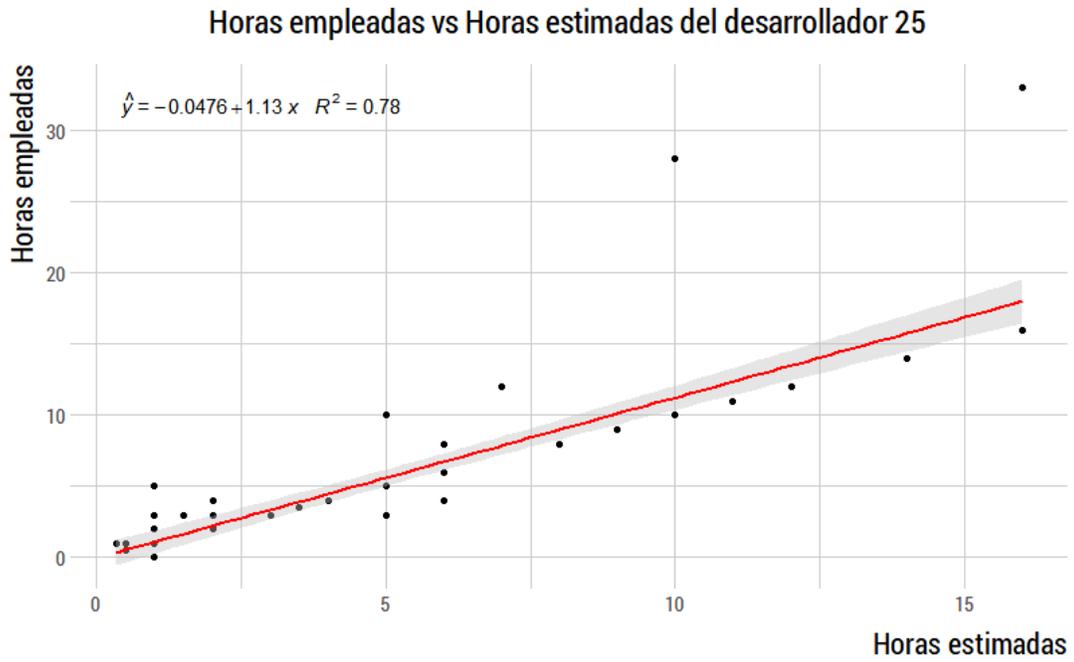
**Figura 7.**

*Regresión lineal de horas empleadas vs horas estimadas por desarrollador 24*



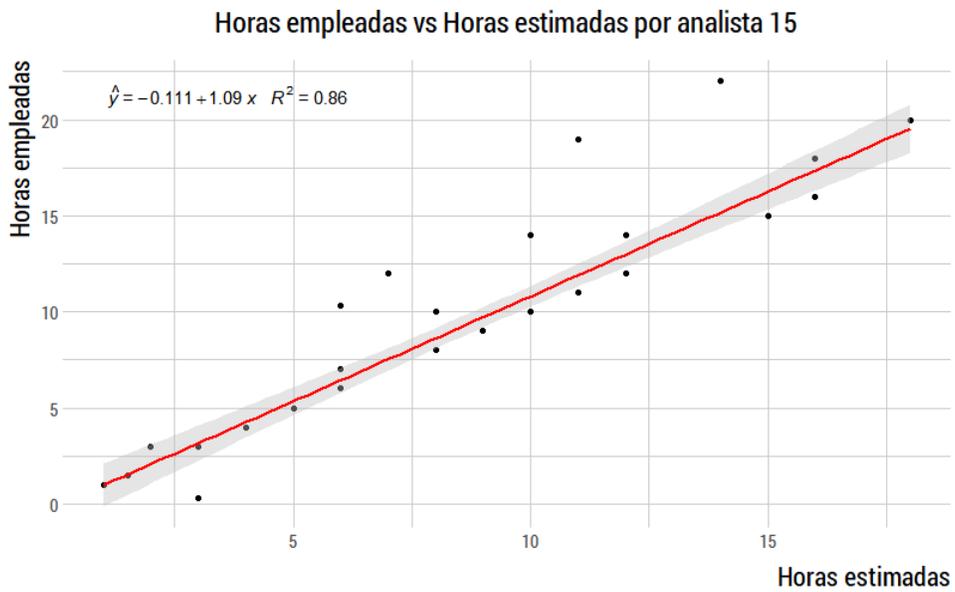
**Figura 8.**

*Regresión lineal de horas empleadas vs horas estimadas por desarrollador 25*



**Figura 9.**

*Regresión lineal de horas empleadas vs horas estimadas por analista 15*



Una vez ajustados los tiempos que se estimaron inicialmente con los tiempos del modelo usados para el ejemplo se presentarían las siguientes desviaciones.

**Tabla 12..**

*Tiempos ajustados.*

# Requerimiento	Tiempo estimado	Tiempo ajustado del analista 15	Desarrollador asignado	Tiempo empleado pronosticado
Requerimiento 1	30	32,589	Desarrollador 24	34,67934
Requerimiento 2	7	7,519	Desarrollador 25	8,44887
Requerimiento 3	10	10,789	Desarrollador 24	11,57134
Requerimiento 4	28	30,409	Desarrollador 25	34,31457
Requerimiento 5	27	29,319	Desarrollador 24	31,21314
Requerimiento 6	5	5,339	Desarrollador 25	5,98547
Requerimiento 7	17	18,419	Desarrollador 24	19,65914
Requerimiento 8	26	28,229	Desarrollador 25	31,85117
Requerimiento 9	15	16,239	Desarrollador 24	17,34834
Requerimiento 10	10	10,789	Desarrollador 25	12,14397
<b>Total</b>	<b>175</b>	<b>189,64</b>		<b>207,21535</b>

Valores los cuales, traducidos a dinero, representan una gran cantidad sólo para el proceso de desarrollo y análisis.

**Tabla 13.***Costos ajustados*

<b>Descripción</b>	<b>Costo con tiempo estimado</b>	<b>Costo con tiempo pronosticado</b>	<b>Variación</b>
Gerente de proyecto	\$ 1.030.255	\$ 1.219.913	\$ 189.657
Análisis	\$ 401.013	\$ 474.835	\$ 73.822
Diseño	\$ 1.085.817	\$ 1.285.703	\$ 199.886
Construcción	\$ 3.259.169	\$ 3.859.142	\$ 599.973
Pruebas	\$ 252.777	\$ 299.310	\$ 46.533
Despliegue y documentación	\$ 126.388	\$ 149.655	\$ 23.267
<b>Totales</b>	<b>\$ 6.155.420</b>	<b>\$ 7.288.558</b>	<b>\$ 1.133.137</b>

De esto se puede concluir que las desviaciones que se presentan en las estimaciones que se realizan con un enfoque determinístico generan pérdidas y desajustes en el calendario de ejecución del proyecto.

#### **4.1.7. Análisis del proceso de asignación de tareas de desarrollo**

La asignación de las tareas de desarrollo se realiza teniendo en cuenta únicamente la disponibilidad de los desarrolladores, la complejidad de la tarea, la demanda de las horas de desarrollo con respecto a todos los proyectos que se están ejecutando, y si se especifica, la prioridad de la tarea. Sin embargo, esta manera de asignar las tareas presenta distintos inconvenientes dado que esta asignación se hace semanalmente de forma manual y no hace parte de una planeación a mediano o corto plazo.

Adicionalmente, la selección del candidato más adecuado para realizar las tareas, se realiza con base a una evaluación empírica y no con una medición numérica de las capacidades ni

rendimientos de los desarrolladores, por lo tanto, genera una dependencia de la persona actualmente encargada del departamento, dificultad de si el volumen de tareas aumenta y también la posibilidad de asignar de manera incorrecta o no óptima un requerimiento. De tal manera, a igual que se planea la producción en una fábrica de manufactura, es necesario realizar una planificación periódica que permita tener en cuenta los tiempos de entrega y los recursos necesarios para cada proyecto.

## **5. Plan de mejora**

A partir de las oportunidades de mejora que se encontraron en el diagnóstico y análisis descrito anteriormente, se realiza un plan de mejora que incluye dos mejoras a los procesos identificados anteriormente, que son la estimación de tiempos y costos, y la asignación de tareas. Por lo tanto, esta sección se divide en estos dos criterios, donde cada uno tiene la explicación del problema a solucionar, la propuesta que se plantea, el plan de implementación y la implementación de la mejora propuesta descrita.

### **5.1. Mejora proceso de estimación de tiempos y costos**

#### ***5.1.1. Problema a solucionar***

En la fase precontractual se lleva a cabo una estimación realizada por una persona con base a los requerimientos definidos por el cliente, sin embargo, no hay una estandarización de los criterios mínimos necesarios para realizar una buena y clara definición de lo que el cliente solicita, además de que la estimación está sujeta a la experiencia de la persona asignada para la tarea. Por lo tanto, las desviaciones en tiempos y costos de un proyecto son significativos además de que ocurren con bastante frecuencia.

#### ***5.1.2. Propuesta de mejora para el proceso de estimación de tiempos y costos***

Con el objetivo de mejorar el proceso de estimación de tiempos y por consiguiente de costos, se plantea la adaptación de una técnica de estimación de tiempos que permita realizar una

evaluación mucho más precisa y de manera más técnica. Una vez adaptada la técnica a los procedimientos de la empresa, se propone la estandarización de los criterios mínimos que se deben tener en cuenta para realizar los análisis de requerimientos de manera completa y así posteriormente, hacer un desglose y discusión de las mismas. A partir de esto, se propone la extracción de los datos que se encuentran en JIRA Software, para realizar un análisis de distribuciones en Python. Luego, se formula el diseño de una base de datos la cual permita la expansión y crecimiento de la misma, con el objetivo de poder consolidar la información resultado del análisis hecho en Python. Después, se programa distintos procedimientos almacenados que facilitan y gestionan la integridad de los datos al momento de escribirlos en la base de datos. Finalmente, a través de Excel, se programa las consultas de los datos consolidados en la base de datos, la simulación de la duración de las tareas y la generación de la información referente a la duración total de las tareas, costos unitarios y totales.

### 5.1.3. Plan de implementación de la mejora propuesta

**Tabla 14.**

*Plan de implementación para la mejora propuesta para estimación de tiempos y costos*

<b>Actividad</b>	<b>Recursos</b>	<b>Responsable</b>	<b>Duración</b>
Elaboración matriz de técnicas de estimación	Marco teórico del proyecto. Diagnóstico realizado.	Autor del proyecto.	3 días
Clasificación de las técnicas de estimación	Matriz resultado del punto anterior	Autor del proyecto Líderes de proyecto Gerente general	2 horas
Selección técnica de estimación	Matriz resultado del punto anterior	Autor del proyecto	1 hora

Estandarización y adaptación técnica de estimación a los procedimientos de la empresa	Matriz resultado del punto anterior	Autor del proyecto	5 días
Extracción de los datos con ayuda de Python	Análisis de los datos consolidados en JIRA	Autor del proyecto	2 semanas
Análisis de los datos extraídos de JIRA	Datos extraídos del punto anterior	Autor del proyecto	2 días
Diseño de la base de datos	Diagnóstico realizado en proyecto	Autor del proyecto Gerente general	1 día
Elaboración procedimientos almacenados	Resultado diseño de base de datos del punto anterior	Autor del proyecto	2 días
Elaboración tabla de costos unitarios por rol	Diagnóstico realizado Reunión recursos humanos	Autor del proyecto	1 día
Programación de la herramienta	Diseño de la base de datos Procedimientos almacenados Resultado del análisis de los datos de las tareas	Autor del proyecto	1 semana
Elaboración de informe con probabilidad de tiempos de ejecución	Resultado punto anterior	Autor del proyecto	1 semana
Elaboración tabla de costos totales con desviaciones	Tabla resultado punto anterior Datos resultados de la simulación hecha con los datos del punto de análisis de datos	Autor del proyecto	1 hora

#### ***5.1.4. Implementación del plan de mejora para el proceso de estimación de tiempos y costos***

#### 5.1.4.1. Elaboración de la matriz y selección de la técnica de estimación

Con el objetivo de facilitar la adaptación de los procesos de la empresa a la técnica de estimación de tiempos y evitar posibles resistencias del personal en la transición, se seleccionó una técnica que representara una ligera modificación a la manera conservando la mejora sustancial. De tal manera, los principales criterios definidos en conjunto con líderes de proyecto y el gerente general para seleccionar la técnica fueron: que no implicara una reestructuración de la técnica actual de estimación, facilidad para implementar y facilidad de integración con el marco de trabajo SCRUM. Teniendo estos criterios en cuenta, se elaboró una matriz que calificaba con un puntaje de 1 a 5, siendo 1 el puntaje más bajo y 5 el puntaje más alto, las opciones investigadas.

**Tabla 15.**

*Matriz de evaluación de técnicas de estimación*

Técnica	Criterio			TOTAL
	No reestructuración	Facilidad de implementación	Facilidad de integración	
Modelo Putnam o SLIM	1	1	3	5
COCOMO	1	1	2	5
Método Delphi	4	5	5	14
Bottom-up	5	5	4	14
Planning Póker	1	5	4	10
Razonamiento basado en casos	5	5	5	15
Redes neuronales	4	2	4	10
Regresión estándar	4	3	4	11
Regresión robusta	3	2	4	9

Puntos de función	1	1	1	3
Puntos de casos de uso	1	1	1	3

Del resultado de la matriz se puede encontrar que las opciones Razonamiento basado en casos, Bottom-up y Método Delphi fueron las que mayor presentan puntaje debido a que se integran naturalmente con el procedimiento que se lleva a cabo actualmente, son técnicas naturales del marco de trabajo SCRUM, son relativamente fáciles de implementar y no requieren gran tiempo de adaptación para el personal. Por consiguiente, con ayuda de distintos líderes de proyecto y el gerente general, se decidió crear una metodología que tomara los mejores aspectos de cada una de las técnicas y crear una nueva a partir de las 3.

En primera medida, para estandarizar la definición de requerimientos en una fase precontractual y establecer unos criterios mínimos bajo los cuales la parte comercial y los analistas de información puedan trabajar, con base en entrevistas informales con los líderes de proyectos de la empresa, se plantea la siguiente matriz la cual contiene distintos puntos que, a partir de la experiencia, se deben tener claros para poder hacer estimaciones acertadas sobre las tareas a realizar y la complejidad de las mismas.

### Tabla 16.

*Matriz de definición de criterios mínimos de un requerimiento*

Criterio	Cumplimiento		
	Si	No	Otro
¿El requerimiento es nuevo o hace parte de una serie de mejoras a algo anteriormente desarrollado?			
Si no es nuevo, ¿afecta arquitectura del sistema?			

¿Ya se está capturando la información o toca crear tablas nuevas?

¿Requiere integración con un sistema de datos interno?

Si requiere integración, ¿se tienen claras las tablas y campos a usar?

¿Cuántos actores intervienen en el proceso? (Requiere aprobación, revisión, etc.)

¿Se requiere apoyo técnico por parte del cliente?

¿El diseño de la pantalla requiere un diseño especial?

¿Se basa en consultas a varias tablas?

¿Se tienen claras las entradas y salidas del proceso?

¿Se necesita crear un documento a partir del resultado?

¿Posee gráficas?

¿Cuántas pantallas se necesitan?

¿Cuántos procedimientos debe realizar?

---

Nota: Esta tabla se generó a partir de entrevistas informales con varios líderes de proyecto y trabajadores que realizan estimaciones de requerimientos dentro de la empresa.

A partir de esta matriz se puede reducir el riesgo de la falta de claridad en la definición, generando de esta manera un mayor entendimiento sobre las características del requerimiento y, por lo tanto, cuando se avance en la etapa de estimación no se presenten tantas dudas y se pueda estimar las tareas que se necesitan.

Posteriormente, una vez definido con claridad el requerimiento, se procede a utilizar la técnica de Bottom-up, la cual consta del fraccionamiento de cada componente donde la persona encargada o un experto técnico estima un tiempo y costo, y posteriormente, se suma todas las tareas necesarias para cumplir con el requerimiento, dando de esta manera, un resultado total de todo lo que se requiere. (Rashid, Nisar, Mahmood, Rehman, & Arafat, 2019)

Luego de tener una serie de tareas definidas que se necesitan para cumplir en su totalidad cada requerimiento, la persona encargada pasa a hacer una evaluación de la complejidad de cada tarea necesaria para cumplir el requerimiento. De esta manera, se plantea la siguiente tabla con una serie de ítems que fueron recolectados en entrevistas informales con desarrolladores, líderes de proyectos y personas encargadas de realizar la estimación de los requerimientos.

**Tabla 17.**

*Estimación de complejidad de las tareas*

N.º	Criterio	Cumplimiento		
		Si	No	Otro Puntaje
1	Se deben crear campos nuevos. ¿Cuántos?			
2	Se deben crear nuevas tablas. ¿Cuántas?			
3	¿Son ventanas nuevas o ya existentes?			
4	¿La pantalla se sale del estándar de K2BTools?			
5	¿Toca crear gráficas?			
6	¿La consulta se basa en varias tablas?			
7	¿El procedimiento cuando se realiza o modifica un registro involucra varias tablas?			
8	¿El procedimiento requiere hacer consultas a otras KBs (Knowledge Base)?			
9	¿Se requiere comprobar la información con bases de datos?			
10	¿Se ha realizado anteriormente el algún proyecto?			
<b>TOTAL</b>				

Nota: Esta tabla se generó a partir de entrevistas informales con varios líderes de proyecto, desarrolladores y analistas que realizan estimaciones de tareas dentro de la empresa.

Una vez realizada llenada la tabla se continúa a asignarle un puntaje a cada ítem dependiendo de la complejidad que representa para el desarrollo de la tarea en una escala de 1 a 3, por lo tanto, se tienen las siguientes series de condiciones.

**Tabla 18.**

*Instrucciones para los puntajes de cada tarea*

N.º	Instrucción
1	Si la cantidad de campos son menores a 5 el puntaje es 1. Si la cantidad de campos está entre 5 a 10 el puntaje es 2, y si la cantidad de mayor a 10 el puntaje es 3.
2	Si la cantidad de tablas es 1, el puntaje es 1. Si la cantidad es 2 o 3 el puntaje 2, y si es mayor a 3, el puntaje es 3.
3	Si la ventana es nueva o la modificación es leve, el puntaje es 1. Si es una modificación moderada el puntaje es 2, y si es una modificación mayor el puntaje es 3.
4	Si la pantalla no se sale del estándar de K2B el puntaje es 1. Si se sale del estándar de K2B levemente el puntaje es 2, y si se sale completamente del estándar el puntaje es 3.
5	Si las gráficas propuestas son las predeterminadas por Genexus, el puntaje es 1. Si a las gráficas toca hacerles modificaciones leves, el puntaje es 2, y si las gráficas toca construirlas a mano o tienen modificaciones mayores el puntaje es 3.
6	Si la consulta se basa en 1 tabla, el puntaje es 1. Si la consulta se basa en 2 o 3 tablas, el puntaje es 2, y si la tabla se basa en más de 3 tablas, el puntaje es 3.
7	Si se involucra 1 tabla, el puntaje es 1. Si se involucran 2 o 3 tablas, el puntaje es 2, y si se involucran más de 3 tablas, el puntaje es 3.
8	Si no quiere hacer consultas a otras KB, el puntaje es 1. De lo contrario, el puntaje es 3.
9	Si no se requiere comprobar información en base de datos, el puntaje es 1. De lo contrario, el puntaje es 3.
10	Si se ha realizado en otros proyectos el puntaje es 1, si no se ha realizado el puntaje es 3.

Nota: Esta tabla se generó a partir de entrevistas informales con varios líderes de proyecto, desarrolladores y analistas que realizan estimaciones de tareas dentro de la empresa.

Después de asignados los puntajes a cada tarea, se continúa totalizando el puntaje para cada tarea definida para los requerimientos, donde se tiene la siguiente tabla de clasificación de las tareas.

**Tabla 19.**

*Clasificación de las complejidades de las tareas*

<b>Complejidad</b>	<b>Rango</b>
Baja	1 a 10 puntos
Media	De 11 a 20 puntos
Alta	De 21 a 30 puntos
Muy alta	Tiene 30 puntos y faltan criterios

Nota: Esta tabla se generó a partir de entrevistas informales con varios líderes de proyecto, desarrolladores y analistas que realizan estimaciones de tareas dentro de la empresa.

Una vez asignados los puntajes y las complejidades de las tareas, se procede a utilizar el método Delphi mediante el cual, se le envía una copia de la descripción de los requerimientos definidos a una persona o dos, dependiendo de los recursos destinados o la complejidad y tamaño del proyecto, y este realiza el mismo procedimiento. De esta manera, se busca otra perspectiva sobre la complejidad de los requerimientos del proyecto.

Luego de que se tengan 2 o 3 estimaciones de los requerimientos del proyecto, se procede a utilizar el método de razonamiento basado en casos, el cual, a partir del histórico de las tareas realizadas en proyectos anteriores, busca tareas que les asignó la misma complejidad y asume que el tiempo empleado para una tarea similar o de la misma complejidad, será el mismo. Sin embargo, para realizar un análisis que tenga en cuenta las posibles variaciones, se realiza una simulación de Montecarlo, la cual, a partir de la agrupación de las tareas de las mismas complejidades, extrae

unos estadísticos, que indican el tipo de distribución de las tareas de determinada complejidad y realizan la simulación de las duraciones, dando como resultado unos rangos de valores, que dependiendo del nivel de confianza que se desee, se puede dar una fecha, duración y costo estimado.

Para realizar esta parte de la estimación se requiere automatizar la conexión entre la herramienta JIRA, donde se registran todos los proyectos con sus respectivos requerimientos tareas y, además, se encuentra todo el corazón de la operación de la empresa día a día, el análisis de las variaciones de las complejidades de las tareas y finalmente dar un resultado. Por lo tanto, para describir a detalle cómo se realiza esta parte final, se hablará sobre las siguientes etapas por separado: análisis de los datos contenidos en JIRA, diseño de la base de datos necesaria para consolidar los datos de manera estructurada, programación de la herramienta en Excel con VBA y finalmente la visualización de la información en tableros.

#### **5.1.4.2. Análisis de los datos contenidos en JIRA**

JIRA Software es una herramienta en línea que ayuda a la administración y gestión de todas las tareas de un proyecto de desarrollo de software. En esta se pueden registrar todos los requerimientos identificados en un proyecto, vincular las tareas asignadas, gestionar los casos de prueba o QA Tests, errores o *bugs* identificados, el versionamiento de los requerimientos, planeación de sprints, entre otras. Debido a su gran utilidad es de amplio uso en la industria de desarrollo de software, y particularmente, en la empresa se utiliza como principal herramienta para gestionar los proyectos.

Adicionalmente, dado su tamaño y reputación a nivel internacional, JIRA tiene la ventaja de que cuenta con un API REST, la cual a grandes rasgos es una interfaz de programación de aplicaciones que se ajusta a los principios REST que sirve para conectar cualquier web, app o script con el back-end o la lógica de negocio que hay detrás de una aplicación, en otras palabras, cuenta con un servicio que facilita el envío de solicitud de información mediante una estructura ya establecida y devuelve la información en un formato, que el caso de JIRA, es JSON. Esta API

REST, facilita la automatización de información debido a que constantemente se le pueden hacer solicitudes de información para procesar los datos y consolidarlos o transformarlos a voluntad.

Para mayor entendimiento de la herramienta a nivel gráfico, se muestran las figuras 8, 9, 10 y 11, las cuales ejemplifican en qué parte se ubica cada dato de un proyecto.

## Figura 10

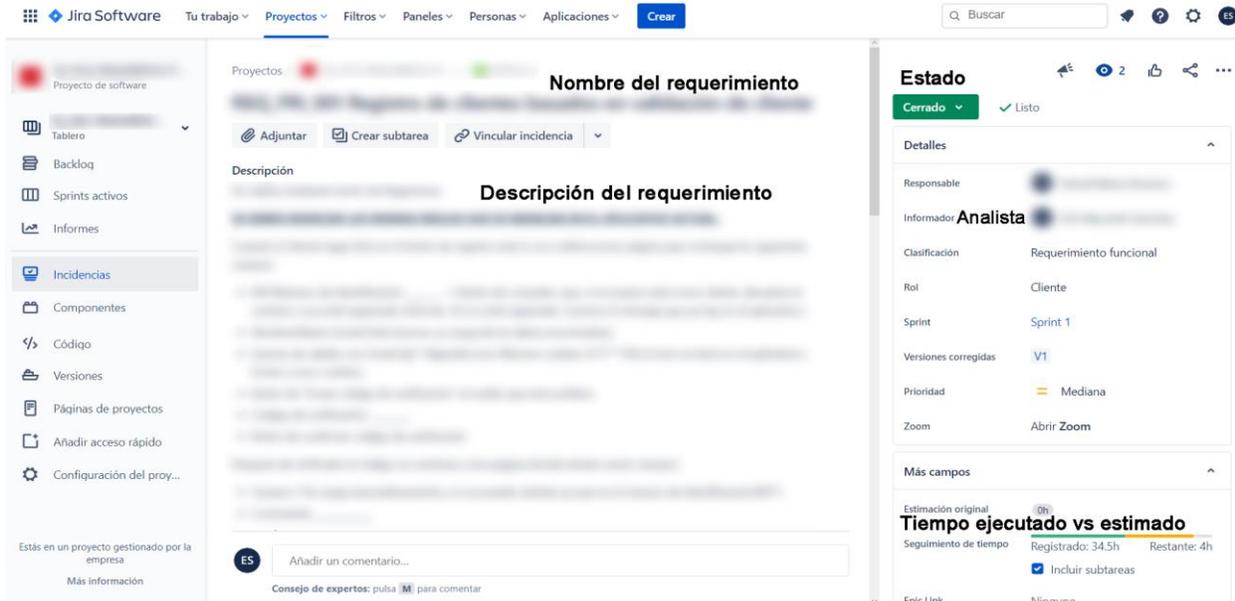
### *Backlog de un proyecto de desarrollo de software en JIRA*

The screenshot shows the JIRA project backlog interface. The top navigation bar includes 'Jira Software', 'Tu trabajo', 'Proyectos', 'Filtros', 'Paneles', 'Personas', 'Aplicaciones', and a 'Crear' button. A search bar is located on the right. The main content area is titled 'Proyectos' and 'Nombre proyecto'. Below this, there is a 'Backlog' section with a search bar and 'Filtros rápidos'. The 'Sprint' section is highlighted with a red box, showing 'Sprint 1' with '22 incidencias' and a date range from '15/sep/20 2:09 PM' to '13/oct/20 2:09 PM'. Below the sprint section, there is a table titled 'Lista de requerimientos' with columns for version (VI), status, and priority. The table contains several rows of data, with the first row highlighted in blue.

Nota: Los textos se encuentran censurados debido a que la empresa solicitó la censura de la información encontrada en las capturas debido a su carácter de confidencialidad.

**Figura 11**

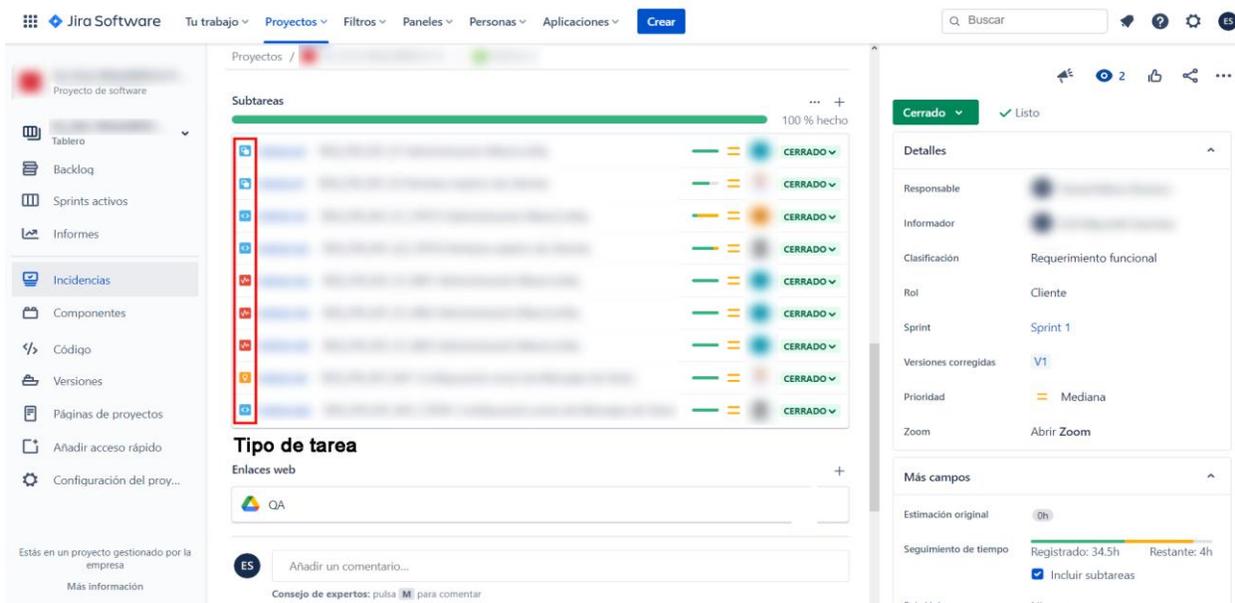
*Dashboard de un requerimiento*



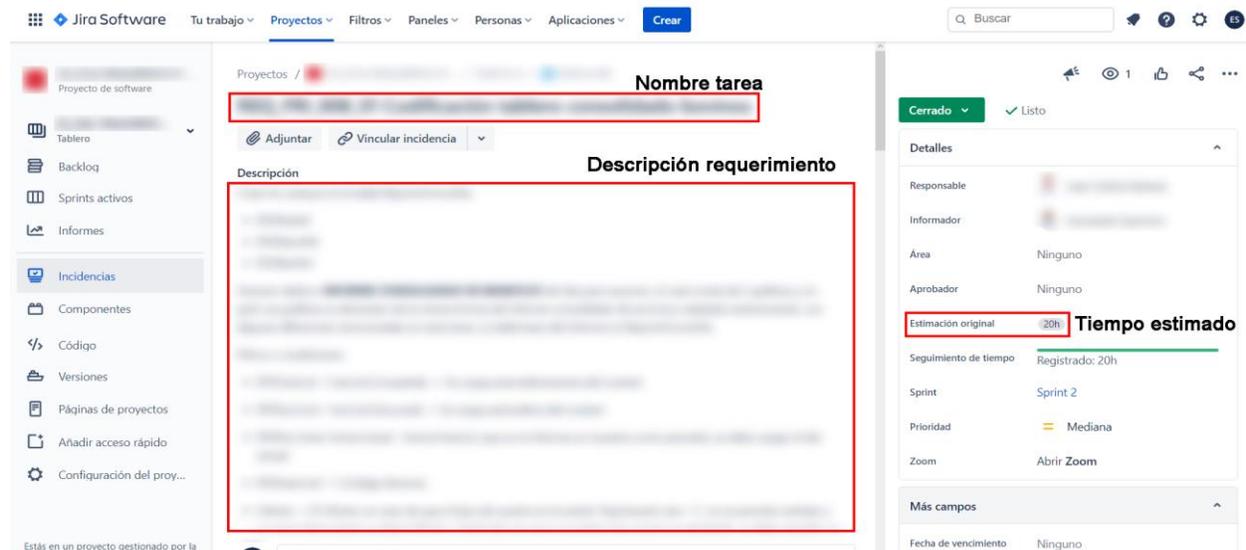
Nota: Los textos se encuentran censurados debido a que la empresa solicitó la censura de la información encontrada en las capturas debido a su carácter de confidencialidad.

**Figura 12**

*Lista de tareas de un requerimiento*



Nota: Los textos se encuentran censurados debido a que la empresa solicitó la censura de la información encontrada en las capturas debido a su carácter de confidencialidad.

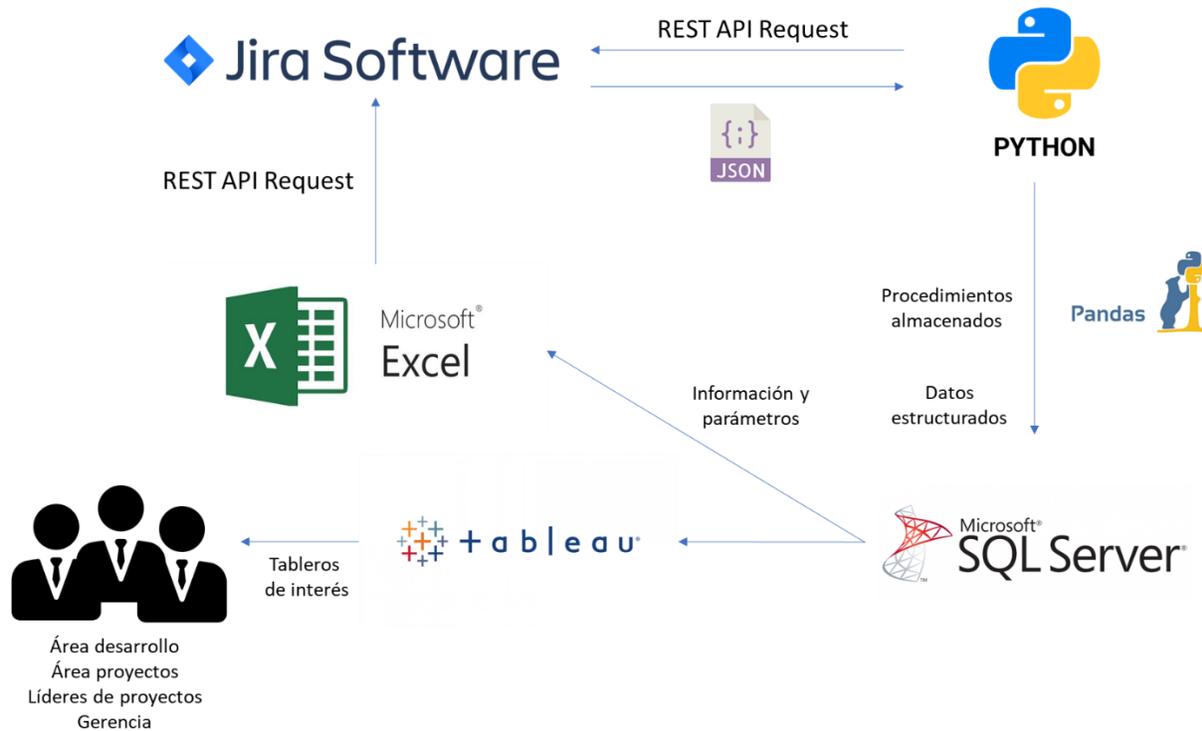
**Figura 13***Dashboard de una tarea*

Nota: Los textos se encuentran censurados debido a que la empresa solicitó la censura de la información encontrada en las capturas debido a su carácter de confidencialidad.

Como se puede observar, en la herramienta de JIRA se deposita toda la información del negocio distribuidas en muchas partes dependiendo de qué se necesite observar. Sin embargo, la dispersión de datos que se encuentra a través del portal dificulta el análisis y la consolidación de la información de manera manual, sin mencionar que se convierte en algo tedioso y demorado. Por lo tanto, para automatizar la extracción de datos se hace uso de Python, el cual permite hacer las solicitudes de información, procesamiento de los datos y la consolidación de los mismos en una base de datos posteriormente. La serie de pasos del algoritmo que debe realizar se muestra en la siguiente figura.

**Figura 14**

*Proceso iterativo que se realiza para procesar la información*



Según el diagrama, todo se deriva desde la solicitud de información a JIRA, la cual se realiza mediante el siguiente código, el cual, a partir de unos parámetros que permite enviar la API REST, se envía la solicitud y este devuelve un archivo JSON.

**Figura 15**

*Función creada en Python para hacer consultas a la API REST*

```
def consulta_jira_api(usuario: str,
                    api_key: str,
                    inicio=0,
                    jql='order by key asc',
                    maximo=100,
                    changelog=False,
                    fields=True,
                    detalle_campos=''):
    headers = {
        'Content-Type': 'application/json',
    }

    if changelog == True and fields == False:
        params = (
            ('expand', 'changelog'),
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
            ('fields', detalle_campos)
        )
    elif changelog == False and fields:
        params = (
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
        )
    elif changelog == True and fields == True:
        params = (
            ('expand', 'changelog'),
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
        )
    elif changelog == True and detalle_campos != '':
        params = (
            ('expand', 'changelog'),
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
            ('fields', detalle_campos),
        )
    elif changelog == False and detalle_campos != '':
        params = (
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
            ('fields', detalle_campos),
        )
    elif changelog == False and fields == False:
        params = (
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
            ('fields', detalle_campos),
        )
    else:
        params = (
            ('jql', jql),
            ('startAt', inicio),
            ('maxResults', maximo),
        )

    response = requests.get('https://scitech.atlassian.net/rest/api/3/search', headers=headers,
                           params=params, auth=(usuario, api_key))

    json_data = response.json()

    return json_data
```

Una vez recibida la información se procesa mediante distintas funciones que leen el archivo JSON, lo analizan y consolidan la información en objetos tipo DataFrames, los cuales posteriormente se almacenarán en la base de datos diseñada. Por lo tanto, dentro del análisis de los datos que se ejecutan, se realiza una clasificación de las tareas que se tienen registradas en JIRA con el objetivo de poder generar un análisis de distribución y poder realizar posteriormente la simulación de Montecarlo.

De tal manera, para clasificar en la misma escala las tareas, analizar las distribuciones, y poder realizar la simulación del tiempo se necesitó asignarles una clasificación a las tareas antiguas de manera paramétrica dado que revisar más de 20.000 tareas, una por una, sería una labor larga y tediosa. Por consiguiente, dado que antiguamente no se clasificaban los requerimientos según su complejidad, sino que directamente se les asignaba un tiempo estimado, en reunión con distintos líderes de proyecto y desarrolladores se llegó a la decisión de que se clasificaría las tareas en función del tiempo asignado. Los intervalos de decidieron de la siguiente manera.

**Tabla 20.**

*Clasificación de requerimientos antiguos*

<b>Complejidad</b>	<b>Intervalo</b>
Baja	De 1 a 5 horas
Media	De 6 a 10 horas
Alta	De 11 a 15 horas
Muy alta	De 16 horas en adelante

Esta escala permite analizar de manera similar los requerimientos y ver las distribuciones que tienen los datos según su complejidad. Así pues, una vez clasificadas las tareas antiguas y descartados los datos atípicos; donde se utilizó como criterio que si un dato era menor que la diferencia entre el cuartil uno menos 1,5 veces el rango intercuartil o si era mayor a la suma del cuartil tres mas 1,5 veces el rango intercuartil, dado que el 90% de los casos se debía a una

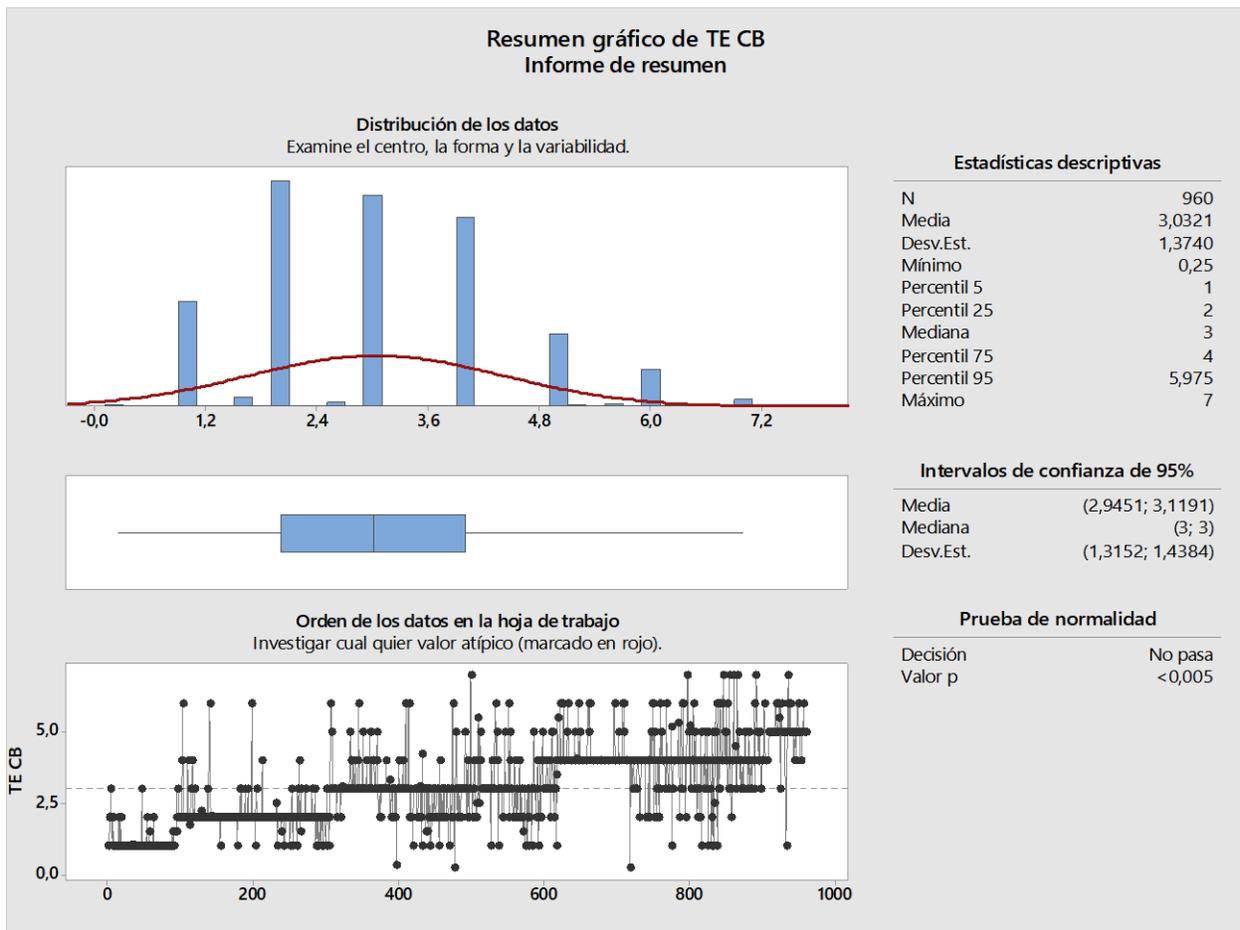
estimación realizada por una persona nueva o sin experiencia; se procedió a hacer análisis de las distribuciones de los datos de cada complejidad mediante Minitab.

#### 5.1.4.2.1. Datos de complejidad baja

Una vez realizado el análisis y descarte de los datos atípicos se pasó a analizar la distribución de los datos en Minitab a través del Resumen gráfico del Asistente, el cual arrojó la siguiente información (Figura 15).

**Figura 16**

*Análisis de la distribución de los datos de complejidad baja*



Como se puede concluir del reporte, los datos no se ajustan a una distribución normal, sin embargo, como el objetivo de la herramienta y la mejora de la estimación es dar valores de referencia para que a partir de los mismos se puedan tomar decisiones, más que brindar datos exactos con alta rigurosidad científica, por análisis visual se puede observar que los datos se distribuyen de manera similar a una gráfica de distribución Gamma. Además, debido a las características de un requerimiento de complejidad baja, hay una mayor probabilidad de que estas tareas duren una cantidad de tiempo menor, y la probabilidad de que duren más es más baja. Por lo tanto, para corroborar que los datos mantengan cierta semejanza, se procedió a hallar Alpha y Beta con las ecuaciones (5) y (6) y a partir de los estadísticos calculados en el análisis anterior y se realizó una simulación con 1000 datos aleatorios con una distribución gamma. Se muestran los resultados en la Figura 16.

Tenemos que

$$\mu = \alpha * \beta \quad (1)$$

$$\sigma^2 = \alpha * \beta^2 \quad (2)$$

Despejando  $\beta$  en (1), se tiene (3)

$$\beta = \frac{\mu}{\alpha} \quad (3)$$

Reemplazando (3) en un solo  $\beta$  de (2)

$$\sigma^2 = \alpha * \frac{\mu}{\alpha} * \beta \quad (4)$$

Despejando  $\beta$  de (4)

$$\beta = \frac{\sigma^2}{\mu} \quad (5)$$

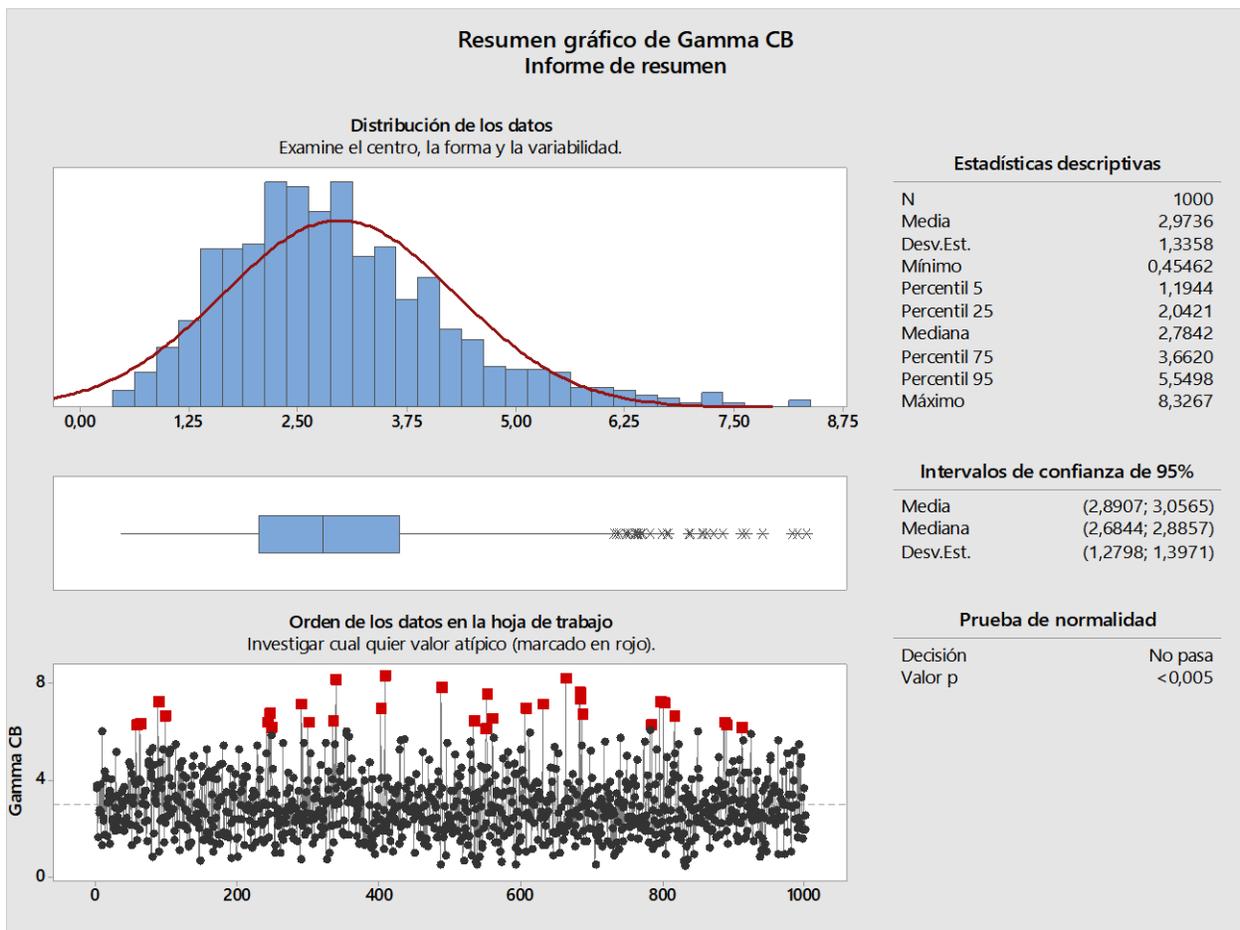
Luego despejando  $\alpha$  en (1)

$$\alpha = \frac{\mu}{\beta} \quad (6)$$

Una vez hallados los valores, se procedió a realizar la simulación.

**Figura 17**

*Simulación con distribución gamma de los datos de complejidad baja*



Como resultado del análisis, se puede concluir que, a pesar de que no se sigan pruebas de bondad de ajuste que prueben o denieguen el ajuste de los datos al conjunto total, la

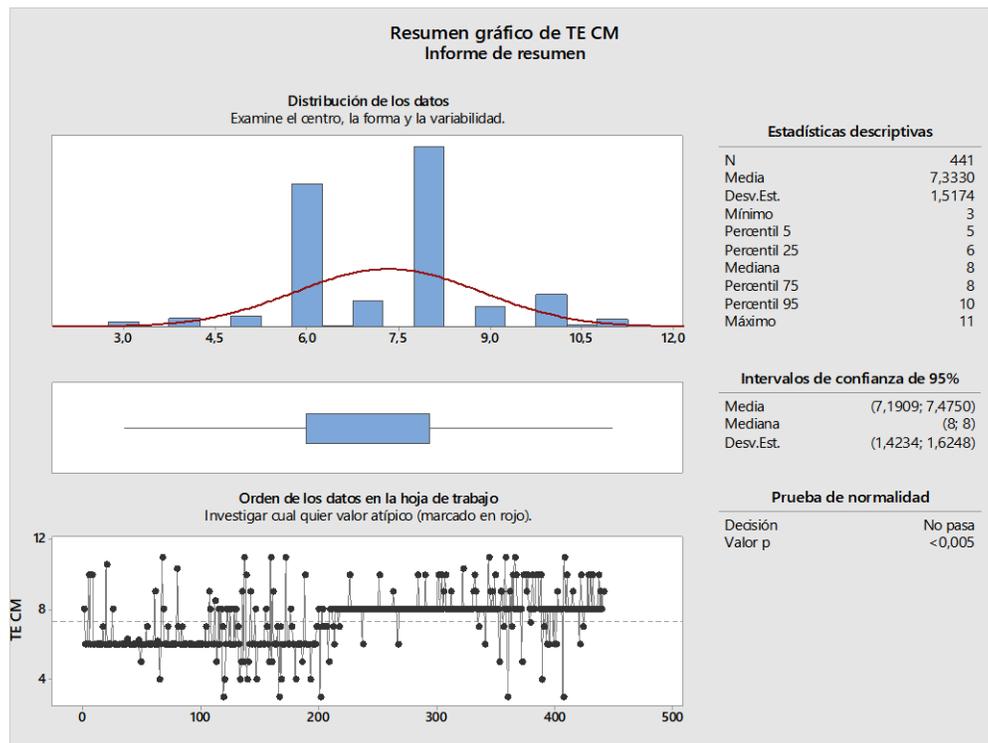
utilización de la distribución Gamma para realizar simulaciones de las posibles duraciones de tiempo de las tareas de complejidad baja es bastante práctica dado que con un análisis no estadístico y empírico se puede observar que esta distribución funciona para describir el conjunto de datos. Adicionalmente, dado que lo que se quiere es brindar valores de referencia más que datos exactos no representa un factor de riesgo para la compañía la utilización de la distribución Gamma.

#### 5.1.4.2.2. Datos de complejidad media

Al igual que se realizó en los datos de complejidad baja, se descartaron los valores atípicos usando el mismo criterio y se procedió a realizar el análisis de los datos en Minitab.

### Figura 18

#### Análisis de la distribución de los datos de complejidad media

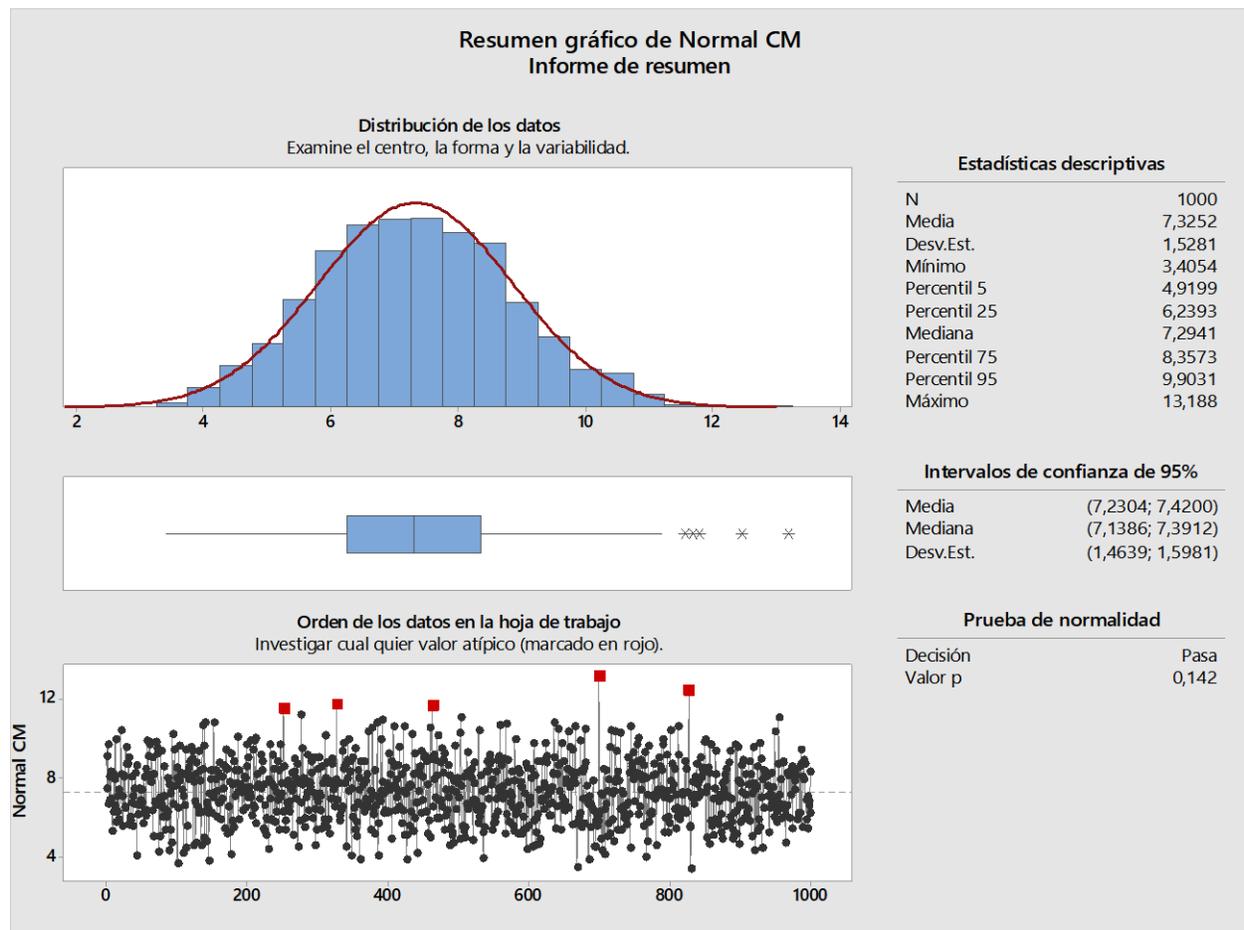


Para los datos de complejidad media, se puede observar que difícilmente estos se ajustan a una distribución normal, sin embargo, siguiendo el mismo criterio de los datos de complejidad

baja, y analizando las características de un requerimiento de complejidad media, se decide realizar una simulación de 1000 datos con una distribución normal, dado que frecuentemente un requerimiento de complejidad media tiene la misma probabilidad de emplear más o emplear menos tiempo, adicionalmente el objetivo es dar un dato de referencia, de tal manera que sirva no como un valor preciso.

### Figura 19

*Simulación con distribución normal de los datos de complejidad media*



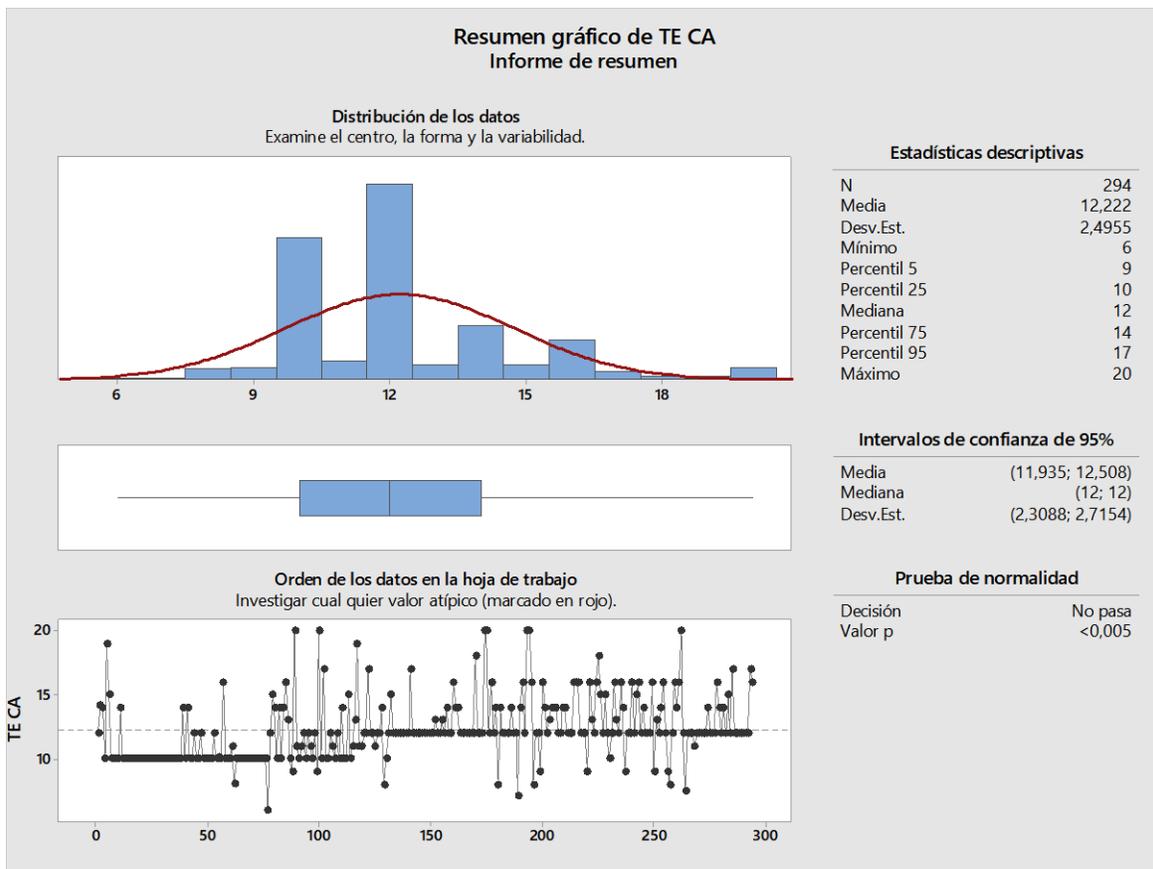
Como conclusión, se puede definir que, para los requerimientos de complejidad media, se usará la distribución normal dado que estos al ser estimados, pueden presentar una complejidad mayor o menor casi en igual probabilidad, por lo tanto, usar este tipo de distribución se ajusta a las características del grupo y facilita la entrega de valores de referencia.

### 5.1.4.2.3. Datos de complejidad alta

Como se ha realizado anteriormente, el análisis de los valores atípicos se realizó descartando los datos que estuvieran por debajo de la diferencia del cuartil uno menos 1,5 veces el rango intercuartil, o que estuvieran por encima de la suma del cuartil tres más 1,5 veces el rango intercuartil. Al igual que en los anteriores análisis, se verificó que estos datos efectivamente pudiesen ser descartados, y como resultado de esta verificación, se encontró que estos valores atípicos correspondían a estimaciones realizadas por alguien inexperto o imprevistos en el desarrollo.

**Figura 20**

*Análisis de la distribución de los datos de complejidad alta*

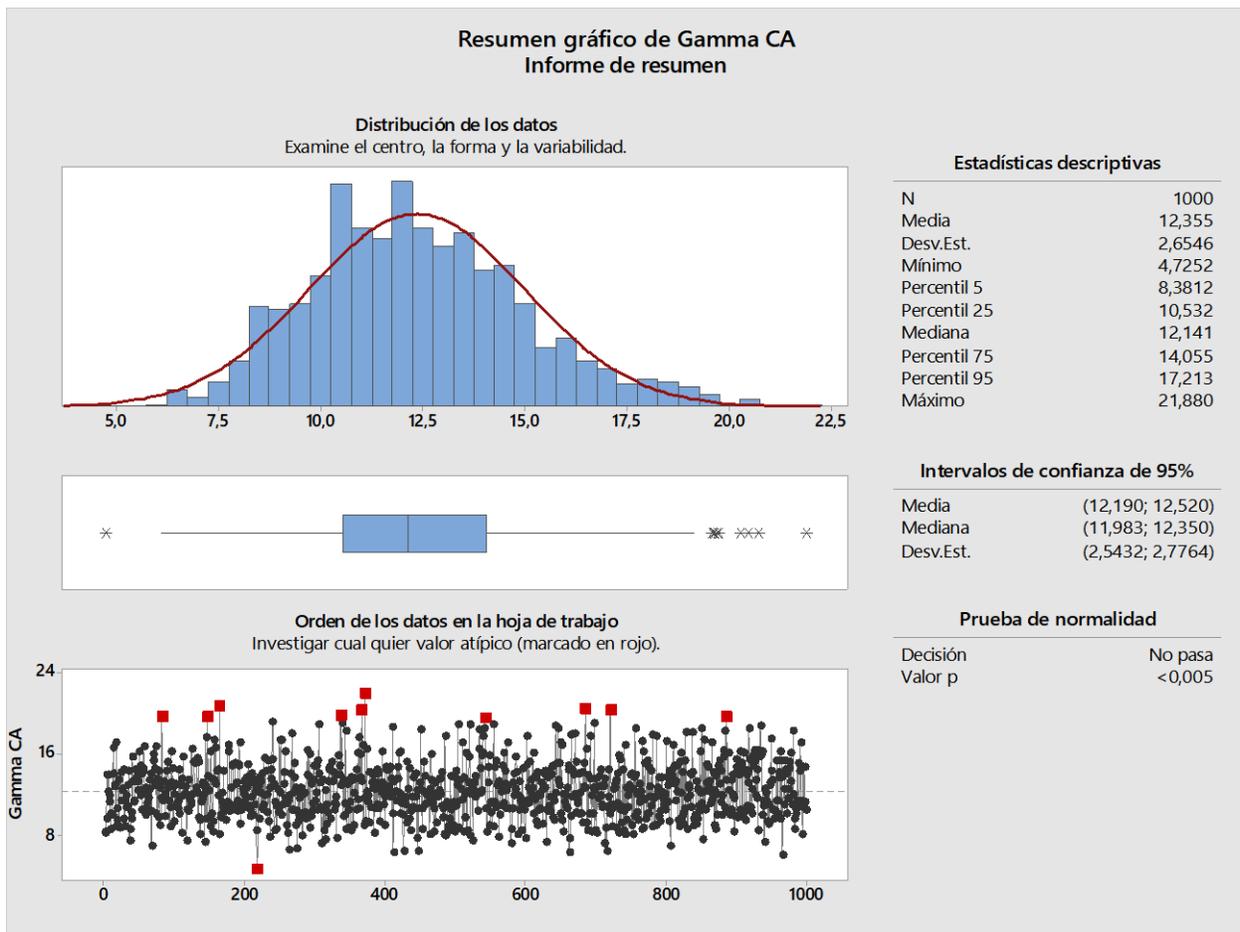


Como se observa en la Figura 19, los datos no se ajustan a una normal, sin embargo, teniendo en cuenta que, dadas las características de las tareas de complejidad alta, hay una mayor

probabilidad que estas duren entre 10 y 13 horas y los demás valores posean una menor probabilidad. Además, visualmente se puede ver cierta semejanza con la distribución Gamma, al igual que los requerimientos de complejidad baja, por lo tanto, se decidió realizar una simulación con 1000 datos con una distribución Gamma.

## Figura 21

*Simulación con distribución Gamma de los datos de complejidad alta*



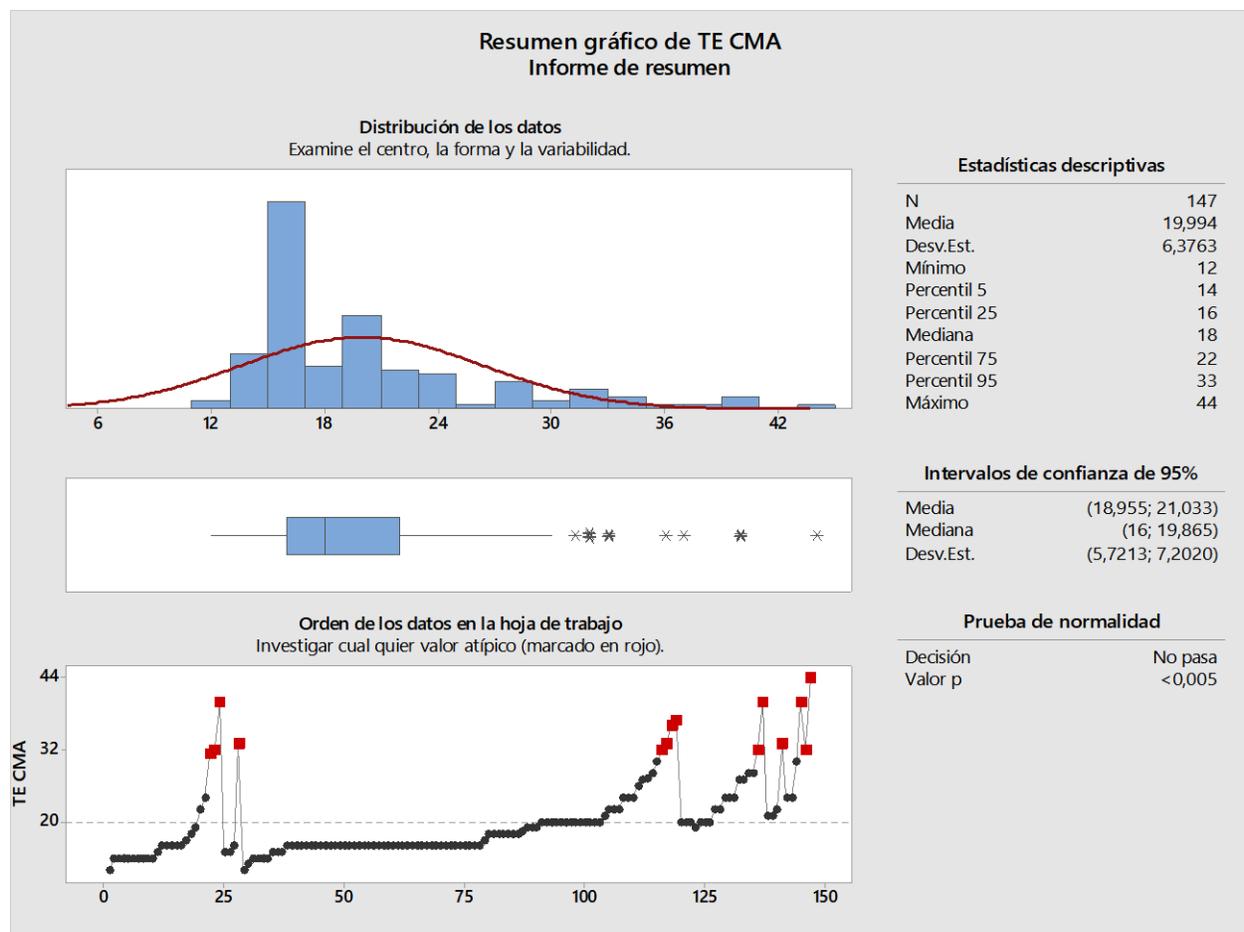
Como conclusión del análisis de los datos de complejidad alta, se puede decir que usar una distribución Gamma, puede servir como un punto de referencia dado que esta se ajusta al comportamiento de la distribución de las tareas.

#### 5.1.4.2.4. Datos de complejidad muy alta

Para el análisis de los datos de complejidad muy alta, no se hizo descarte de los valores atípicos, debido a que se consideró que estos valores tenían peso dada la dificultad que representa estimar un requerimiento que se sale de los estándares. Por lo tanto, ningún valor fue descartado y se procedió a hacer análisis con todos los datos extraídos.

### Figura 22

*Análisis de la distribución de los datos de complejidad muy alta*



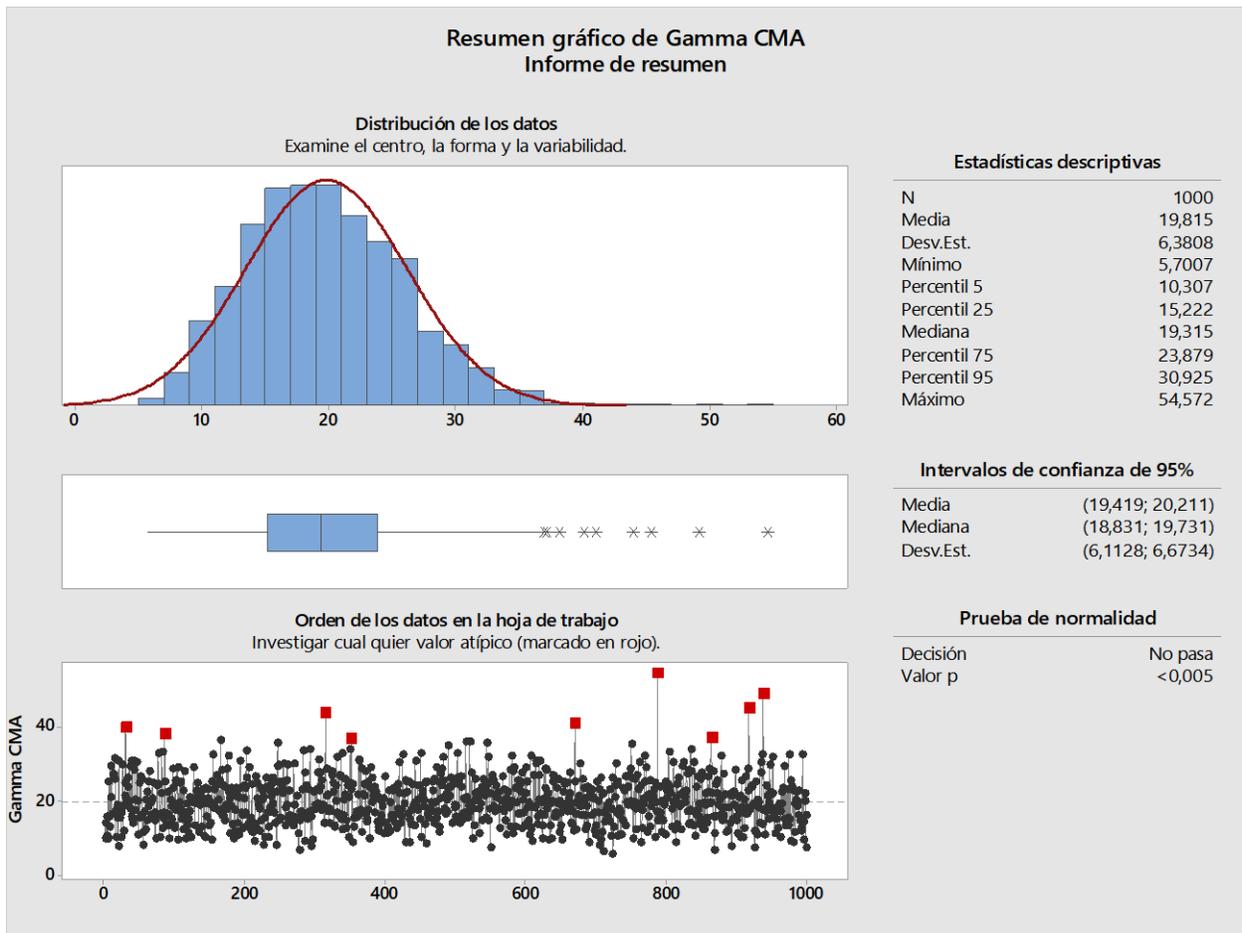
Nota: El orden de los datos es irrelevante para el análisis que se realizó dado que en la base de datos se encuentran ordenados por tiempo estimado.

Posterior al análisis, se decidió al igual que en las demás complejidades, realizar una simulación de 1000 datos con una distribución gamma, esto debido a que el comportamiento de

un requerimiento de este tipo suele tener una tendencia de asimetría a la derecha. Sin embargo, para la distribución se puede observar que el parámetro  $\beta = 2,0334$ , tiene un valor más alto, lo que indica que la cola tiene una mayor probabilidad, incluyendo en la distribución la posible desviación en las estimaciones.

**Figura 23**

*Simulación con distribución Gamma de los datos de complejidad muy alta*



Concluyendo el análisis de los datos de complejidad muy alta, se puede comentar que dentro de la distribución gamma, se tiene en cuenta que las tareas de esta clasificación puede representar una desviación alta, por lo tanto, esta distribución se ajusta bien al comportamiento de los datos y sirve para dar una serie de valores de referencia para realizar las simulaciones de Montecarlo.

### 5.1.4.3. Diseño de la base de datos

Con el objetivo de almacenar todos los datos que se encuentran depositados en JIRA, y guardar el análisis de datos que se realizó previamente, para poder realizar la solución de programación en Excel y mantener centralizados todos los datos de la empresa, se necesita consolidar todos estos datos en un solo repositorio, por consiguiente, se debe diseñar una base de datos que sea escalable y responda a las necesidades de la información actuales. De tal manera, se planteó un diseño que tuviera la tercera forma normal. Para esto se realizó una reunión con el Gerente General, el cual manifestaba que su principal interés era poder ver día a día la operación, dado que actualmente se realizaban comités de gerencia donde se analizaban las horas destinadas por proyecto, y desarrollador cada mes, lo cual para él era un rango de tiempo muy grande, por lo tanto, necesitaba poder ver la información cada día en cualquier momento. Adicionalmente, el diseño realizado se pensó para poder discriminar y realizar numerosos análisis posteriormente que le permitan al área de proyectos y de desarrollo, identificar requerimientos con base en sus características y estimar en promedio las duraciones, bugs por tarea, entre otras. Con base en ambos criterios, se plantearon las siguientes tablas.

La tabla Requerimientos se diseñó como la base y la tabla principal de todos los análisis que se pueden realizar. En ella, se encuentran los identificadores únicos de las tareas en JIRA que vinculan cada tipo de tarea a una serie de campos como proyecto y sprint al que pertenece, el tipo de tarea, si está contenida dentro de un requerimiento vincula la tarea a ese requerimiento, el nombre de la tarea o requerimiento, los responsables e informadores, la fecha que se creó y actualizó, el estado actual, el tiempo estimado, el tiempo empleado, fecha de entrega y la complejidad de la tarea. Esta tabla mediante las constantes ejecuciones del script de Python (Ver apéndice A. Código fuente Python), se estarán actualizando, reflejando de esta manera el estado actual de toda la operación.

**Tabla 21.***Diseño de datos de la tabla Requerimientos*

<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de dato</b>
jiraKey	Identificador único del requerimiento, tarea, bug o subtarea.	VARCHAR(30)
proyecto	Llave foránea del proyecto al que pertenece la tarea.	VARCHAR(30)
sprint	Nombre del sprint al que pertenece la tarea.	VARCHAR(100)
tipoTarea	Nombre del tipo de tarea.	VARCHAR(20)
padre	Identificador único del requerimiento al que pertenece la tarea.	VARCHAR(100)
nombre	Nombre del requerimiento o tarea.	VARCHAR(200)
informador	Llave foránea del usuario de JIRA el cual analizó o reportó la tarea.	VARCHAR(100)
responsable	Llave foránea del usuario de JIRA el cual es responsable por el desarrollo de la tarea o del requerimiento.	VARCHAR(100)
creada	Fecha en la cual se creó el requerimiento.	DATETIME
actualizada	Última fecha en la cual se presentó un movimiento de la tarea.	DATETIME
estado	Indicada el estado actual de la tarea, ya sea Nuevo, En desarrollo, Pruebas QA, Cerrado, o Pruebas UAT.	VARCHAR(20)
tiempo_estimado	Tiempo que se destinó para la tarea.	DECIMAL(6,3)
tiempo_empleado	Tiempo que se empleó desarrollando la tarea.	DECIMAL(6,3)
fecha_entrega	Fecha de entrega que se necesita la tarea.	
complejidad	Llave foránea de la complejidad asignada al requerimiento.	INT

La tabla Movimientos se diseñó con el objetivo de que registrar y mantuviera cada uno de los cambios de estados y datos que se realizaran en JIRA. De esta manera, se puede monitorear el movimiento que se tenga por proyecto o usuario, facilitando de esta manera la supervisión y control de los líderes de proyecto y su respectiva ejecución del proyecto.

**Tabla 22***Diseño de datos de la tabla Movimientos*

<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de dato</b>
idMovimiento	Llave primaria del movimiento referente a cada requerimiento.	VARCHAR(30)
idHistoria	Identificador único de cada movimiento en JIRA.	BIGINT
jiraKey	Llave foránea del identificador único del requerimiento, tarea, bug o subtarea.	VARCHAR(100)
autor_cambio	Llave foránea del usuario de JIRA el cual es responsable por el desarrollo de la tarea o del requerimiento.	VARCHAR(20)
modificada	Fecha de generación del movimiento realizado	DATETIME
campo_modificado	Estado o campo de JIRA modificado.	VARCHAR(50)
estado_anterior	Estado o dato anterior al cambio.	VARCHAR(100)
estado_posterior	Estado o dato posterior al cambio.	VARCHAR(100)
descripción	Descripción del cambio realizado.	VARCHAR(300)

La tabla de Roles se diseñó con el objetivo de que esta contuviese la lista de actores que se involucraban en un proyecto, facilitando de esta manera, mantener un control sobre qué función desempeña cada encargado en la tabla de Requerimientos y Movimientos.

**Tabla 23.***Diseño de datos de la tabla de Roles*

<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de dato</b>
idRol	Llave primaria que identifica únicamente a cada rol.	INT
rol	Nombre del rol que tiene asignado cada usuario dentro del sistema.	VARCHAR(59)
desRol	Descripción del rol que tiene asignado cada usuario dentro del sistema.	VARCHAR(300)

La tabla de UsuariosJIRA tiene como objetivo almacenar los identificadores únicos de cada usuario de JIRA para poder realizar la asignación directamente desde la herramienta ofimática. Adicionalmente, en esta se registra qué función cumple cada persona que tenga asignado un Requerimiento y el estado dentro de la empresa. Además, su función sirve para almacenar la clasificación que se le asigne a cada desarrollador en función de sus habilidades y desempeño, de esta manera se plantea más la mejora para la asignación de las tareas de programación.

**Tabla 24.***Diseño de datos de la tabla UsuariosJIRA*

<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de dato</b>
usuarioJIRA	Nombre del usuario de JIRA.	VARCHAR(100)
cuentaId	Id interno de JIRA para realizar la asignación de las tareas en JIRA.	VARCHAR(200)
rol	Llave foránea del rol asignado a la persona.	INT
estado	Estado activo/inactivo de la cuenta.	VARCHAR(10)
puntaje	Puntaje asignado a los roles de desarrolladores según su desempeño.	DECIMAL(4,2)

La tabla de Proyectos se diseñó con el objetivo básico de tener una lista de todos los proyectos que la empresa ha trabajado o se encuentran en ejecución. Esto es especialmente útil para garantizar la integridad de los datos con respecto a las demás tablas. Adicionalmente, en esta tabla se relaciona el identificador único que se tiene en JIRA con la información de ejecución de los proyectos en la empresa, añadiéndole un indicador que deja claro en qué etapa dentro del ciclo de vida de desarrollo de software se encuentra determinado proyecto.

**Tabla 25.**

*Diseño de datos de la tabla Proyectos*

Nombre	Descripción	Tipo de dato
proyectoKey	Llave primaria que identifica cada proyecto únicamente en JIRA.	VARCHAR(30)
nombreProyecto	Nombre del proyecto.	VARCHAR(200)
estado	Estado en ejecución, soporte o culminado del proyecto.	VARCHAR(10)

La tabla Complejidades se diseñó con el objetivo de almacenar la información resultado del análisis realizado anteriormente de las complejidades y distribuciones de cada requerimiento. De esta manera, se puede consultar fácil y rápidamente los datos que se usarán para realizar la simulación de Montecarlo.

**Tabla 26.**

*Diseño de datos de la tabla Complejidades*

Nombre	Descripción	Tipo de dato
idComplejidad	Llave primaria que identifica únicamente a cada complejidad.	INT
complejidad	Nombre de la complejidad.	VARCHAR(50)
promedio	Valor del promedio de la complejidad.	DECIMAL(8,6)

desviacionEst	Desviación estándar de la complejidad	DECIMAL(8,6)
tipoModelo	Nombre del tipo de modelo asignado a este tipo de complejidad.	VARCHAR(20)

La tabla AsignacionesSemanales se diseñó con el objetivo de mantener un registro de todas las asignaciones que se realizan semana a semana. Esta tabla almacena la información realizada en el punto de mejora para la asignación de tareas. Esta contiene campos que permiten determinar los rangos de asignación, la fecha creada y modificada, esto con el objetivo de más adelante presentar la información en tableros, y poder hacer auditoría sobre cuánto se modificó por última vez cada asignación.

### Tabla 27.

*Diseño de datos de la tabla AsignacionesSemanales*

Nombre	Descripción	Tipo de dato
idAsignacion	Llave primaria de cada asignación semanal.	INT
titulo	Título de la asignación.	'Asignación semanal desde el ' + CAST(fecha_desde AS VARCHAR(10)) + ' hasta el ' + CAST(fecha_hasta AS VARCHAR(10))
fecha_desde	Fecha de inicio de la asignación.	DATE
fecha_hasta	Fecha final de la asignación semanal.	DATE
creada	Fecha y hora de creación de la asignación semanal.	DATETIME
modificada	Fecha y hora de modificación de la tarea semanal.	DATETIME

La tabla de *DetalleAsignacionSemanal* se planteó con el objetivo de que esta almacenara la relación de las tareas que se asignaron en cada registro de *AsignacionesSemanales*, es decir, esta tabla mantiene el registro de que en la semana del 20 de septiembre de 2021 al 27 de septiembre de 2021 se asignaron determinadas tareas y adicionalmente registra qué persona fue asignada a la respectiva tarea.

**Tabla 28.**

*Diseño de datos de la tabla de DetalleAsignacionSemanal*

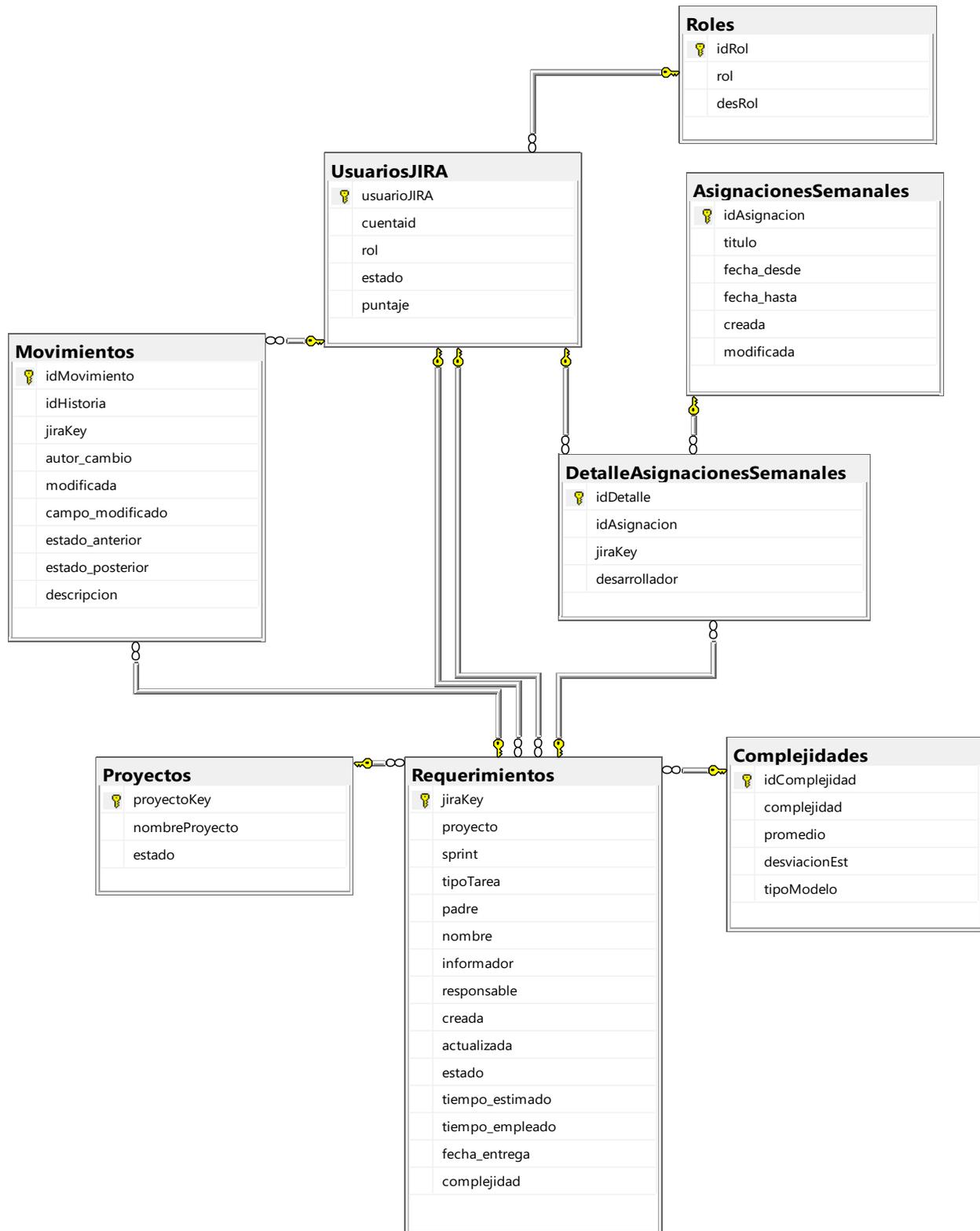
Nombre	Descripción	Tipo de dato
idDetalle	Identificador único de cada detalle de la asignación semanal.	INT
idAsignacion	Llave foránea de la asignación semanal.	INT
jiraKey	Llave foránea de la tarea asignada.	VARCHAR(30)
desarrollador	Llave foránea del usuario asignado para la tarea.	VARCHAR(100)

De tal manera, todo el diseño de datos se realizó con el objetivo de que permitiese un crecimiento y expansión sencilla sin que esta perdiese validez o necesitase una reestructuración total. Adicionalmente, para garantizar la rapidez de la ejecución del script de Python, se crearon procedimientos almacenados, los cuales se encargan de mantener la integridad de los datos y realizar la inserción de los datos desde Python (Ver Apéndice C. Procedimientos almacenados).

El resultado de las relaciones entre distintas tablas se puede observar en la Figura 24, la cual hace referencia al diagrama de entidad relación. Este diagrama permite ver las llaves foráneas y primarias de cada tabla que indican cómo se están entrelazando todos los datos almacenados en la base de datos.

**Figura 24**

*Diagrama de entidad relación*



Nota: Diagrama generado automática mediante SQL Server Management Studio.

#### 5.1.4.4. Elaboración costos unitarios por rol

A partir del diagnóstico realizado y los costos salariales mínimos que se presentan dentro de la empresa se plantea la siguiente tabla de costos unitarios por rol.

**Tabla 29.**

*Tabla de costos unitarios por hora por rol*

<b>Rol</b>	<b>Costo unitario / h</b>
Desarrollador	\$ 20.000
QA Tester	\$ 15.000
Gerente de proyecto	\$ 30.000
Arquitecto de software	\$ 35.000
Analista de software	\$ 20.000

Los valores de costo unitario se calcularon con base un promedio salarial del cargo dentro de la empresa.

#### 5.1.4.5. Programación herramienta en Excel e informes de simulaciones

Con base en el análisis de los datos realizados anteriormente, y la tabla de costos se procede a realizar la programación de la herramienta en Excel a partir de macros. Por lo tanto, para la primera parte de la herramienta se plantea la escritura manual de las tareas a voluntad de la persona que desee realizar una simulación. En esta tabla la persona debe llenar obligatoriamente el campo de complejidad y un nombre de la tarea para poder simular el registro, de lo contrario, el sistema no lo tomará.

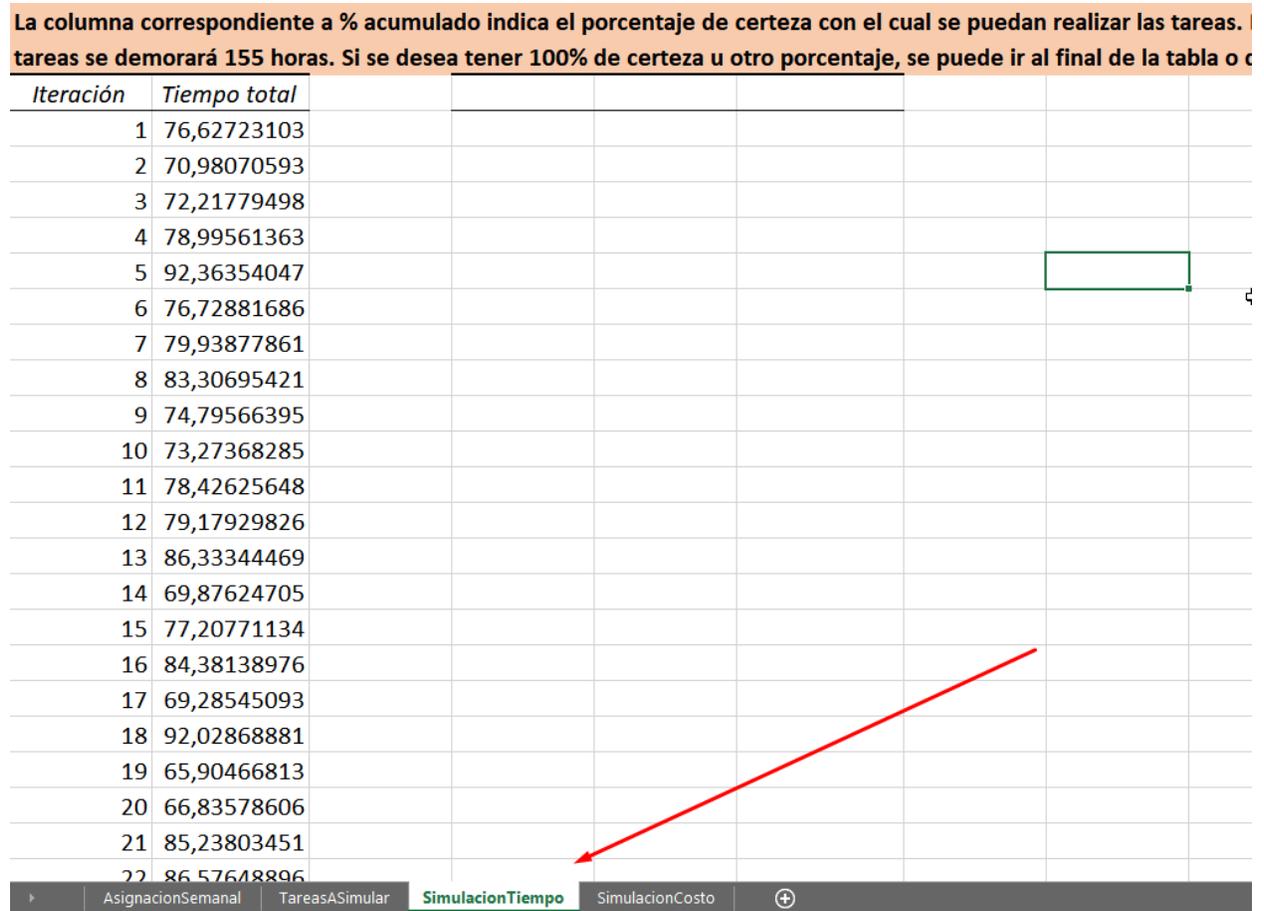




Luego, el sistema va registrando cada iteración en la hoja de SimulacionTiempo, para que a continuación se pueda realizar el análisis mediante la herramienta de Histograma de Excel, la cual genera una gráfica de frecuencia y frecuencia acumulativa,

**Figura 28**

*Hoja de informe de simulación del tiempo total estimado*

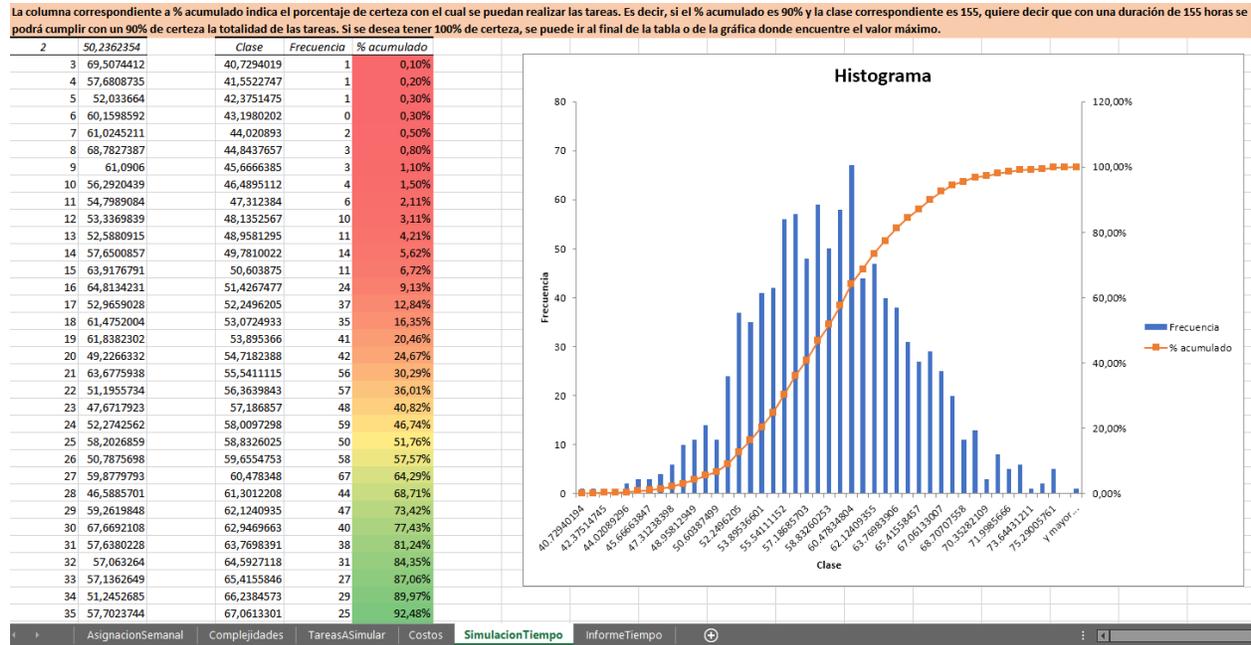


Igualmente, de manera paralela el sistema va registrando el total del costo estimado a partir de la tabla de costos unitarios anteriormente y la distribución de los distintos roles que toman parte en un proyecto que están registrados en la tabla costos.

Después de realizar las iteraciones, se redirige a la persona al informe de la simulación, donde este puede observar mediante un histograma y una tabla, la frecuencia y frecuencia acumulada de la cantidad de tiempo en total que se estimarían para la lista de requerimientos.

**Figura 29**

*Reporte de simulación del tiempo total*



Allí mediante un texto se le explica al usuario cómo realizar la interpretación de la frecuencia acumulada y visualmente se le indica mediante una escala de color, los porcentajes de cobertura de tiempo. De esta manera, se tiene unos valores de referencia los cuales facilitan la tarea de estimación de tiempos a partir de una simulación que tiene en cuenta las posibilidades de desviaciones en la ejecución del proyecto.

De igual manera para la simulación del costo posible, se va copiando el valor total del costo de la hoja TareasASimilar y se va pegando en la hoja SimulacionCosto con el número de iteración correspondiente. Adicionalmente, después de cada iteración los aleatorios van cambiando, lo que permite simular un nuevo costo,

**Figura 30***Hoja de informe de simulación del costo total estimado*

<i>Iteración</i>	<i>Costo total</i>						
1	\$ 2.928.208						
2	\$ 2.861.630						
3	\$ 2.981.958						
4	\$ 2.786.854						
5	\$ 3.009.436						
6	\$ 2.463.305						
7	\$ 2.620.138						
8	\$ 3.303.864						
9	\$ 2.882.294						
10	\$ 3.025.779						
11	\$ 2.931.097						
12	\$ 2.864.634						
13	\$ 2.601.850						
14	\$ 2.941.839						
15	\$ 2.890.507						
16	\$ 2.970.491						
17	\$ 2.659.779						
18	\$ 2.919.768						
19	\$ 2.855.034						
20	\$ 2.876.650						
21	\$ 3.228.305						
22	\$ 2.792.820						
23	\$ 2.522.822						
24	\$ 2.662.704						
25	\$ 2.828.050						
26	\$ 2.656.357						
27	\$ 2.771.582						
28	\$ 2.939.837						

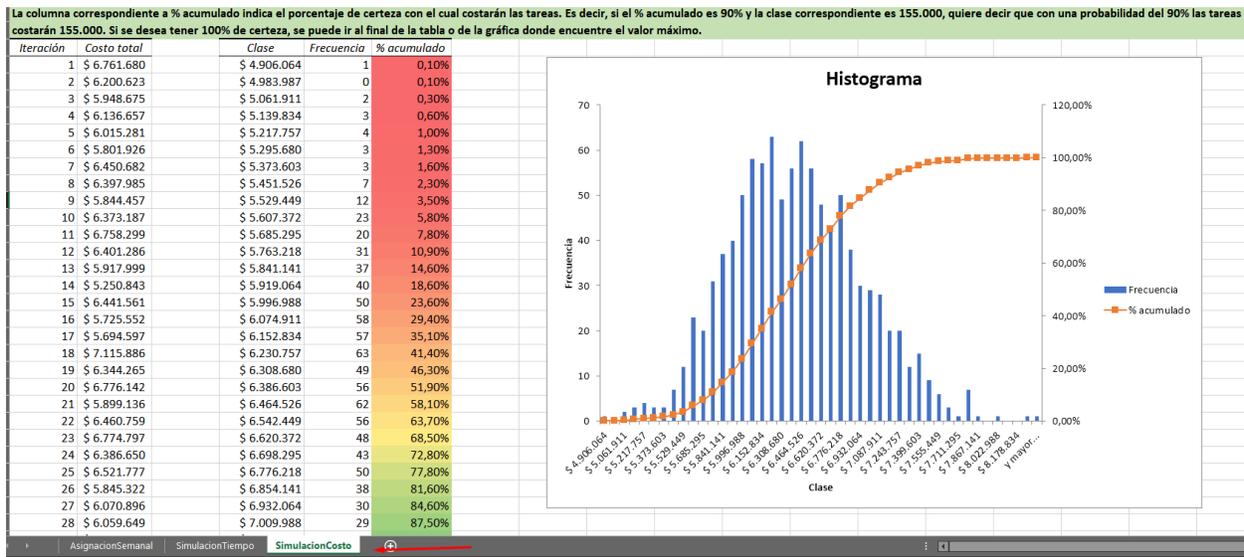


AsignacionSemanal | SimulacionTiempo | **SimulacionCosto** | +

Finalmente, mediante el histograma de Excel, se realiza un análisis de todas las simulaciones, y se presenta la información mediante una gráfica, una tabla que contiene las frecuencias y frecuencias acumuladas, además de un texto que explica cómo interpretar el gráfico.

**Figura 31**

*Informe de simulación del costo total*



**5.2. Plan de mejora para el proceso de asignación de tareas**

**5.2.1. Problema a solucionar**

Una vez el proyecto entra en ejecución las tareas no se asignan con base a un criterio que mida el desempeño de los desarrolladores, ni tampoco pertenece a una planeación a mediano o corto plazo, lo que genera que en algunos casos se asignen tareas muy complejas a desarrolladores que no poseen la experiencia o el conocimiento suficiente para realizarlas. Adicionalmente, la asignación se hace de forma manual y depende enteramente de la persona encargada del departamento.

**5.2.2. Propuesta de mejora para el proceso de estimación de asignación de tareas**

Con el fin de mejorar la asignación de tareas y convertirlo en un procedimiento más ágil, se plantea la medición de los desarrolladores según un criterio unificado el cual permita establecer una guía sobre la asignación de las tareas. También se propone una forma de asignar las tareas de

desarrollo con base a una planeación a corto plazo, lo que permita priorizar las tareas según su fecha de entrega y mejorar los tiempos de entrega de los requerimientos de todos los proyectos en ejecución.

### 5.2.3. Plan de implementación de la mejora propuesta

**Tabla 30.**

*Plan de implementación para la mejora propuesta para la asignación de tareas*

<b>Actividad</b>	<b>Recursos</b>	<b>Responsable</b>	<b>Duración</b>
Definición de criterios de desempeño de los desarrolladores	Diagnóstico inicial Identificación de los procesos	Autor del proyecto Coordinador de desarrollo	4 horas
Elaboración matriz de rendimiento de desarrolladores	Criterios resultados punto anterior	Autor del proyecto	1 día
Elaboración procedimiento de planeación de tareas para asignar	Diagnóstico inicial Identificación de los procesos Análisis de datos en JIRA	Autor del proyecto	2 días
Programación de la herramienta	Procedimiento de planeación de tareas Resultados de matriz de rendimiento de desarrolladores	Autor del proyecto	5 días

#### 5.2.4. *Implementación del plan de mejora para el proceso de asignación de tareas*

##### 5.2.4.1. **Definición de criterios de desempeño de los desarrolladores**

Con el objetivo de medir establecer una métrica sobre el posible rendimiento de los desarrolladores para mejorar la asignación de las tareas, se realizó una reunión con el coordinador de desarrollo en la cual se definieron una lista de criterios. Por lo tanto, los criterios fueron: años de experiencia con Genexus, años de experiencia con SQL y cantidad de lenguajes que ha manejado, dado que estas características fueron las que se consideró que más influían en el desempeño de los desarrolladores.

##### 5.2.4.2. **Elaboración matriz de desempeño**

Una vez establecidos los criterios se procedió a elaborar una matriz para la calificación y elaboración del puntaje asignado a cada desarrollador.

#### **Tabla 31.**

*Matriz de calificación desempeño*

<b>Criterio</b>	<b>Puntaje</b>
Años de experiencia con Genexus	
Años de experiencia con SQL	
Cantidad de lenguajes que ha manejado	

De tal manera, para establecer un puntaje para cada criterio se analizó de manera empírica con líderes de proyecto y con base a todos los desarrolladores y sus desempeños unos criterios que permitieran asignar puntajes.

Para El primer criterio de años de experiencia se estableció la puntuación que se puede ver en la **Tabla 29**.

**Tabla 32.**

*Puntajes para el criterio de año de experiencia con Genexus*

<b>Criterio</b>	<b>Puntaje</b>
De 0 a 2 años	1
De 3 a 5 años	2
De 5 años en adelante	3

Posteriormente, se estableció el criterio para los años de experiencia con SQL.

**Tabla 33.**

*Puntaje para el criterio de años de experiencia con SQL*

<b>Criterio</b>	<b>Puntaje</b>
De 0 a 1 año	1
De 2 a 3 años	2
De 3 años en adelante	3

Luego, se estableció el puntaje para el criterio de cantidad de lenguajes que conoce.

**Tabla 34.**

*Puntaje para el criterio de cantidad de lenguajes*

<b>Criterio</b>	<b>Puntaje</b>
De 1 a 2 lenguajes	1

De 3 a 4 lenguajes	2
De 4 lenguajes en adelante	3

Finalmente, se estableció un criterio que permitiese clasificar cada desarrollador según sus puntajes, de tal manera se elaboró la siguiente matriz la cual los agrupa en 3 clases, Desarrollador básico, Desarrollador junior y Desarrollador senior.

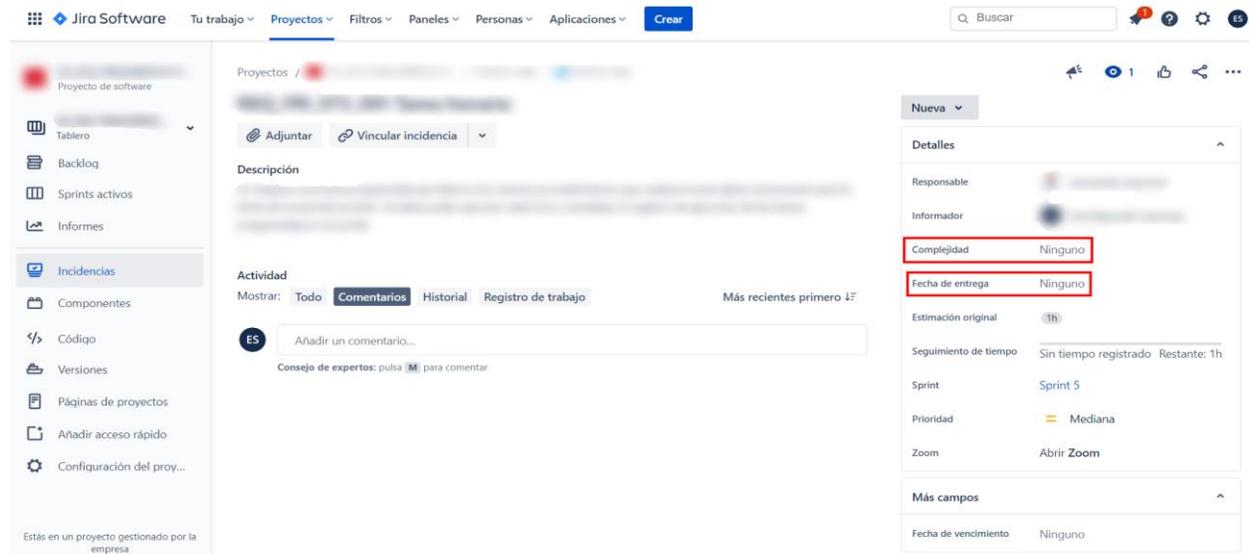
**Tabla 35.**

*Clasificaciones de los desarrolladores*

<b>Criterio</b>	<b>Puntaje</b>
Desarrollador básico	De 1 a 3
Desarrollador junior	De 3 a 6
Desarrollador senior	De 6 en adelante

#### **5.2.4.3. Elaboración del procedimiento de planeación de tareas para asignar**

Una vez registrados los puntajes de cada desarrollador en la tabla UsuariosJIRA y cada rol, se procede a realizar el procedimiento de planeación de las tareas para asignar. Para tal fin, se solicita que en cada tarea de desarrollo se tenga un campo en JIRA que permita registrar la fecha en la cual se proyecta la entrega de cada proyecto. Esta fecha tiene que ser planteada por cada líder de proyecto y se deja a criterio de cada líder dado que las características de planeación varían en función de cada proyecto.

**Figura 32.***Nuevo campo en las tareas de desarrollo*

Una vez incluido este campo y registrado en la base de datos, se procede a realizar la asignación de tareas de manera automática teniendo en cuenta el rendimiento de cada desarrollador y la fecha de entrega de la tarea.

#### 5.2.4.4. Programación de la herramienta

Inicialmente, se programaron 3 botones. El primer botón se llama Ver última asignación, la cual realiza una consulta a la base de datos donde extrae el detalle de la última asignación semanal y lo despliega en la hoja.

**Figura 33**

*Botón Ver última asignación*

**idAsignacion:** 1  
**Título:** Asignación semanal desde el 12/09/2021 hasta el 19/09/2021  
**Fecha desde:** 12/09/2021  
**Fecha hasta:** 19/09/2021  
**Creada:** 11/10/2021 5:31  
**Modificada:**

Ver última asignación Nueva asignación Modificar asignación Limpiar Simular

Automatizar asignación Realizar asignación semanal

proyecto	jiraKey	Título	Responsable	estado	fecha_entrega	actualizada	complejidad
PROYECTO1	PROYECTO1-26	TÍTULO TAREA DE DESARROLLO PROYECTO1-26	Desarrollador 10	Cerrado	8/10/2021	25/09/2021	Baja
PROYECTO2	PROYECTO2-39	TÍTULO TAREA DE DESARROLLO PROYECTO2-39	Desarrollador 8	Nueva	11/01/2021	28/12/2020	Baja
PROYECTO2	PROYECTO2-5	TÍTULO TAREA DE DESARROLLO PROYECTO2-5	Desarrollador 7	En progreso	6/10/2021	25/09/2021	Alta
PROYECTO2	PROYECTO2-36	TÍTULO TAREA DE DESARROLLO PROYECTO2-36	Desarrollador 3	Pruebas QA	6/10/2021	25/09/2021	Alta
PROYECTO2	PROYECTO2-88	TÍTULO TAREA DE DESARROLLO PROYECTO2-88	Desarrollador 3	En progreso	5/10/2021	25/09/2021	Baja
PROYECTO2	PROYECTO2-38	TÍTULO TAREA DE DESARROLLO PROYECTO2-38	Desarrollador 6	Nueva	26/05/2021	12/05/2021	Baja
PROYECTO2	PROYECTO2-77	TÍTULO TAREA DE DESARROLLO PROYECTO2-77	Desarrollador 6	En progreso	19/05/2021	12/05/2021	Baja
PROYECTO2	PROYECTO2-71	TÍTULO TAREA DE DESARROLLO PROYECTO2-71	Desarrollador 7	Pruebas QA	27/05/2021	14/05/2021	Baja
PROYECTO2	PROYECTO2-80	TÍTULO TAREA DE DESARROLLO PROYECTO2-80	Desarrollador 2	Pruebas QA	25/05/2021	19/05/2021	Baja
PROYECTO2	PROYECTO2-100	TÍTULO TAREA DE DESARROLLO PROYECTO2-100	Desarrollador 7	Pruebas QA	26/05/2021	24/05/2021	Baja
PROYECTO3	PROYECTO3-89	TÍTULO TAREA DE DESARROLLO PROYECTO3-89	Desarrollador 1	Pruebas QA	27/06/2021	18/06/2021	Baja
PROYECTO3	PROYECTO3-24	TÍTULO TAREA DE DESARROLLO PROYECTO3-24	Desarrollador 7	Nueva	2/07/2021	28/06/2021	Baja
PROYECTO4	PROYECTO4-19	TÍTULO TAREA DE DESARROLLO PROYECTO4-19	Desarrollador 5	En progreso	5/09/2021	24/08/2021	Media
PROYECTO3	PROYECTO3-67	TÍTULO TAREA DE DESARROLLO PROYECTO3-67	Desarrollador 7	En progreso	23/09/2021	13/09/2021	Baja
PROYECTO3	PROYECTO3-85	TÍTULO TAREA DE DESARROLLO PROYECTO3-85	Desarrollador 8	Nueva	28/09/2021	16/09/2021	Baja
PROYECTO3	PROYECTO3-31	TÍTULO TAREA DE DESARROLLO PROYECTO3-31	Desarrollador 9	Cerrado	29/09/2021	16/09/2021	Baja
PROYECTO3	PROYECTO3-73	TÍTULO TAREA DE DESARROLLO PROYECTO3-73	Desarrollador 9	Cerrado	18/10/2021	6/10/2021	Baja
PROYECTO3	PROYECTO3-30	TÍTULO TAREA DE DESARROLLO PROYECTO3-30	Desarrollador 2	Cerrado	12/10/2021	11/10/2021	Baja

Posteriormente, se programó el botón de Nueva Asignación, el cual se realiza una consulta a la base de datos de todas las tareas de desarrollo las cuales estén en estado Sin asignar y en estado Nuevo. De esta manera, se puede realizar una revisión de qué está pendiente por asignar.

**Figura 34**

*Botón Nueva asignación*

**idAsignacion:**  
**Título:**  
**Fecha desde:**  
**Fecha hasta:**  
**Creada:**  
**Modificada:**

Ver última asignación Nueva asignación Modificar asignación Limpiar Simular

Automatizar asignación Realizar asignación semanal

proyecto	jiraKey	Título	Responsable	estado	fecha_entrega	actualizada	complejidad
PROYECTO1	PROYECTO1-74	TÍTULO TAREA DE DESARROLLO PROYECTO1-74	Sin asignar	Nueva	6/10/2021	25/09/2021	Baja
PROYECTO2	PROYECTO2-79	TÍTULO TAREA DE DESARROLLO PROYECTO2-79	Sin asignar	Nueva	9/01/2021	28/12/2020	Baja
PROYECTO2	PROYECTO2-74	TÍTULO TAREA DE DESARROLLO PROYECTO2-74	Sin asignar	Nueva	5/10/2021	25/09/2021	Alta
PROYECTO2	PROYECTO2-78	TÍTULO TAREA DE DESARROLLO PROYECTO2-78	Sin asignar	Nueva	2/10/2021	25/09/2021	Alta
PROYECTO2	PROYECTO2-37	TÍTULO TAREA DE DESARROLLO PROYECTO2-37	Sin asignar	Nueva	27/09/2021	25/09/2021	Baja
PROYECTO2	PROYECTO2-63	TÍTULO TAREA DE DESARROLLO PROYECTO2-63	Sin asignar	Nueva	23/05/2021	12/05/2021	Baja
PROYECTO2	PROYECTO2-74	TÍTULO TAREA DE DESARROLLO PROYECTO2-74	Sin asignar	Nueva	26/05/2021	12/05/2021	Baja
PROYECTO2	PROYECTO2-76	TÍTULO TAREA DE DESARROLLO PROYECTO2-76	Sin asignar	Nueva	18/05/2021	14/05/2021	Baja
PROYECTO2	PROYECTO2-82	TÍTULO TAREA DE DESARROLLO PROYECTO2-82	Sin asignar	Nueva	3/06/2021	19/05/2021	Baja
PROYECTO2	PROYECTO2-76	TÍTULO TAREA DE DESARROLLO PROYECTO2-76	Sin asignar	Nueva	2/06/2021	24/05/2021	Baja
PROYECTO3	PROYECTO3-80	TÍTULO TAREA DE DESARROLLO PROYECTO3-80	Sin asignar	Nueva	30/06/2021	18/06/2021	Baja
PROYECTO3	PROYECTO3-66	TÍTULO TAREA DE DESARROLLO PROYECTO3-66	Sin asignar	Nueva	1/07/2021	28/06/2021	Baja
PROYECTO4	PROYECTO4-46	TÍTULO TAREA DE DESARROLLO PROYECTO4-46	Sin asignar	Nueva	27/08/2021	24/08/2021	Media
PROYECTO3	PROYECTO3-61	TÍTULO TAREA DE DESARROLLO PROYECTO3-61	Sin asignar	Nueva	25/09/2021	13/09/2021	Baja
PROYECTO3	PROYECTO3-78	TÍTULO TAREA DE DESARROLLO PROYECTO3-78	Sin asignar	Nueva	25/09/2021	16/09/2021	Baja
PROYECTO3	PROYECTO3-88	TÍTULO TAREA DE DESARROLLO PROYECTO3-88	Sin asignar	Nueva	22/09/2021	16/09/2021	Baja
PROYECTO3	PROYECTO3-38	TÍTULO TAREA DE DESARROLLO PROYECTO3-38	Sin asignar	Nueva	12/10/2021	6/10/2021	Baja
PROYECTO3	PROYECTO3-79	TÍTULO TAREA DE DESARROLLO PROYECTO3-79	Sin asignar	Nueva	24/10/2021	11/10/2021	Baja

Luego, se programó el botón Modificar asignación, el cual realiza una consulta a la base de datos donde muestra el historial de todas las asignaciones semanales, permite seleccionar una y modificarla.

**Figura 35**

*Botón Modificar asignación*

The screenshot shows a web application interface for task assignment management. On the left, there is a summary of the current assignment: **idAsignacion:** 1, **Título:** Asignación semanal desde el 12/09/2021 hasta el 19/09/2021, **Fecha desde:** 12/09/2021, **Fecha hasta:** 19/09/2021, **Creada:** 11/10/2021 5:31, and **Modificada:**. To the right of this summary are several buttons: 'Ver última asignación', 'Nueva asignación', 'Modificar asignación', 'Limpiar', and 'Simular'. Below these are two larger buttons: 'Automatizar asignación' and 'Realizar asignación semanal'. The main part of the interface is a table with columns 'proyecto', 'jiraKey', and 'Título'. A modal window titled 'Seleccione una asignación para modificar:' is open, displaying a table with columns 'id', 'Título', 'Desde', 'Hasta', and 'Creada'. The modal has 'Seleccionar' and 'Cancelar' buttons.

proyecto	jiraKey	Título
PROYECTO1	PROYECTO1-38	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-65	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-18	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-61	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-16	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-62	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-81	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-14	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-61	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO2	PROYECTO2-73	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-23	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-99	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO4	PROYECTO4-18	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-40	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-92	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-16	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-78	TÍTULO TAREA DE DESARROLLO PROYECT
PROYECTO3	PROYECTO3-81	TÍTULO TAREA DE DESARROLLO PROYECT

Después, se programó el botón de Automatizar asignación el cual ordena las tareas de desarrollado desde la que tiene fecha de entrega más cercana hasta la fecha más lejana. Luego de esto, empieza a realizar la asignación de cada tarea a partir del rendimiento de cada desarrollador, por lo tanto, si una tarea de desarrollo tiene complejidad Muy alta, se asigna solamente a un desarrollador senior, si tiene complejidad Alta se asigna a un desarrollador junior, o si tiene complejidad media o baja se le asigna a un desarrollador básico. Adicionalmente, dentro de la comprobación, se simula un posible tiempo de desarrollo para la tarea, de esta manera se tiene en cuenta posibles variaciones según su complejidad y además de eso, se tiene la condición de que la asignación semanal de cada desarrollador no supere las 48 horas laborales semanales, posibilitando de esta manera la asignación de distintas tareas a distintos desarrolladores según su disponibilidad de tiempo y desempeño.

Figura 36

Botón Automatizar asignación

idAsignacion:  
 Título  
 Fecha desde:  
 Fecha hasta:  
 Creada:  
 Modificada:

Ver última asignación   Nueva asignación   Modificar asignación   Limpiar   Simular

Automatizar asignación   Realizar asignación semanal

proyecto	jiraKey	Título	Responsable	estado	fecha_entrega	actualizada	complejidad
PROYECTO2	PROYECTO2-87	TÍTULO TAREA DE DESARROLLO PROYECTO2-87	Desarrollador 1	Nueva	21/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-64	TÍTULO TAREA DE DESARROLLO PROYECTO2-64	Desarrollador 1	Nueva	21/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-96	TÍTULO TAREA DE DESARROLLO PROYECTO2-96	Desarrollador 1	Nueva	21/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-73	TÍTULO TAREA DE DESARROLLO PROYECTO2-73	Desarrollador 1	Nueva	21/09/2021	17/09/2021	Media
PROYECTO3	PROYECTO3-20	TÍTULO TAREA DE DESARROLLO PROYECTO3-20	Desarrollador 1	Nueva	22/09/2021	17/09/2021	Alta
PROYECTO3	PROYECTO3-27	TÍTULO TAREA DE DESARROLLO PROYECTO3-27	Desarrollador 15	Nueva	22/09/2021	17/09/2021	Alta
PROYECTO3	PROYECTO3-21	TÍTULO TAREA DE DESARROLLO PROYECTO3-21	Desarrollador 15	Nueva	22/09/2021	17/09/2021	Alta
PROYECTO3	PROYECTO3-20	TÍTULO TAREA DE DESARROLLO PROYECTO3-20	Desarrollador 15	Nueva	22/09/2021	17/09/2021	Alta
PROYECTO3	PROYECTO3-80	TÍTULO TAREA DE DESARROLLO PROYECTO3-80	Desarrollador 15	Nueva	22/09/2021	18/09/2021	Media
PROYECTO3	PROYECTO3-28	TÍTULO TAREA DE DESARROLLO PROYECTO3-28	Desarrollador 17	Nueva	22/09/2021	18/09/2021	Media
PROYECTO3	PROYECTO3-6	TÍTULO TAREA DE DESARROLLO PROYECTO3-6	Desarrollador 17	Nueva	22/09/2021	18/09/2021	Media
PROYECTO3	PROYECTO3-6	TÍTULO TAREA DE DESARROLLO PROYECTO3-6	Desarrollador 17	Nueva	22/09/2021	18/09/2021	Media
PROYECTO2	PROYECTO2-100	TÍTULO TAREA DE DESARROLLO PROYECTO2-100	Desarrollador 17	Nueva	23/09/2021	17/09/2021	Alta
PROYECTO2	PROYECTO2-99	TÍTULO TAREA DE DESARROLLO PROYECTO2-99	Desarrollador 17	Nueva	23/09/2021	17/09/2021	Alta
PROYECTO2	PROYECTO2-41	TÍTULO TAREA DE DESARROLLO PROYECTO2-41	Desarrollador 2	Nueva	23/09/2021	17/09/2021	Alta
PROYECTO2	PROYECTO2-25	TÍTULO TAREA DE DESARROLLO PROYECTO2-25	Desarrollador 2	Nueva	23/09/2021	17/09/2021	Alta
PROYECTO1	PROYECTO1-50	TÍTULO TAREA DE DESARROLLO PROYECTO1-50	Desarrollador 1	Nueva	23/09/2021	19/09/2021	Baja
PROYECTO1	PROYECTO1-53	TÍTULO TAREA DE DESARROLLO PROYECTO1-53	Desarrollador 15	Nueva	23/09/2021	19/09/2021	Baja
PROYECTO1	PROYECTO1-21	TÍTULO TAREA DE DESARROLLO PROYECTO1-21	Desarrollador 2	Nueva	23/09/2021	19/09/2021	Baja
PROYECTO1	PROYECTO1-47	TÍTULO TAREA DE DESARROLLO PROYECTO1-47	Desarrollador 2	Nueva	23/09/2021	19/09/2021	Baja
PROYECTO2	PROYECTO2-90	TÍTULO TAREA DE DESARROLLO PROYECTO2-90	Desarrollador 2	Nueva	23/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-61	TÍTULO TAREA DE DESARROLLO PROYECTO2-61	Desarrollador 3	Nueva	23/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-68	TÍTULO TAREA DE DESARROLLO PROYECTO2-68	Desarrollador 3	Nueva	23/09/2021	17/09/2021	Media
PROYECTO2	PROYECTO2-27	TÍTULO TAREA DE DESARROLLO PROYECTO2-27	Desarrollador 3	Nueva	23/09/2021	17/09/2021	Media
PROYECTO3	PROYECTO3-63	TÍTULO TAREA DE DESARROLLO PROYECTO3-63	Desarrollador 1	Nueva	24/09/2021	18/09/2021	Baja
PROYECTO3	PROYECTO3-68	TÍTULO TAREA DE DESARROLLO PROYECTO3-68	Desarrollador 2	Nueva	24/09/2021	18/09/2021	Baja
PROYECTO3	PROYECTO3-84	TÍTULO TAREA DE DESARROLLO PROYECTO3-84	Desarrollador 3	Nueva	24/09/2021	18/09/2021	Baja
PROYECTO3	PROYECTO3-69	TÍTULO TAREA DE DESARROLLO PROYECTO3-69	Desarrollador 3	Nueva	24/09/2021	18/09/2021	Baja

AsignacionSemanal   TareasASimilar   SimulacionTiempo   SimulacionCosto   UsuariosJIRA

Finalmente, se programó el botón de Realizar asignación semanal, el cual se conecta a JIRA mediante una POST Request a la API REST y automática cambia el valor del campo de asignado para que posteriormente el script de Python lea el cambio en la asignación de la tarea y lo registre dentro de la base de datos, permitiendo así que se le haga seguimiento a todos los proyectos mediante un dashboard de Tableau.

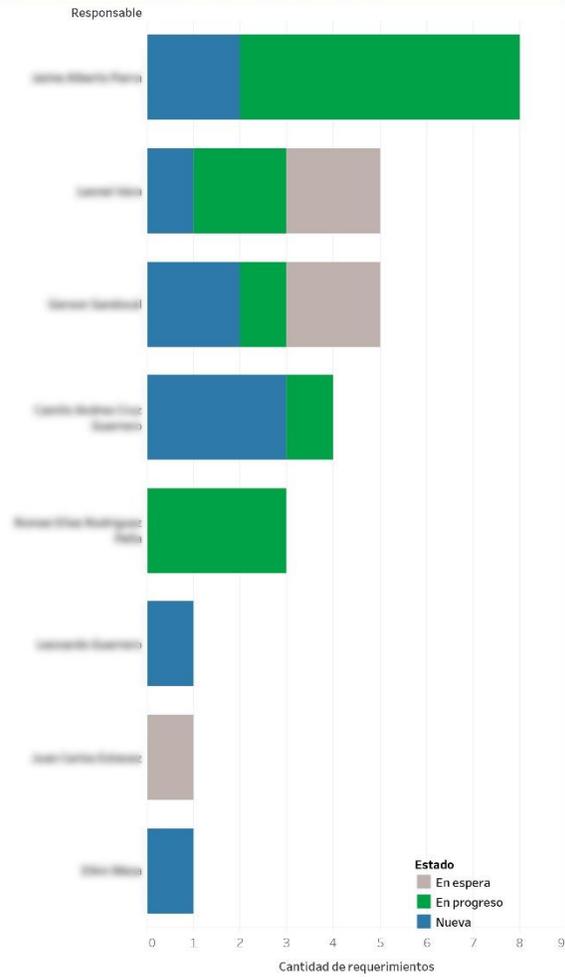
**Figura 37**

Tablero de gestión

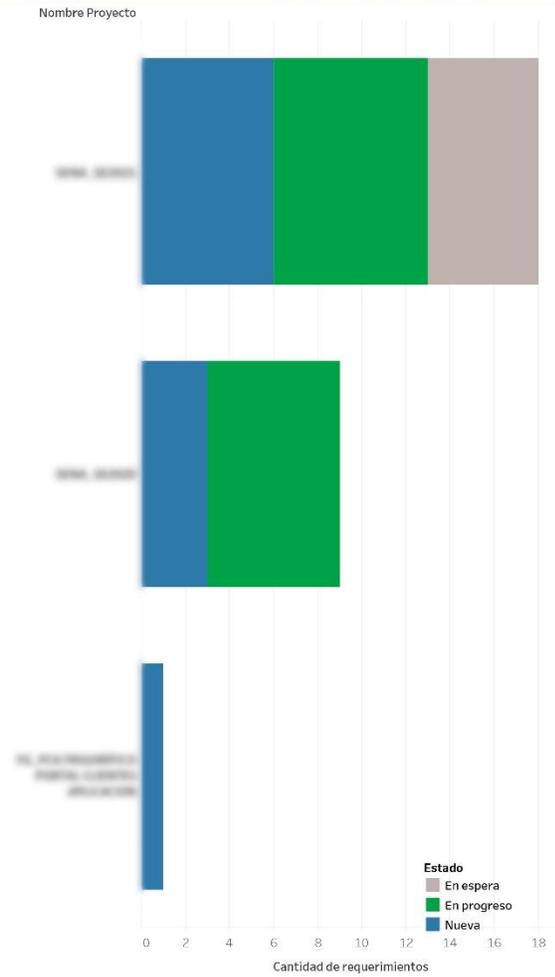


### TABLERO DE GESTIÓN

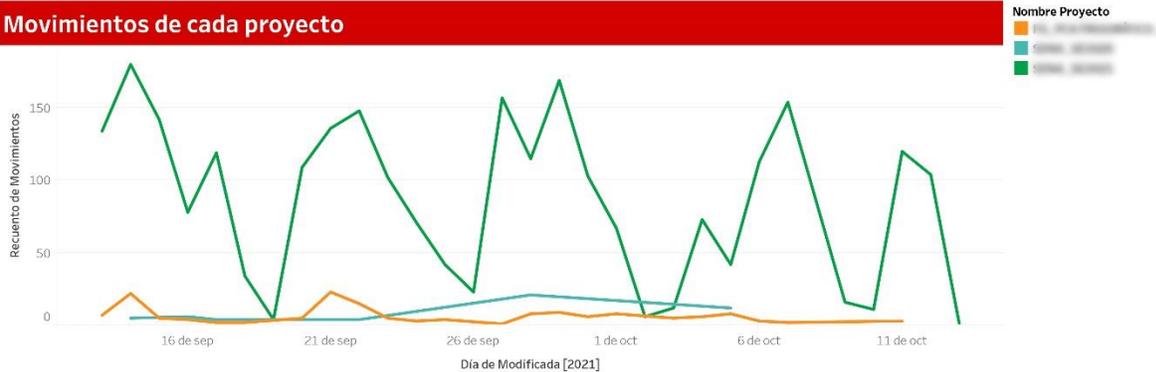
#### Estados de tareas por desarrollador



#### Estado de tareas de desarrollo por proyecto



#### Movimientos de cada proyecto



## 6. Conclusiones

- El diagnóstico realizado a los procedimientos de estimación de tiempos y costos permitió identificar las desviaciones que frecuentemente se presentan en los proyectos de desarrollo de software. A través de este análisis se permitió identificar que adicionalmente las tareas de desarrollo tenían una variación dependiendo de su dificultad, elemento el cual, no se tenía en cuenta, por lo tanto, se encontraron que los procesos a mejorar correspondían a la asignación de tareas y las estimaciones de tiempos y costos.

- Se formuló un plan de mejora que atendía las principales problemáticas que afectaban a los procesos identificados en el diagnóstico y con la ayuda del gerente, distintos líderes de proyecto y desarrolladores, se logró realizar las mejoras planteadas para cada proceso. Las mejoras hacen referencia a la adaptación de una técnica de estimación de tiempos, la cual requería una estandarización de criterios mínimos para estimar las tareas, un sistema de clasificación y discusión entre pares, el análisis de la información contenida en su herramienta de seguimiento JIRA, diseño de una base de datos para almacenar la información analizada y finalmente una herramienta en Excel que se conectara a la base de datos como a la herramienta en JIRA y permitiese simular la duración de las tareas de desarrollo en una fase precontractual o de ejecución y adicionalmente, mejorara la asignación de tareas basándose en el desempeño de los desarrolladores de la empresa.

- La implementación del plan de mejora permitió al coordinador de desarrollo agilizar y mejorar la asignación de las tareas de desarrollo, una prueba de esto, se ve en que según reporta el coordinador, el tiempo que le tomaba asignar las tareas oscilaba entre 3 y 4 horas, con la herramienta se redujo en un 75%, mejorando este proceso de manera sustancial. Adicionalmente, la simulación permitió que el área comercial tuviera en cuenta más los imprevistos, mejorando su cobertura de costos en un 10%.

- La centralización de los datos en una base de datos permitió que el gerente mantuviese un control mucho más claro sobre la ejecución de todos los proyectos, facilitando así la toma de decisiones de manera temprana y evitando retrasos en todos los proyectos.

- El diseño de datos implementado facilita la expansión y el mantenimiento de la base de datos, de tal manera, la posibilidad de implementar modelos de estimación más robustos como redes neuronales se puede realizar sin ningún problema.

- Finalmente, se realizó una reunión con gerencia donde se les explicaba y planteaba todas las mejoras implementadas, en la cual, todos se mostraron muy satisfechos con el rendimiento y resultado del proyecto.

## **7. Recomendaciones**

- Se recomienda el análisis constante de la complejidad de las tareas de manera que se esté constantemente revisando datos atípicos con el objetivo de ajustar los procesos si es necesario o modificar el modelo y la distribución asignada.

- Se recomienda realizar una evaluación semestral del rendimiento de los desarrolladores de manera que permita identificar la posibilidad de asignar tareas de mayor complejidad a los desarrolladores y así potenciar su desempeño.

- Se recomienda la implementación de modelos más robustos de estimación de tiempos y costos que tenga en cuenta variables como características del cliente, riesgos presentados, características del líder de proyecto, entre otras, de esta manera se tendría una estimación mucha más certeza sobre el tiempo de culminación de un proyecto y el costo asociado.

- Se recomienda la interacción más constante entre el departamento de proyectos y el coordinador de desarrollo, de manera que se le pueda hacer seguimiento a las tareas enviadas por cada líder de proyecto y de esa manera, establecer métricas que permitan identificar factores de riesgo en los proyectos, además de métricas de rendimiento de cada líder de proyecto.

- Se recomienda la implementación de que a partir de la centralización de los datos, cada líder de proyecto mantenga un formato estandarizado de seguimiento a los proyectos en Tableau, de manera que se pueda manejar indicadores de gestión claros y la toma de decisiones se facilite.

### Referencia

- Ampuero, M. A. (2015). *Un modelo para la asignación de recursos humanos a equipos de desarrollo de software*.
- Anderson, D. R., Sweeny, D. J., Williams, T. A., Camm, J. D., & Martin, K. (2011). *Métodos cuantitativos para los negocios* (Onceava ed.). México D.F: Cengage Learning.
- Cadavid, A. N., Martínez, J. D., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva. Una nueva visión para la ingeniería*, 30-39.
- Caruso, N. M., Castillo, E. P., & Navarro, K. S. (2015). *Mejora de los procesos de estimación de costos de software. Caso del sector de software de Barranquilla*.
- Cuevas, S. S., & Leal, A. C. (2014). Estandarización de los procesos asociados al desarrollo de proyectos informáticos: un caso de estudio. *Computación y Sistemas*. doi:<http://dx.doi.org/10.13053/CyS-18-2-2014-038>
- Hansen, D. R., & Mowen, M. M. (2006). *Administración de costos: Contabilidad y control*. (Quinta ed.). Santa Fé, México: CENGAGE Learning.
- Harrington, J. (1998). *Mejoramiento de los procesos de la empresa*. McGraw-Hill.
- Lawson, D., & Marion, G. (2008). *An Introduction to Mathematical Modelling*.
- Leitón, B. V., & Abrego, M. C. (2009). Modelo estocástico para estimación de valores económicos de rasgos productivos y funcionales en bovinos lecheros.
- Mercado-Caruso, N., Castillo, E. P., & Oliveira, H. P. (2017). Aplicación del método Delphi para establecer un modelo conceptual de estimación de costos de software. *Espacios*.
- Note, M. (2015). *Project Management for Information Professionals*. Elsevier Science & Technology.
- Orea, S. V. (2010). Estimación de proyectos de software con puntos de casos de uso.
- Pacienza, J., & Maida, E. G. (2015). *Metodologías de desarrollo de software*.
- Peña, D. (2014). *Fundamentos de estadística*.
- Portillo, J. A.-S., & Paredes, R. I. (2015). *Estimación y Planificación de Proyectos Software con Ciclo de Vida Iterativo-Incremental y empleo de Casos de Uso*.
- Project Management Institute. (2017). *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK), sexta edición*. Newtown Square: Project Management Institute.

- Rahman, A., Muneeb, U., Farhan, U. M., Abdullah, Tufail, K., & Mukhtiar, A. (2020). *Software Cost Estimation – A Comparative Study of COCOMO-II and Bailey-Basili Models*. doi:doi:10.1109/aect47998.2020.9194166
- Rashid, J., Nisar, M. W., Mahmood, T., Rehman, A., & Arafat, S. Y. (2019). A Study of Software Development Cost Estimation Techniques and Models. doi:10.22581/muet1982.2002.18
- Rodríguez, F. S. (1999). *Planificación y Gestión de Sistemas de Información* .
- Rojas, J. V., & Caballero, R. R. (2016). *Plan de mejoramiento de los procesos de planificación de compras, gestión de inventarios y gestión de almacenamiento en la clínica FOSCAL*.
- Salazar, G. (2009). Estimación de proyectos de software: un caso práctico. *Ingeniería y ciencia*.
- Schwaber, K., & Sutherland, J. (2020). *La Guía Definitiva de Scrum: Las Reglas del Juego*.
- Sevilla, E. S. (s.f.). *Métodos Y Técnicas Para La Gestión De Proyectos Software*.
- Thomopoulos, N. (2013). *Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models*. doi:10.1007/978-1-4614-6022-0
- Walston, C. E., & Felix, C. P. (1997). A method of programming measurement and estimation. 54-73. doi:10.1147/sj.161.0054
- Wysocki, R. K. (2019). *Effective Project Management Traditional, Agile, Extreme, Hybrid*. John Wiley & Sons, Inc.