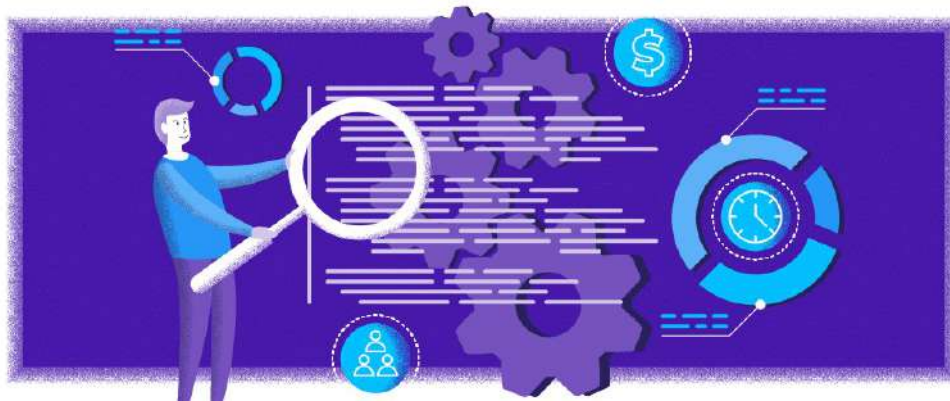




# **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**



Martín, Mateo Tomás.

Trabajo Final de la Carrera Ingeniería Industrial

Departamento de Ingeniería Industrial

Facultad de Ingeniería - Universidad Nacional de Mar del Plata

Mar del Plata, diciembre 2025

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software:  
Estudio de caso en una empresa marplatense.**

**AUTOR**

Martín, Mateo Tomás

**DIRECTOR**

Ing. Antonio Morcela

Facultad de Ingeniería, UNMdP

**CODIRECTOR**

Ing. Isaac Melian

Facultad de Ingeniería, UNMdP

## CONTENIDO

CONTENIDO .....	III
ÍNDICE DE ILUSTRACIONES.....	IV
ÍNDICE DE TABLAS .....	IV
1. INTRODUCCIÓN.....	1
1.1 Objetivos .....	2
2. MARCO TEÓRICO .....	3
2.1 Desarrollo de software.....	3
2.2 Estimaciones de planificación.....	4
2.2.1 Medición de precisión en estimaciones .....	6
2.2.2 Enfoques para estimación.....	7
2.2.3 Juicio de expertos .....	8
2.3 Mejora Continua .....	12
3. METODOLOGÍA.....	14
4. DESARROLLO .....	15
4.1 Introducción a la empresa.....	15
4.1.1 Proceso de abordaje de proyectos .....	18
4.1.2 Técnicas de estimación utilizadas .....	20
4.1.3 Revisión y ajuste de estimaciones durante el proyecto .....	21
4.2 Analizar proyectos pasados para identificar desvíos significativos entre las estimaciones iniciales y los tiempos reales de ejecución.....	22
4.2.1 Detección y tratamiento de valores atípicos .....	24
4.2.2 Análisis de dispersión y estándares de la empresa .....	25
4.3 Determinar los factores clave afectan la variabilidad del proyecto.....	28
4.4 Investigar y comparar metodologías, herramientas y tecnologías disponibles para la estimación de proyectos de software, evaluando su aplicabilidad al contexto de la empresa.....	31
4.4.1. Técnicas tradicionales .....	31
4.4.2 Técnicas de estimación en metodologías ágiles .....	42
4.5 Diseñar una propuesta de mejora para el proceso de estimación de duración de proyectos de software, basada en los hallazgos del análisis.....	44
4.5.1 Checklist de datos a relevar .....	45
4.5.2 Planning Poker .....	47
4.5.3 KPIs .....	48
4.5.4 Ciclo PDCA para la mejora del proceso de estimación.....	49
5. DISCUSIÓN .....	52

6. CONCLUSIONES .....	55
7. BIBLIOGRAFÍA .....	58
8. ANEXO .....	60

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Organigrama empresarial .....	15
Ilustración 2: Mapa conceptual de proyecto.....	20
Ilustración 3: Porcentaje de desviación de proyectos chicos .....	23
Ilustración 4: Porcentaje de desviación de proyectos medianos.....	23
Ilustración 5: Porcentaje de desviación de proyectos grandes .....	24
Ilustración 6: Diagrama de caja y bigote de proyectos de mejoras .....	24
Ilustración 7: Diagrama de caja y bigote de proyectos originales .....	25
Ilustración 8: Desviación absoluta en proyectos originales .....	26
Ilustración 9: Desviación absoluta en proyectos de mejoras.....	26
Ilustración 10: Diagrama de Pareto.....	28
Ilustración 11: Diagrama de Ishikawa .....	31
Ilustración 12: Factores de escala.....	36
Ilustración 13: Áreas clave del proceso.....	37
Ilustración 14: Multiplicadores de esfuerzo .....	38

## ÍNDICE DE TABLAS

Tabla 1: Peso de actores .....	39
Tabla 2: Peso de casos de uso .....	39
Tabla 3: Factores de complejidad técnica.....	40
Tabla 4: Factores ambientales .....	41
Tabla 5: Tabla de proyectos .....	60

## RESUMEN

Este Trabajo Final evalúa las prácticas de estimación de la duración de proyectos de software en una empresa tecnológica de Mar del Plata y diseña una propuesta de mejora orientada a reducir desvíos. Para el análisis se usó una base histórica de proyectos divididos en dos categorías, proyectos originales y proyectos de mejora. Dicha base permitió determinar la precisión en las estimaciones. Los resultados muestran una magnitud promedio del desvío absoluto cercana al 35%, por encima del estándar interno ( $\leq 15\%$ ). En proyectos de mejora la mediana del desvío es +23%, evidenciando tendencia a la subestimación; en proyectos originales la mediana ronda +5%, sin sesgo claro. Las causas encontradas que generan errores en las estimaciones incluyen: involucramiento tardío de usuarios clave, cambios tecnológicos o regulatorios, demoras de insumos del cliente, omisiones de alcance y fallas de comunicación. Con base en estos hallazgos y la literatura de estimación, se propone una intervención metodológica sustentada en cuatro pilares: (i) creación de un *checklist* estandarizado para asegurar completitud y trazabilidad de requisitos; (ii) inclusión de la técnica de estimación conocida como *Planning poker* para estructurar el juicio experto y mitigar sesgos individuales; (iii) definición y cálculo de *kpis* que habiliten control y aprendizaje; y (iv) implementación de un ciclo PDCA para institucionalizar la mejora continua. La contribución práctica es un marco operativo, medible y replicable orientado a obtener estimaciones más precisas y una mejor previsibilidad en los proyectos.

*Palabras clave:* estimación; software; juicio experto; Planning Poker; checklist; KPIs; ciclo PDCA; mejora continua.

## ABSTRACT

This Final Project evaluates the practices used to estimate the duration of software projects in a technology company based in Mar del Plata and designs an improvement proposal aimed at reducing deviations. For the analysis, a historical database of projects was used, divided into two categories: original projects and improvement projects. This database made it possible to determine estimation accuracy. The results show an average magnitude of absolute deviation close to 35%, above the internal standard ( $\leq 15\%$ ). In improvement projects, the median deviation is +23%, indicating a tendency toward underestimation; in original projects, the median is around +5%, showing no clear bias. The identified causes of estimation errors include late involvement of key users, technological or regulatory changes, delays in client deliverables, scope omissions, and communication failures. Based on these findings and estimation literature, a methodological intervention is proposed, grounded on four pillars: (i) creation of a standardized checklist to ensure completeness and traceability of requirements; (ii) inclusion of the estimation technique known as *Planning Poker* to structure expert judgment and mitigate individual biases; (iii) definition and calculation of KPIs that enable control and learning; and (iv) implementation of a PDCA cycle to institutionalize continuous improvement. The practical contribution is an operational, measurable, and replicable framework aimed at achieving more accurate estimates and better project predictability.

*Keywords:* software; estimation; expert judgment; Planning Poker; checklist; KPIs; PDCA; continuous improvement.

# 1. INTRODUCCIÓN

La industria del software ha experimentado un crecimiento notable en las últimas dos décadas, triplicando su peso en el empleo privado y multiplicando por siete su contribución a las exportaciones del país. La estructura empresarial con la que cuenta es diversificada, con grandes empresas transnacionales que comercializan productos en el exterior y prestan servicios al mercado local, conviviendo con empresas grandes y medianas de capital nacional que desarrollan sus propios productos y ofrecen diversos servicios informáticos. También participan de este entramado una gran cantidad de pequeñas empresas locales, microempresas y trabajadores autónomos, particularmente en segmentos con bajas barreras de entrada, como el desarrollo de software de gestión empresarial y videojuegos.

La creciente presencia de empresas del sector del software y la tecnología han hecho que la ciudad de Mar del Plata se posicione como uno de los grandes polos tecnológicos del país. Uno de los actores clave en este crecimiento es ATICMA (Asociación de tecnologías de la información y la comunicación en Mar del Plata y zona), una asociación que agrupa a profesionales independientes, emprendedores, empresas y universidades del sector TIC de la zona. Su objetivo es potenciar la creatividad y la innovación inteligente a través de la tecnología, para favorecer el crecimiento y desarrollo del sector. La asociación, en constante crecimiento, ya agrupa a más de 90 organizaciones asociadas.

A pesar del crecimiento del sector, las empresas de desarrollo de software enfrentan un desafío recurrente: la estimación precisa de la duración de los proyectos.

Históricamente, la estimación en el desarrollo de software ha estado marcada por una alta incertidumbre. A diferencia de otras industrias con procesos más estandarizados y repetitivos, el desarrollo de software presenta una gran variabilidad. Esto puede llevar a retrasos en las entregas, aumento de costos, disminución de la calidad del producto y finalmente a un menor grado de satisfacción del cliente.

# **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

En este contexto, el objetivo de este trabajo será analizar las técnicas de estimación utilizadas por una empresa del sector, identificar los factores que provocan desviaciones respecto a lo planificado e investigar herramientas tecnológicas que puedan mejorar el proceso.

## **1.1 Objetivos**

### **Objetivo general**

Evaluar y analizar las técnicas de estimación utilizadas por una empresa tecnológica marplatense y diseñar una propuesta de mejora en el proceso.

### **Objetivos específicos**

- a) Relevar y describir las técnicas de estimación actualmente utilizadas por la empresa en proyectos de desarrollo de software.
- b) Analizar proyectos pasados para identificar desvíos significativos entre las estimaciones iniciales y los tiempos reales de ejecución.
- c) Determinar los factores clave afectan la variabilidad del proyecto.
- d) Investigar y comparar metodologías, herramientas y tecnologías disponibles para la estimación de proyectos de software, evaluando su aplicabilidad al contexto de la empresa.
- e) Diseñar una propuesta de mejora para el proceso de estimación de duración de proyectos de software, basada en los hallazgos del análisis.

## 2. MARCO TEÓRICO

### 2.1 Desarrollo de software

El desarrollo de software es un proceso que comprende un conjunto de actividades diseñadas para crear, mantener y evolucionar sistemas de software. Abarca desde la especificación de requerimientos hasta el diseño, la implementación, las pruebas, la puesta en producción y el mantenimiento continuo del producto. De acuerdo con Sommerville (2004), el desarrollo de software no solo comprende la programación, sino también todas las actividades asociadas con el ciclo de vida del software, incluyendo la definición de objetivos, el análisis de requerimientos y la gestión del proyecto.

Pressman (2005) define el desarrollo de software como parte de un enfoque sistemático y evolutivo que responde a la necesidad de resolver problemas del mundo real mediante la construcción de productos intangibles, cambiantes y orientados al conocimiento.

El desarrollo de software presenta una serie de características particulares que lo distinguen de otras disciplinas. En primer lugar, se trata de un proceso típicamente iterativo e incremental. Depende en gran medida de las habilidades del equipo para que el proyecto tenga éxito. Además, el desarrollo de software suele estar sujeto a cambios frecuentes ya que los requerimientos de los clientes evolucionan constantemente. Por último, es difícil de medir y de estimar debido a su naturaleza intangible.

Suele dividirse en fases:

- 1) Análisis y especificación de requisitos
- 2) Diseño y desarrollo
- 3) Pruebas
- 4) Implementación
- 5) Mantenimiento y soporte

A estas fases se las conoce como el ciclo de vida del software, y pueden organizarse bajo diferentes enfoques metodológicos como metodologías ágiles o métodos tradicionales como cascada. Los métodos tradicionales, como el modelo en cascada, se basan en una planificación rigurosa y secuencial. En cambio, los enfoques ágiles, como Scrum y Extreme Programming (XP), promueven la flexibilidad, la entrega incremental y la participación activa del cliente durante todo el proceso de desarrollo (Mallidi & Sharma, 2021).

## 2.2 Estimaciones de planificación

Primero es importante aclarar que la estimación de costo y esfuerzo del software nunca será una ciencia exacta (Pressman, 2010).

La estimación es una predicción de cuánto costará o cuál será la duración de un proyecto. Según Jørgensen (2004), la estimación basada en juicio experto es la estrategia más comúnmente utilizada en proyectos de software debido a su flexibilidad y capacidad de adaptarse al contexto específico de cada proyecto. Karner (1993) señala que estimar permite anticipar el esfuerzo requerido a partir de los elementos funcionales del software.

La estimación en software implica la determinación de una o más de las siguientes estimaciones (Rodríguez Soria & Martínez Herráiz, 2010):

- esfuerzo (medido normalmente en hombres-mes).
- duración del proyecto (tiempo de calendario).
- coste (en la moneda correspondiente).

Una estimación precisa en proyectos de software es fundamental para:

- Realizar una planificación financiera efectiva: Permite prever los costos asociados al proyecto, evitando desvíos presupuestarios y facilitando la asignación adecuada de recursos financieros.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

- Asignar recursos de manera eficiente: Ayuda a determinar la cantidad y tipo de recursos humanos y técnicos necesarios, optimizando su utilización y evitando sobrecargas o subutilización.
- Establecer cronogramas realistas: Facilita la creación de plazos alcanzables, mejorando la gestión del tiempo y reduciendo el riesgo de retrasos.
- Gestionar las expectativas de las partes interesadas: Proporciona una base sólida para comunicar el alcance, duración y costos del proyecto a las partes interesadas, fomentando la confianza y la transparencia.
- Identificar de manera temprana posibles riesgos: Permite detectar posibles obstáculos y desafíos antes de que se conviertan en problemas críticos, facilitando la implementación de medidas preventivas.

Hay una serie de problemas que hacen que el software sea difícil de estimar y, por consiguiente, difícil la planeación del proyecto.

Por una parte, esto es debido a las características intrínsecas al software. Por definición, el software es complejo e irreducible, es decir, no se puede simplificar sin pérdida de información. Tiene cuatro características clave que contribuyen a esto, haciendo que el software sea difícil de construir (Ramos Cardozzo, 2016.):

- Complejidad: Entidades de software son extremadamente complejas por su tamaño y no existen dos partes iguales en el nivel de algoritmo.
- Cumplimiento: El software cuenta con varias interfaces diferentes definidas por diferentes personas y por lo general hay poco o nada de cumplimiento en estos entornos.
- Mutabilidad: El software siempre parece ser presionado para cambiar. La mayoría de los cambios están destinados a hacer que el software vaya más allá de sus limitaciones iniciales.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

- La invisibilidad y la intangibilidad: Si bien existen interfaces y lenguajes amigables, el software no se puede ver a través de imágenes. En general, el desarrollador y el usuario no pueden ver el software como un todo.

La precisión de una estimación de proyecto de software se basa en algunos factores (Pressman, 2010):

- a) El grado en el que se estimó adecuadamente el tamaño del producto que se va a construir.
- b) La habilidad para traducir la estimación de tamaño en esfuerzo humano, tiempo calendario y dinero.
- c) El grado en el que el plan del proyecto refleja las habilidades del equipo de software.
- d) La estabilidad de los requisitos del producto y el entorno que soporta el esfuerzo de ingeniería de software.

### 2.2.1 Medición de precisión en estimaciones

Conocer la precisión de las estimaciones realizadas es fundamental para poder identificar necesidades de mejora en ellas. Existen diversas métricas que permiten medir el grado de exactitud de las estimaciones. Los autores Sheppard y Schofield (1997) proponen el uso de 2 de ellas: el MMRE (*Mean Magnitude of Relative Error*<sup>1</sup>) y el Pred(25) (predicción al 25%).

El MMRE se define como el promedio de los errores porcentuales absolutos entre el esfuerzo real (E) y el estimado ( $\hat{E}$ ), y se expresa matemáticamente como:

$$MMRE = \frac{1}{n} \sum_{i=1}^n \left| \frac{E_i - \hat{E}_i}{E_i} \right| * 100$$

Esta métrica es conservadora y tiende a penalizar con mayor severidad las sobreestimaciones.

---

<sup>1</sup> Mean Magnitude of Relative Error: Magnitud Media del Error Relativo.

El Pred(25) representa el porcentaje de predicciones cuya desviación respecto al valor real se encuentra dentro del 25%. Se calcula mediante la siguiente fórmula:

$$Pred(25) = \frac{\text{Número de estimaciones con } \left| \frac{E_i - \hat{E}_i}{E_i} \right| \leq 0.25}{n} * 100$$

A diferencia del MMRE identifica las estimaciones que son consistentemente precisas, aunque podría ocultar estimaciones gravemente erróneas si estas no superan el umbral del 25% (Shepperd & Schofield, 1997). Esto quiere decir que una técnica utilizada podría mostrar un Pred(25) alto sin importar si en algunas estimaciones presenta errores graves.

Para determinar la precisión de las estimaciones en la empresa analizada en este trabajo se utilizarán ambas métricas. La combinación de ambas métricas tiene la intención de obtener una vista de ambos enfoques y en base a ello realizar un análisis que permita una medición más adecuada de la precisión.

## 2.2.2 Enfoques para estimación

Los enfoques más viables, según Pressman (2010), son dos:

- Las técnicas de descomposición.
- Los modelos empíricos.

Las primeras implican dividir el proyecto en componentes manejables, estimando esfuerzo y costos para cada uno; los segundos utilizan fórmulas derivadas de datos históricos y métricas como las líneas de código (LOC) o los puntos de función (PF). Ambos enfoques pueden complementarse entre sí y con herramientas automatizadas de estimación, siempre que se cuente con bases de datos confiables.

Una estimación precisa depende de una adecuada estimación del tamaño del producto, la disponibilidad de métricas confiables, la correcta planificación y la estabilidad de los requisitos. Para abordar la incertidumbre, se sugiere utilizar estimaciones de tres puntos (optimista, más probable y pesimista), calculando un valor esperado a través de una distribución beta ponderada (Pressman, 2010).

### 2.2.3 Juicio de expertos

El juicio de expertos es una de las técnicas más utilizadas para la estimación en proyectos de software. La técnica consiste en dialogar y consultar con personas con experiencia relevante para que realicen estimaciones basadas en su conocimiento de proyectos similares. Según Jørgensen (2005) no existe evidencia empírica que favorezca el uso de modelos formales de estimación.

Según Santos (2014), el juicio de expertos se basa en la experiencia del estimador, sin necesidad de aplicar modelos matemáticos o algoritmos. La ventaja es que permite utilizar conocimiento tácito que difícilmente puede ser formalizado. Además, puede aplicarse de forma individual o grupal, a través de métodos estructurados como Delphi o Wideband Delphi, que ayudan a reducir sesgos y alcanzar consenso.

Jørgensen (2002) sostiene lo siguiente: La estimación por expertos es la estrategia de estimación más utilizada en los proyectos de software, no existe evidencia sustancial a favor del uso de modelos de estimación, y hay situaciones en las que podemos esperar que las estimaciones de expertos sean más precisas que los modelos de estimación formales. (Jørgensen, 2004, 1)

El autor llega a la conclusión de que es el método más utilizado en la industria del software. Esto se debe en parte a su facilidad de aplicación, flexibilidad y a la capacidad del método de adaptarse a distintos proyectos. En su trabajo realiza una revisión de distintos estudios que indican que las estimaciones basadas en juicios de expertos han sido igual de acertadas o hasta superiores que modelos formales. De sus estudios de estimación de expertos y de estimación por modelos se pueden destacar las siguientes conclusiones:

- Las estimaciones de expertos son más precisas que las de los modelos cuando los expertos poseen (y aplican eficientemente) conocimientos importantes del dominio que no están incluidos en los modelos de estimación.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

- Las estimaciones de expertos son más precisas que las de los modelos cuando la incertidumbre es baja. Las estimaciones de los modelos son más precisas cuando la incertidumbre es alta.
- Los expertos utilizan estrategias de estimación simples (heurísticas) y se desempeñan tan bien como, o mejor que, los modelos de estimación cuando estas estrategias simples (heurísticas) son válidas.
- Los expertos pueden estar fuertemente sesgados y ser influenciados por información irrelevante, por ejemplo, tendiendo al exceso de optimismo. Los modelos de estimación presentan menos sesgos.

Según Jørgensen (2004) hay 12 principios para una estimación experta. los primeros 6 son principios orientados a reducir los sesgos situacionales y humanos:

- 1) Evaluar la precisión en las estimaciones, pero a la vez evitar la alta presión de evaluación: La evaluación constante ayuda a identificar errores sistemáticos y lograr mejoras, pero si el estimador se siente muy presionado puede terminar alterando sus estimaciones para protegerse.
- 2) Evitar metas de estimación en conflicto: Mezclar objetivos distintos como presupuestar, planificar o estimar el esfuerzo más probable puede afectar la precisión. Además, las estimaciones suelen verse influenciadas por el deseo de obtener resultados positivos (*wishful thinking*), lo cual genera un sesgo optimista que quita claridad a la estimación.
- 3) Pedir a los estimadores que justifiquen y critiquen sus estimaciones: Es una práctica que impulsa la reflexión y requiere que los expertos justifiquen y critiquen sus estimaciones. Hace que el proceso tome un carácter más analítico aumentando la confianza en la estimación.
- 4) Evitar información de estimación irrelevante y poco confiable: Aun sabiendo que la información es irrelevante, las estimaciones de los expertos se ven

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

negativamente impactadas por ella. Es por eso que se debe eliminar y evitar por todos los medios posibles.

- 5) Usar datos documentados de tareas de desarrollo previas: Se recomienda que los expertos utilicen datos documentados de proyectos anteriores para obtener una mejora en sus estimaciones. De esta forma los estimadores pueden lograr una estimación más analítica con menos influencias y sesgos humanos.
- 6) Buscar estimadores expertos con conocimientos relevantes del dominio y buenos antecedentes de estimaciones: Se suele pensar que las personas más preparadas para resolver las tareas son quienes deberían estimarla. Jørgensen considera que es necesario hacer algunos ajustes a este supuesto. Entre otras cosas, el autor cree que la relevancia de experiencias solo es aplicable a situaciones similares. También explica que las personas se suelen confiar por demás cuando reciben una serie de estimaciones más complejas que las habituales. Por estas y otras razones es que se debe buscar personas con experiencia relevante pero también con buenos resultados en estimación.

Los siguientes 4 principios apoyan al proceso de estimación:

- 7) Estimar utilizando estrategias *top-down*<sup>2</sup> y *bottom-up*<sup>3</sup> de forma independiente: Con el objetivo de complementar las fortalezas y reducir las debilidades de ambos métodos el autor propone implementarlos de manera independiente para luego comparar los resultados. Al realizarlos de manera independiente se evita el efecto anclaje en donde las estimaciones se ven influenciadas por las otras. De esta forma, en caso de existir desviaciones importantes en los resultados, se puede agregar más información a la estimación o agregar expertos.

---

<sup>2</sup> *Top-down*: Arriba-abajo.

<sup>3</sup> *Bottom-up*: Abajo-arriba.

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

La estrategia top-down es más robusta en cuanto a eventos inesperados o tareas olvidadas. Se basa en el pensamiento basado en la historia. Mientras tanto, la estrategia bottom-up lleva a un aumento en la comprensión de la ejecución y planeación del proyecto, pero puede ser más propensa al optimismo.

- 8) Usar listas de verificación de la estimación: Ayuda a las personas a cargo de la estimación a recordar tareas y considerar eventos inesperados. Mejoran la consistencia de las estimaciones evitando desviaciones cuando la información de entrada es la misma. Además, fomentan la generación de alternativas posibles contrarrestando la tendencia de las personas a considerar solo las opciones que les fueron presentadas.
- 9) Combinar estimaciones de diferentes expertos y estrategias: La combinación de estimaciones de múltiples expertos puede mejorar la precisión respecto a basarse en una sola estimación. Las formas de combinación son diversas. Se puede ponderar las estimaciones individuales, realizar discusiones grupales para llegar a una única estimación final o promediar las estimaciones, entre otras.

Los factores que afectan a la estimación son: 1) El número de expertos; 2) La precisión de cada experto; 3) La relación entre los expertos; 4) El método de combinación de las estimaciones.

- 10) Evaluar la incertidumbre de la estimación: Evaluar la incertidumbre de una estimación es importante para realizar una planificación realista y aprender de estimaciones pasadas. Una mala estimación no se debe necesariamente a un error en el proceso, sino que puede darse cuando hay un alto grado de incertidumbre.

Es por esto que el autor propone trabajar con intervalos de predicción. En sus estudios se dio cuenta de que los estimadores suelen tener sobre confianza y calculan un intervalo muy estrecho. Para reducir esa sobre confianza (Jørgensen & Teigen, 2002, 3) proponen lo siguiente: 1) Estimar el esfuerzo más probable; 2) Calcular el mínimo y máximo esfuerzo con proporciones fijas; 3) Decidir el nivel de confianza.

Los últimos 2 principios están basados en el *feedback*<sup>4</sup> y oportunidades de entrenamiento.

- 11) Proveer feedback sobre la precisión de la estimación y sobre relaciones entre tareas: El mayor problema con el feedback sobre estimaciones es que hay un lapso de tiempo elevado entre la estimación y el momento de recepción del feedback. Se debe trabajar para poder dar un feedback de manera más temprana ya que permite aumentar el aprendizaje sobre la precisión de las estimaciones.
- 12) Proveer oportunidades de entrenamiento de estimaciones: Las organizaciones deben proveer oportunidades de entrenamiento sobre estimación basadas en bases de datos de proyectos completados. Las sesiones deberían incluir la aplicación de diferentes procesos de estimación sobre proyectos completados basados en la información disponible que se tenía al momento de estimar.

## 2.3 Mejora Continua

La mejora continua es un enfoque sistemático y permanente orientado a optimizar procesos, productos y servicios mediante la identificación y eliminación de ineficiencias, promoviendo así la eficiencia y la calidad en las organizaciones.

---

<sup>4</sup> *Feedback*: Retroalimentación.

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Según Deming (1986), la mejora continua se basa en la idea de que cada proceso es susceptible de perfeccionamiento y que este avance debe ser sostenido en el tiempo a través de la aplicación de metodologías estructuradas. En este sentido, Deming (1986) introduce el ciclo PDCA (*Plan-Do-Check-Act*) como una herramienta clave para la implementación de mejoras progresivas en la gestión de calidad. Este ciclo permite a las organizaciones abordar la mejora continua de manera estructurada:

- Plan (Planificar): Identificación de oportunidades de mejora y establecimiento de objetivos claros.
- Do (Hacer): Implementación de cambios en un entorno controlado.
- Check (Verificar): Evaluación de resultados y análisis de efectividad.
- Act (Actuar): Estandarización de mejoras y reinicio del proceso.

### **3.METODOLOGÍA**

Se realizará un diagnóstico inicial para evaluar las técnicas utilizadas por la empresa. Para ello se realizarán entrevistas semiestructuradas con el personal involucrado y se hará una revisión de documentación interna (análisis documental).

Con el objetivo de descubrir desviaciones entre los resultados observados y los planificados, identificar patrones y buenas prácticas, se analizarán proyectos finalizados (revisión documental y sistematización de experiencias). Se buscará descubrir causas en los errores en la estimación (identificación de causa raíz).

Se investigará acerca de las buenas prácticas (benchmarking), herramientas y tecnología disponible (vigilancia tecnológica) en la industria utilizadas para conseguir una estimación adecuada.

Finalmente se buscará diseñar una propuesta de mejora para el proceso de estimación de la duración de los proyectos basada en los resultados obtenidos en los análisis e investigaciones previas.

## 4. DESARROLLO

### 4.1 Introducción a la empresa

EMMSA es una empresa tecnológica con sede en Mar del Plata que cuenta con más de 30 años de trayectoria en el desarrollo e implementación de soluciones informáticas orientadas a organizaciones de alta complejidad. Su equipo está conformado por más de 50 profesionales especializados en sistemas, con una sólida formación técnica y experiencia en sectores estratégicos como la energía, los servicios públicos y la industria. A continuación, se presenta el organigrama de la empresa. La empresa posee la siguiente estructura organizativa interna (roles):

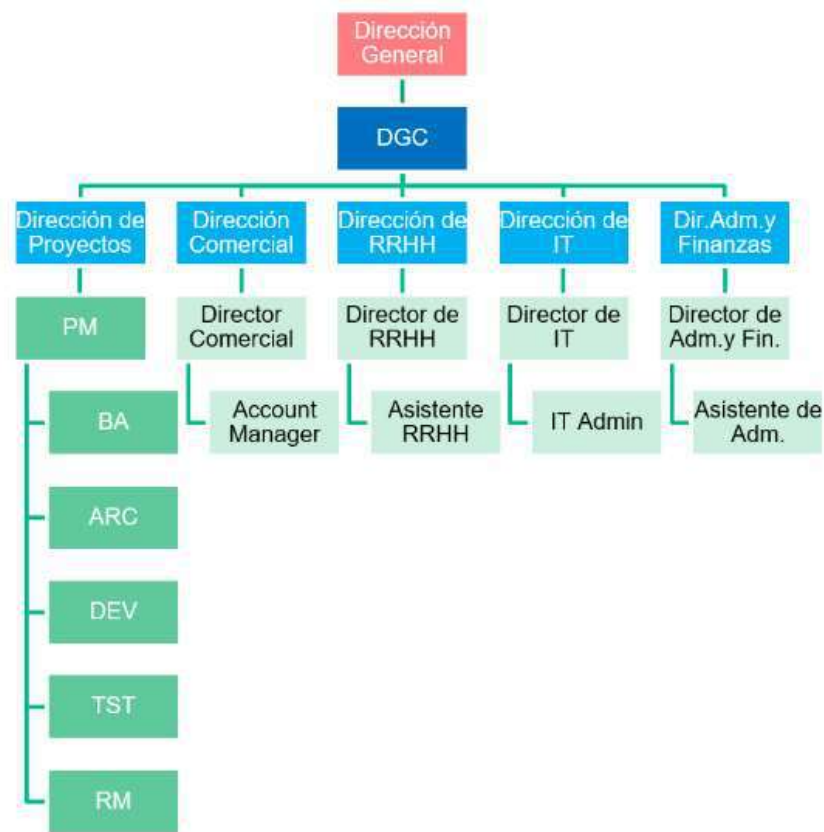


Ilustración 1: Organigrama empresarial. Fuente: Emmsa (2025)

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

A lo largo de su historia, EMMSA ha consolidado una cartera de clientes compuesta por empresas nacionales e internacionales de primer nivel, destacándose por su capacidad para gestionar proyectos de gran envergadura con altos estándares de calidad. Entre sus principales clientes y aliados estratégicos se encuentran empresas del sector energético como Tecpetrol, Pluspetrol, EDEA S.A., ENERSA y Siemens Gamesa, con quien mantiene vínculos desde hace más de 15 años.

La empresa sigue un enfoque metodológico propio, diseñado a partir de la experiencia acumulada y basado en buenas prácticas reconocidas a nivel internacional. Su modelo de trabajo integra los lineamientos del Project Management Institute (PMI) para la gestión de proyectos y principios de metodologías ágiles para el desarrollo de software. Este enfoque permite alinear los objetivos del cliente con una planificación flexible.

EMMSA cuenta con certificaciones ISO 9001:2015 e ISO 90003, lo cual respalda su compromiso con la mejora continua y el cumplimiento de estándares internacionales en sus procesos de diseño, construcción y aseguramiento de calidad del software. Asimismo, ha desarrollado múltiples soluciones integradas con plataformas empresariales como SAP y Salesforce, aplicando tecnologías Microsoft y contando con certificaciones de competencias otorgadas por esta firma.

La empresa organiza su actividad en torno a dos grandes líneas de proyectos:

1. Proyectos de productos propios: desarrollo de soluciones orientadas a mercados verticales, especialmente en los sectores de energía y seguros.

En el ámbito energético, destacan dos productos principales:

GEASIS: un sistema integral para la gestión del ciclo activo de las áreas comerciales, administrativas y técnicas de empresas distribuidoras de servicios públicos.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

ADGAS: una solución especializada para el sector de transporte y distribución de gas, que permite realizar el cálculo de despacho y transporte, optimizando la gestión operativa del recurso.

En cuanto al sector de aseguradoras, la empresa ha desarrollado un producto modular llamado ADInsurance que les permite a las empresas tener una gestión segura, eficiente e integrada.

2. Proyectos de desarrollo a medida: La empresa también ofrece servicios de desarrollo de software especializado para clientes que requieren soluciones a medida.

La empresa categoriza a los proyectos según la cantidad de horas hombre que le va a costar. Las categorías en las que caen los proyectos son 3: Proyectos Chicos, Medianos y Grandes.

Cuenta con un número variado que suele rondar entre los 20 y 30 proyectos al año. La gran mayoría de esos proyectos caen dentro de la categoría de proyectos medianos. Luego cuenta con alrededor de 3 proyectos grandes y 6 proyectos chicos. De esos proyectos, la mayoría están relacionados con el mantenimiento (preventivo, normativo y correctivo) y actualización de software ya desarrollado por la empresa siendo un número considerablemente menor el de nuevos desarrollos.

En cuanto al parámetro para categorizar los proyectos la empresa utiliza los siguientes intervalos: menos de 500 horas hombre se categoriza como proyecto chico; entre 500 y 1500 horas hombre se categoriza como proyecto mediano; y más de 1500 horas hombre se categoriza como proyecto grande.

#### 4.1.1 Proceso de abordaje de proyectos

EMMSA posee un proceso estructurado para la recepción, análisis y planificación de nuevos proyectos, las etapas principales se describen a continuación:

1. Recepción del RFI (*Request for Information*): El proceso se inicia con la recepción de una solicitud de información por parte del cliente potencial, la cual permite obtener una visión preliminar de las necesidades y del contexto del proyecto. Esta etapa tiene como objetivo establecer un primer contacto formal y validar el encuadre general de la solución requerida.
2. Recepción del RFP (*Request for Proposal*): En esta etapa, el cliente presenta una solicitud detallada de propuesta, donde se especifican los requerimientos funcionales, técnicos y operativos del proyecto. A partir de este documento, EMMSA comienza a trabajar en una devolución estructurada que contempla tanto los aspectos técnicos como los comerciales.
3. Análisis de requerimientos: Esta fase implica un estudio profundo de los requerimientos planteados en el RFP. El equipo de EMMSA realiza una interpretación detallada de cada necesidad, evaluando posibles soluciones y adaptaciones en función de su conocimiento técnico y funcional.
4. Evaluación de factibilidad interna: Una vez analizados los requerimientos, se lleva a cabo una reunión interna. Este encuentro tiene por objetivo realizar una doble evaluación de factibilidad: Factibilidad técnica, que se determina cómo se podría implementar cada requerimiento desde el punto de vista tecnológico; y Factibilidad funcional, que se analiza si la solución propuesta es viable en términos de cumplimiento de los objetivos del cliente.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Durante esta etapa se recurre al juicio de expertos dentro de la organización, quienes, en base a su experiencia, aportan estimaciones de esfuerzo, identifican posibles riesgos, y sugieren estrategias de implementación. Como resultados se obtienen:

- Definición preliminar del equipo necesario (perfiles técnicos y niveles de experiencia).
  - Estimación inicial del calendario de trabajo.
  - Identificación de aspectos críticos que requieren investigación o validación adicional.
5. Elaboración y envío de la propuesta técnico-comercial: Finalmente, se consolida la información obtenida en una propuesta formal que incluye:
- Una descripción técnica de la solución.
  - El alcance funcional.
  - La estimación de esfuerzo y calendario.
  - La estructura de costos asociada.

Esta propuesta es enviada al cliente como respuesta integral al RFP, constituyendo la base para la posterior negociación y eventual formalización del contrato.

A continuación, se puede observar un mapa conceptual de un proyecto a modo de ejemplo (Ilustración 2).

### Mapa conceptual del Proyecto Relación Fases Temporales/ Módulos de Servicios

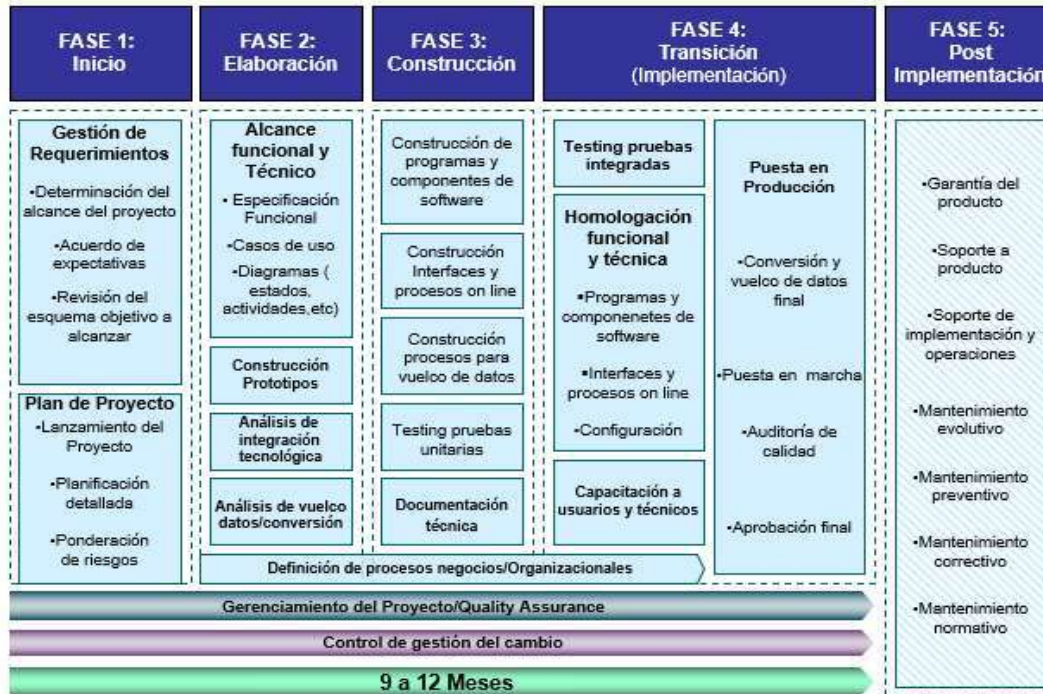


Ilustración 2: Mapa conceptual de proyecto. Fuente: Emmsa (2025)

#### 4.1.2 Técnicas de estimación utilizadas

EMMSA emplea actualmente una técnica de estimación basada en el juicio de expertos, la cual se apoya en la extensa trayectoria profesional de su equipo técnico. Los profesionales involucrados en esta práctica han liderado y participado en numerosos proyectos de desarrollo e implementación de software a lo largo de varias décadas, abarcando sectores diversos como energía, seguros y servicios públicos.

El proceso de estimación se estructura en torno a reuniones técnicas donde se convocan expertos con conocimiento relevante en función del tipo de proyecto. A través de una puesta en común, se discuten las posibles soluciones, se analiza la factibilidad técnica y funcional, y se proyecta un orden de magnitud en términos de recursos humanos, perfiles necesarios y plazos de ejecución. Esta estimación se basa no sólo en criterios técnicos, sino también en la experiencia directa adquirida a lo largo del recorrido de la empresa.

#### 4.1.3 Revisión y ajuste de estimaciones durante el proyecto

El proceso de estimación no se limita a una única instancia previa al inicio del proyecto, sino que puede ser revisado y ajustado a lo largo de su desarrollo. Esta práctica responde principalmente a la dinámica natural de los requerimientos del cliente, los cuales pueden experimentar modificaciones en función de nuevas necesidades, cambios en el entorno operativo o redefinición de prioridades.

Cuando se presentan cambios en los requerimientos, el equipo analiza el impacto técnico y funcional de las modificaciones propuestas. En función de la magnitud y complejidad de dichos cambios, se sigue uno de los siguientes cursos de acción:

- Para ajustes menores, se genera una adenda a la cotización original, la cual se integra al alcance acordado sin alterar significativamente los plazos ni los recursos.
- En casos donde las modificaciones implican nuevas funcionalidades o alteraciones sustanciales al diseño inicial, se procede a una nueva cotización, que incluye una estimación actualizada en términos de esfuerzo, costos y cronograma.

## 4.2 Analizar proyectos pasados para identificar desvíos significativos entre las estimaciones iniciales y los tiempos reales de ejecución.

Para la elaboración de este estudio se solicitó a la empresa un listado completo de proyectos desarrollados entre 2016 y 2025. El listado inicial contenía 140 proyectos.

Se identificó un conjunto de proyectos cuya duración efectiva fue inferior a las 100 horas. Con el objetivo de analizar su impacto en el estudio, se realizó un examen específico de este subconjunto.

Se observó que en ninguno de estos proyectos se registraron retrasos en la entrega al cliente. Desde el punto de vista de la estimación, estos proyectos presentan particularidades que hacen que su inclusión en el análisis general carezca de sentido. La empresa establece un mínimo de facturación para los trabajos, de manera que, aun cuando un proyecto demande entre 10 y 20 horas reales de desarrollo, el cliente es facturado como si se hubiesen utilizado 100 horas. Este criterio responde a razones de política comercial, destinadas a cubrir los costos fijos de gestión, coordinación y administración. Es por eso que las estimaciones de estos proyectos no reflejan una medida realista sobre el esfuerzo requerido. Esto genera una distorsión en los indicadores de desviación entre horas estimadas y reales

Por esta razón se decidió excluir a los proyectos de menos de 100 horas de los análisis principales resultando en una serie conformada por 77 proyectos (ver tabla 5). A continuación, los proyectos se agruparon en dos grandes categorías según su naturaleza:

- **Proyectos de mejora:** intervenciones sobre sistemas ya existentes.
- **Proyectos originales:** desarrollo de nuevos sistemas.

Dicha categorización se realizó debido a que se esperaba que los proyectos originales estén asociados a una mayor desviación dada la falta de conocimiento previo del sistema.

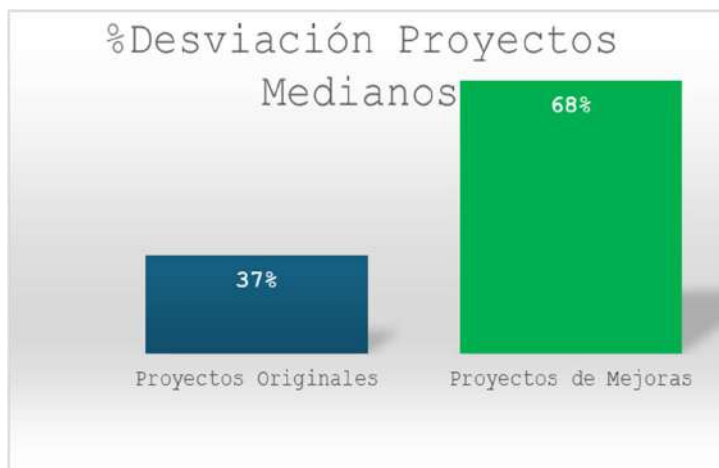
**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Dentro de cada clase se realizó, inicialmente, una subdivisión de proyectos por tamaño—pequeños, medianos y grandes—con criterio de la propia empresa, esperando que los proyectos de menor duración mostraran menores desviaciones relativas. Sin embargo, el análisis de seis gráficos de barras ubicados a continuación, evidenció la ausencia de correlación consistente entre el tamaño y el porcentaje de desviación. Por este motivo, las categorías de dimensión quedaron fuera del análisis comparativo principal, manteniendo únicamente la distinción entre proyectos de mejora y originales.



*Ilustración 3: Porcentaje de desviación de proyectos chicos. Fuente: Elaboración propia*



*Ilustración 4: Porcentaje de desviación de proyectos medianos. Fuente: Elaboración propia*



Ilustración 5: Porcentaje de desviación de proyectos grandes. Fuente: Elaboración propia

#### 4.2.1 Detección y tratamiento de valores atípicos

Para identificar *outliers*<sup>5</sup> en ambas series de datos, se construyeron diagramas de caja y bigotes (Figuras). Los puntos extremos localizados fueron extraídos del conjunto de datos y estudiados individualmente, con el fin de comprender las causas de sus desviaciones excepcionales, sin que estos influenciaran los estadísticos globales de cada categoría.

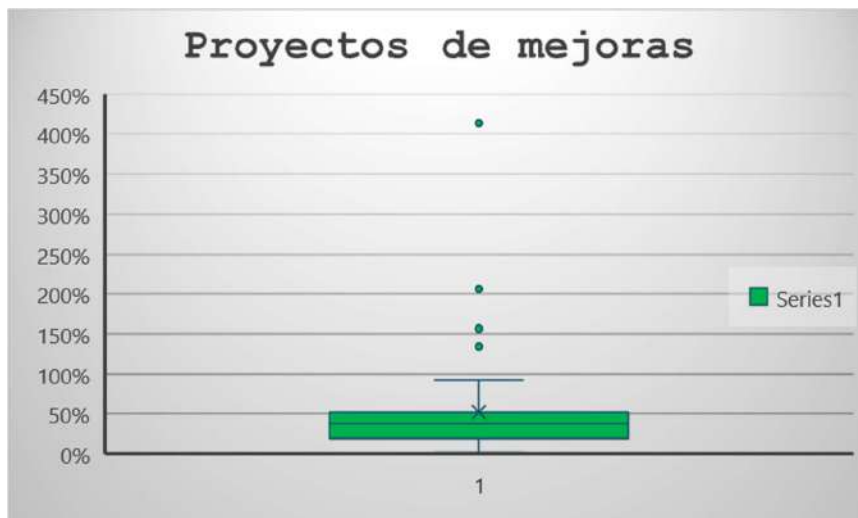


Ilustración 6: Diagrama de caja y bigote de proyectos de mejoras. Fuente: Elaboración propia

<sup>5</sup> *Outliers*: Valores atípicos

Se pueden identificar 4 proyectos cuya desviación está fuera de los parámetros de la serie. Los códigos de dichos proyectos son 4008, 4356, 4137, 4164.

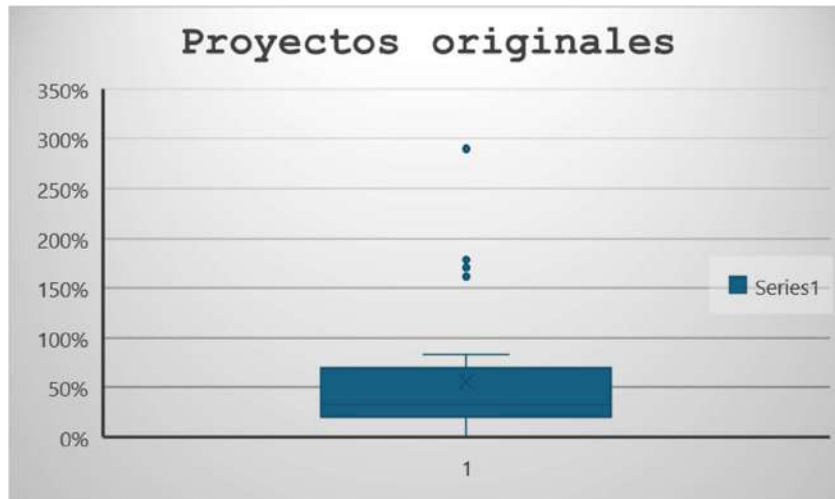


Ilustración 7: Diagrama de caja y bigote de proyectos originales. Fuente: Elaboración propia

Se pueden identificar 4 proyectos cuya desviación está fuera de los parámetros de la serie. Los códigos de dichos proyectos son 4734, 4357, 4724, 4147.

#### 4.2.2 Análisis de dispersión y estándares de la empresa

Con las series limpias, se graficó la desviación absoluta de cada proyecto en función de su orden de ejecución (eje x) versus el % de desviación (eje y), generando dos diagramas de dispersión: uno para proyectos de mejora y otro para originales (Figuras). En ambos gráficos se trazaron dos líneas de referencia correspondientes a los umbrales definidos por la empresa:

- **Buena estimación:** desviación menor al 15 %.
- **Mala estimación:** desviación superior al 40 %.

Esta estrategia permitió clasificar visualmente los proyectos según cumplimiento de estándares y focalizar el análisis documental en aquellos casos con una pequeña o gran desviación. Para dicho análisis, se utilizaron los documentos de cierre de proyectos.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

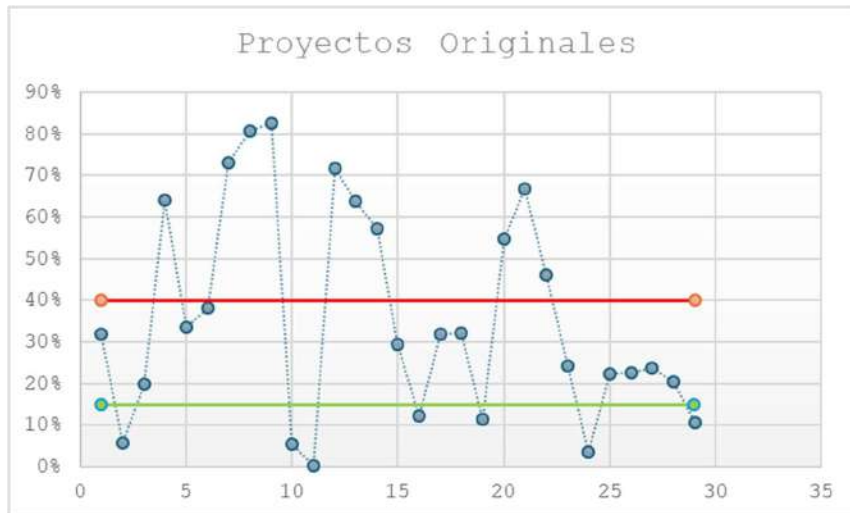


Ilustración 8: Desviación absoluta en proyectos originales. Fuente: Elaboración propia

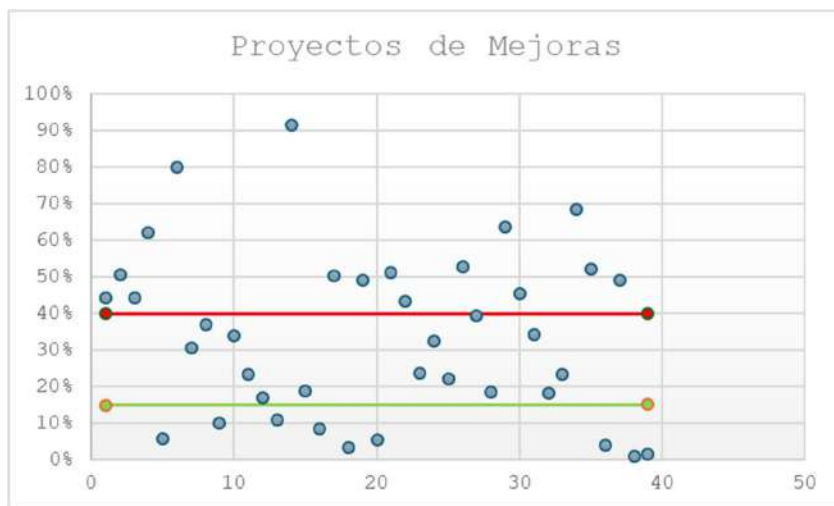


Ilustración 9: Desviación absoluta en proyectos de mejoras. Fuente: Elaboración propia

Las conclusiones obtenidas del análisis fueron:

- Proyectos de mejoras: la mediana del desvío fue de +23%, lo que indica una tendencia a subestimar el esfuerzo. Solo un 22% de los proyectos alcanzó el estándar de buena estimación (<15% de desvío absoluto), mientras que un 42% presentó desvíos superiores al 40%.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

*Martin, M.*

- Proyectos originales: la mediana del desvío fue de +5%. Esto significa que no hay una tendencia clara hacia la sobre o subestimación del esfuerzo. En este grupo, un 24% de los proyectos cumplió con el estándar de precisión, y un 34% tuvo desvíos mayores al 40%.

En ambos casos, la magnitud promedio del desvío absoluto se mantiene cercana al 35%, lo que implica que, aun sin la influencia de valores atípicos, la dispersión es considerable y la precisión general está por debajo de lo esperado según la política de la empresa. El bajo porcentaje de proyectos que cumplen el estándar del 15% refuerza la necesidad de ajustar las prácticas de estimación.

En paralelo se realizó un análisis de Pareto para observar el comportamiento de la serie e identificar los proyectos que concentran el mayor impacto en las estimaciones.

Para el análisis se definió un indicador de impacto del desvío, calculado como:

$$\text{Impacto} = |\% \text{ de desvío}| * \text{Horas Reales}$$

Este indicador pondera simultáneamente la magnitud relativa del error en la estimación y el tamaño efectivo del proyecto en términos de esfuerzo. De este modo, se evita que un error porcentual elevado en un proyecto pequeño aparezca con el mismo peso que un error de igual magnitud en un proyecto de gran escala.

Con dicho indicador se realizó un gráfico de Pareto (Ilustración 10). La serie no cumple estrictamente con las condiciones de Pareto. De un total de 69 proyectos, 21 (30%) son los que explican el 80% del impacto total en los desvíos de estimación. Se puede observar una curva con un crecimiento gradual lo que indica que los errores en la estimación no están altamente concentrados y que estos se distribuyen a lo largo de los proyectos.



Ilustración 10: Diagrama de Pareto. Fuente: Elaboración propia

### 4.3 Determinar los factores clave afectan la variabilidad del proyecto.

Durante el relevamiento realizado en la empresa se identificaron múltiples factores que, desde la experiencia del equipo de desarrollo, contribuyen de manera significativa a las desviaciones entre los valores estimados y los resultados reales. A continuación, se nombran los factores principales a los que los referentes de la organización atribuyen el desvío asociado a los proyectos cuya estimación se considera inadecuada.

Una de las causas más recurrentes es la falta de involucramiento temprano de usuarios clave. En numerosos casos, las personas que poseen el mayor conocimiento sobre las necesidades del negocio se incorporan de manera tardía al ciclo de desarrollo. Esto genera que la definición de requerimientos no sea adecuada impactando negativamente en el esfuerzo necesario y los plazos de entrega. Asimismo, muchas veces los clientes no tienen en claro sus necesidades lo que concluye en suposiciones incorrectas.

Otro factor corresponde a la modificación de las plataformas tecnológicas en donde será instalado el sistema.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Cambios en el lenguaje de programación se asocian a desvíos significativos en la estimación. Estos cambios en el lenguaje requieren de un proceso de aprendizaje y adaptación por parte del equipo.

El incumplimiento de plazos de entrega por parte del cliente tanto en información relevante o en la habilitación de accesos a sistemas y bases de datos interrumpe la continuidad del trabajo del equipo de desarrollo y retrasa tareas. En muchos casos esto lleva a una reprogramación de actividades reduciendo los plazos e incrementando el riesgo de errores en el desarrollo.

En industrias reguladas como lo es la energía cambios normativos por parte del ente regulador es otro factor que afecta de manera significativa al esfuerzo estimado en un proyecto. Dichos cambios están asociados a nuevas condiciones que el sistema debe cumplir lo cual conlleva nuevos desarrollos.

Otro factor mencionado por la empresa es la resistencia al cambio por parte de las partes interesadas en el entorno del cliente. Se manifiesta cuando la solución propuesta implica modificaciones sustanciales en los procesos, roles o hábitos establecidos. Aunque el producto cumple con los requisitos funcionales y técnicos, los usuarios demuestran una resistencia y esta oposición genera re trabajos.

Muchas veces la empresa pasa por alto tareas o elementos que luego agregan trabajo no previsto y generan desvíos. Esto ocurre tanto por olvido como por una falta de compromiso con el proceso de desglose del proyecto.

Una mala comunicación entre el cliente, los desarrolladores y los usuarios. La falta de retroalimentación continua y bidireccional impide la detección temprana de desviaciones. Si el cliente no revisa con la frecuencia adecuada los prototipos o entregas parciales, el equipo carece de la información necesaria para corregir el rumbo antes de completar mayor cantidad

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

de trabajo. Esta falta de validación temprana incrementa el riesgo de llegar a etapas avanzadas de desarrollo con requisitos equivocados.

Con el objetivo de visualizar e identificar las causas raíz que generan errores en la estimación se confeccionó un diagrama de Ishikawa. En el mismo se observa una línea central que representa el problema principal (errores en las estimaciones). A partir de esta línea se despliegan en distintas categorías las causas potenciales de dicho error. Se pueden distinguir 5:

- Factores técnicos: Este factor se relaciona con los aspectos propios del software como la complejidad de los sistemas o decisiones en cuanto al diseño y construcción.
- Factores funcionales: Hace referencia a la definición de los objetivos y requerimientos del sistema.
- Factores de gestión del proyecto: Se vincula al modo en que se planifica, organiza y conduce el desarrollo del proyecto.
- Factores organizacionales: Está relacionado con la estructura y el contexto de la empresa, tanto del cliente como de la empresa.
- Factores tecnológicos: Hace referencia a los elementos vinculados al entorno tecnológico sobre el cual se desarrolla el proyecto.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

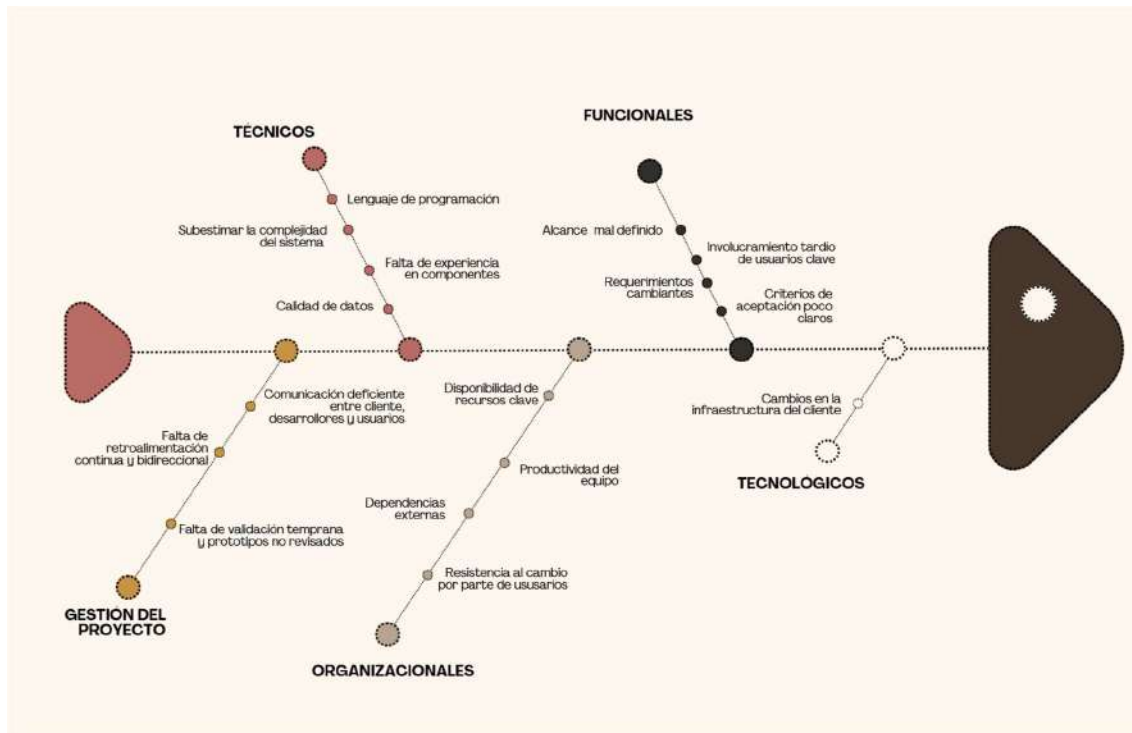


Ilustración 11: Diagrama de Ishikawa. Fuente: Elaboración propia

### 4.4 Investigar y comparar metodologías, herramientas y tecnologías disponibles para la estimación de proyectos de software, evaluando su aplicabilidad al contexto de la empresa

#### 4.4.1. Técnicas tradicionales

Estas técnicas se basan en datos históricos y fórmulas matemáticas. Entre las más reconocidas se encuentran modelos algorítmicos como COCOMO o puntos de fusión, métodos de descomposición y estimación por experiencia como juicio de expertos y analogías.

##### 4.4.1.1 Estimación por analogía

La estimación por analogía se basa en la comparación del proyecto actual con otros proyectos similares previamente realizados. La suposición es que si dos proyectos tienen

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

características similares (tamaño, complejidad, el equipo de trabajo, etcétera) entonces el esfuerzo para realizarlos también debe serlo. Esta técnica utiliza información histórica de proyectos anteriores para predecir el esfuerzo, costo o duración de un nuevo proyecto, identificando aquellos casos más parecidos y extrapolando los valores estimados a partir de ellos (Shepperd & Schofield, 1997).

Según Santos (2014), esta técnica es especialmente útil cuando se dispone de un historial confiable de proyectos previos, ya que permite reutilizar el conocimiento empírico acumulado y realizar ajustes según las diferencias específicas entre proyectos.

La técnica presenta las siguientes ventajas (Shepperd & Schofield, 1997):

- Evita los problemas asociados tanto con la elicitación del conocimiento como con la extracción y codificación del mismo.
- Los sistemas basados en analogía solo necesitan abordar los problemas que realmente ocurren en la práctica, mientras que los sistemas algorítmicos deben manejar todos los problemas posibles.
- Los sistemas basados en analogía también pueden manejar casos fallidos (es decir, aquellos casos para los cuales no se realizó una predicción precisa). Esto es útil ya que permite a los usuarios identificar situaciones potencialmente de alto riesgo.
- La analogía es capaz de manejar dominios poco comprendidos (como los proyectos de software), ya que las soluciones se basan en lo que realmente ha ocurrido, a diferencia de los sistemas basados en reglas, que se apoyan en cadenas de reglas.
- Los usuarios pueden estar más dispuestos a aceptar soluciones provenientes de sistemas basados en analogía, ya que estas se derivan de una forma de razonamiento más cercana a la resolución de problemas humanos.

La estimación por analogía puede aplicarse de manera informal (basada en memoria y experiencia del estimador) o mediante enfoques más sistemáticos, como modelos basados en casos o técnicas estadísticas. En su revisión, Jørgensen (2004) señala que la analogía es una de las estrategias más frecuentes dentro del juicio experto, y que su efectividad depende fuertemente de la calidad de los datos y del juicio del estimador.

El método permite aprovechar la experiencia real de proyectos anteriores y se puede adaptar a distintos ámbitos. Además, resulta más intuitiva que otros modelos que tienen un mayor nivel de complejidad. Es una técnica que requiere tener acceso a datos históricos que hayan sido correctamente documentados. Se puede caer en un sesgo si el proyecto seleccionado para la analogía no es adecuado.

Para reducir los errores en la aplicación de este método, se recomienda utilizar métricas objetivas y bases de datos internas, y aplicar técnicas como la normalización de características clave o el ajuste por factores contextuales (Clemmons, 2006).

La técnica de estimación por analogía presenta 2 dificultades principales. La primera es la selección de los atributos relevantes. Es con estos atributos que se determina la similitud entre proyectos. Algunos atributos a considerar pueden ser la tecnología a utilizar, el tamaño del software, el equipo a asignar, etc. La segunda dificultad que presenta es la medición de la similitud entre proyectos. No hay un consenso universal sobre qué medida de similitud es la más adecuada. Se suelen usar medidas como la distancia euclidiana.

#### 4.4.1.2 Estimación por descomposición

La estimación por descomposición consiste en dividir el proyecto en partes más pequeñas y manejables (normalmente tareas o módulos) y luego estimar el esfuerzo requerido para cada una de ellas individualmente. Finalmente, estas estimaciones se suman para obtener el esfuerzo total del proyecto.

Este enfoque se basa en la idea de que los errores de estimación tienden a ser menores cuando se trabaja con unidades de menor tamaño, ya que son más fáciles de

comprender y evaluar. Es especialmente común en metodologías tradicionales y también puede integrarse con técnicas ágiles para estimar sprints o tareas específicas (Santos, 2014).

Según Jørgensen (2004), este método es frecuentemente combinado con el juicio de expertos: los especialistas analizan cada componente por separado, aplicando su experiencia para realizar ajustes específicos según su complejidad, interdependencias y riesgos asociados.

La técnica de estimación otorga un mayor nivel de detalle y control sobre las estimaciones. Además, permite asignar recursos más precisamente y facilita la creación de un plan de trabajo. El problema de este método es que requiere un grado elevado de conocimiento del alcance funcional del proyecto desde el inicio y también puede requerir mucho tiempo si el proyecto es grande.

#### 4.4.1.3 COCOMO

El modelo COCOMO (*Constructive Cost Model*<sup>6</sup>) es uno de los enfoques algorítmicos más conocidos y utilizados para estimar el esfuerzo y la duración de proyectos de software. Fue desarrollado originalmente por Barry W. Boehm en 1981 y ha tenido sucesivas actualizaciones, siendo la más relevante COCOMO II.

A diferencia de los métodos basados en juicio o analogía, COCOMO se basa en una fórmula matemática que relaciona el tamaño estimado del software (expresado en líneas de código fuente (KLOC)) con el esfuerzo requerido, a través de ecuaciones calibradas empíricamente. La versión básica del modelo COCOMO establece:

$$\text{Esfuerzo (persona - mes)} = a * (\text{KLOC})^b$$

Donde a y b son constantes determinadas según el tipo de proyecto (orgánico, semiacoplado o empotrado).

---

<sup>6</sup> *Constructive Cost Model: Modelo constructivo de costos.*

#### 4.4.1.4 COCOMO II

El modelo COCOMO II fue desarrollado por Barry Boehm como una evolución del modelo original, con el objetivo de adaptarse a los cambios tecnológicos y metodológicos de la industria del software. A diferencia de otros enfoques, COCOMO II estima el esfuerzo de desarrollo a partir del tamaño del producto y de múltiples factores técnicos y organizacionales, expresando el resultado en meses-persona (PM) (Gómez et al., s.f.).

COCOMO II está compuesto por tres modelos de estimación, que se utilizan según la etapa del ciclo de vida en que se encuentre el proyecto y el nivel de información disponible:

- Composición de aplicaciones: Se utiliza en los desarrollos de software en etapa de prototipo. Para la estimación se utiliza Puntos Objeto.
- Diseño temprano: Se enfoca en estimar los costos de un proyecto antes de tener su arquitectura.
- Post-arquitectura: Se aplica cuando la arquitectura está definida. Usa líneas de código (SLOC) o puntos función, 17 multiplicadores de esfuerzo y 5 factores de escala.

Los 3 modelos se adaptan tanto a las necesidades de cada sector, como a la cantidad y el tipo de información disponible en cada etapa del ciclo de vida del desarrollo (Gómez et al., s.f.).

Fórmula general del esfuerzo (PM):

$$PM_{estimado} = A * (KSLOC)^B * \prod_{i=1}^{17} EM_i$$

Donde:

- A = Constante que refleja los efectos lineales según la variación de tamaño (A = 2.94).
- KSLOC = Miles de líneas de código fuente.

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

- B = Factor exponencial que incluye características relacionadas con las economías y deseconomías de escala producidas cuando un proyecto aumenta su tamaño.
- EMI = Multiplicadores de esfuerzo para 17 factores clasificados en 4 categorías: producto, plataforma, personal y proyecto.

El factor exponencial B se calcula como:

$$B = 1.01 + 0.01 * \sum_{i=1}^5 W_j$$

Donde los factores de  $W_j$  corresponden a:

- PREC: Precedencia del proyecto.
- FLEX: Flexibilidad de requerimientos.
- RESL: Resolución de riesgo.
- TEAM: cohesión del equipo.
- PMAT: Madurez del proceso.

Cada uno puede tomar un valor de Muy Bajo a Extra Alto según la siguiente imagen.

Factor de Escala $W_j$	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
<b>Precedencia PREC</b>	Completamente sin precedentes	Ampliamente sin precedentes	Algún precedente	Generalmente Familiar	Ampliamente Familiar	Completamente Familiar
<b>Flexibilidad en el desarrollo FLEX</b>	Rígurosa	Relajación Ocasional	Alguna Relajación	Conformidad en general	Alguna Conformidad	Metas generales
<b>Arquitectura/ Resolución de riesgo RESL</b>	Poca (20%)	Alguna (40%)	Siempre (60%)	Generalmente 75%)	Principalmente (90%)	Completo (100%)
<b>Cohesión de equipo TEAM</b>	Interacciones difíciles	Interacciones con alguna dificultad	Interacciones básicamente cooperativas	Ampliamente Cooperativas	Altamente Cooperativas	Interacciones Sin Fisuras
<b>Madurez del proceso PMAT</b>						

*Ilustración 12: Factores de escala. Fuente: (Gómez et al., s.f.)*

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Para el cálculo de la madurez del proceso (PMAT) se tienen en cuenta las dieciocho áreas de procesos claves (KPAs) del modelo del Software Engineering Institute.

Áreas de Procesos Claves	Casi Siempre (90%)	A menudo (60-90%)	La mitad de las veces (40-60%)	Ocasionalmente (10-40%)	Casi nunca (<10%)	No se aplica	No se conoce
Administración de Requerimientos							
Planificación del Proyecto de Software							
Seguimiento y supervisión del Proyecto de Software							
Administración de Subcontratos							
Aseguramiento de la Calidad							
Administración de la Configuración							
Objetivo del Proceso de Organización							
Definición del Proceso de Organización							
Programa de Entrenamiento							
Administración Integrada de Software							
Ingeniería del Producto							
Coordinación entre Grupos							
Revisión por Pares							
Administración Cuantitativa							
Administración de la Calidad							
Prevención de Defectos							
Administración de las Tecnologías de Cambio							
Administración de los Procesos de Cambio							

*Ilustración 13: Áreas clave del proceso. Fuente: (Gómez et al., s.f.)*

Los multiplicadores de esfuerzo están determinados por la siguiente tabla:

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

	Factor	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
Producto	RELY	Inconvenientes insignificantes, que afectan solamente a los desarrolladores	Mínimas pérdidas al usuario, fácilmente recuperables	Pérdidas moderadas al usuario recuperables sin grandes inconvenientes	Pérdida financiera elevada o inconveniente humano masivo	Vida humana en riesgo	
	DATA		DB bytes/Pgm SLOC < 10	10<=D/P<100	100<=D/P<1000	D/P > 1000	
	RUSE		Ningún componente reusable	Reusable dentro del mismo proyecto	Reusable dentro de un mismo programa	Reusable dentro de una misma línea de productos	Reusable dentro de múltiples líneas de producto
	DOCU	Muchas necesidades del ciclo de vida sin cubrir	Algunas necesidades del ciclo de vida sin cubrir	Necesidades del ciclo de vida cubiertas en su justa medida	Necesidades del ciclo de vida cubiertas ampliamente	Necesidades del ciclo de vida cubiertas excesivamente	
Plataforma	TIME			Uso de <= 50% del tiempo de ejecución disponible	70%	85%	95%
	STOR			Uso de <= 50% del porcentaje total de almacenamiento	70%	85%	95%
	PVOL		Un cambio principal cada 12 meses. Un cambio menor todos los meses	Cambio principal cada 6 meses. Cambio menor cada 2 semanas	Cambio principal cada 2 meses. Cambio menor uno por semana	Cambio principal cada 2 semanas. Cambio menor cada 2 días	
Personal	ACAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil	
	PCAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil	
	PCON	48 % por año	24 % por año	12 % por año	6% por año	3 % por año	
	AEXP	<= 2 meses	<= 6 meses	1 año	3 años	6 años	
	PEXP	<= 2 meses	<= 6 meses	1 año	3 años	6 años	
	LTEX	<= 2 meses	<= 6 meses	1 año	3 años	6 años	
Proyecto	TOOL	Herramientas que permiten editar, codificar, depurar	Herramientas simples con escasa integración al proceso de desarrollo	Herramientas básicas, integradas moderadamente	Herramientas robustas y maduras, integradas moderadamente	Herramientas altamente integradas a los procesos, métodos y reuso	
	SITE Ubicación Espacial	Internacional	Multi-ciudad y multi-compañía	Multi-ciudad o multi-compañía	Misma ciudad o área metropolitana	Mismo Edificio o complejo	Completamente Centralizado
	SITE Comunicación	Algún teléfono, mail	Teléfonos individuales, FAX	Email de banda angosta	Comunicaciones electrónicas de banda ancha	Comunicaciones electrónicas de banda ancha, ocasionalmente videoconferencia	Multimedia Interactiva
	SCED	75% del nominal	85% del nominal	100% del nominal	130% del nominal	160% del nominal	

Ilustración 14: Multiplicadores de esfuerzo. Fuente: (Gómez et al., s.f.)

Para el cálculo del tiempo total en meses se utiliza la siguiente fórmula:

$$TDEV = 3.67 * (PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$$

Dónde PM\* es el esfuerzo estimado sin tener en cuenta el multiplicador SCED.

La ventaja del método es que está calibrado a partir de datos empíricos. Un inconveniente que presenta es que requiere de información documentada detallada para poder realizar su aplicación de manera efectiva.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

### 4.4.1.5 Puntos de función

La técnica de estimación mediante casos de uso, conocida como *Use Case Points* (UCP), es un método estructurado para calcular el esfuerzo de desarrollo de un proyecto de software. Según Karner (1993), el método analiza el contenido de los casos de uso, sus actores asociados y diversos factores técnicos y ambientales que influyen en el esfuerzo necesario para completar el proyecto.

Se basa en la fórmula general:

$$UCP = UUCP * TCF * EF$$

Donde:

- UUCP = *Unadjusted Use Case Points* (puntos sin ajustar).
- TCF = *Technical Complexity Factor* (factor técnico).
- EF = *Environmental Factor* (factor ambiental).

Para el cálculo de los UUCP se utilizan las siguientes 2 tablas:

Tabla 1: Peso de actores

Complejidad	Definición	Peso
Simple	Un actor es simple si representa otro sistema con una interfaz de aplicación programada definida.	1
Promedio	Un actor es promedio si es: <ol style="list-style-type: none"><li>1. Una interacción con otro sistema a través de un protocolo.</li><li>2. Una interacción humana con una línea terminal.</li></ol>	2
Complejo	Un actor es complejo si interactúa a través de una interfase gráfica.	3

Fuente: (Karner, 1993).

Tabla 2: Peso de casos de uso

Complejidad	Definición	Peso
Simple	Un caso de uso es simple si tiene 3 o menos transacciones.	5

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

Martin, M.

Promedio	Un caso de uso es promedio si tiene entre 3 y 7 transacciones.	10
Complejo	Un caso de uso es promedio si tiene más de 7 transacciones.	15

Fuente: (Karner, 1993)

$$UUCP = \sum_{i=1}^6 n_i * W_i$$

Donde n es el número de actores o casos de uso y W el peso de cada uno.

Para el cálculo del TCF se evalúan 13 factores técnicos

Tabla 3: Factores de complejidad técnica

Fi	Factor que contribuye a la complejidad	Wi
F1	Sistemas distribuidos	2
F2	Aplicación de objetivos de desempeño	1
F3	Eficiencia de usuario final	1
F4	Procesado interno complejo	1
F5	Reusabilidad	1
F6	Facilidad de instalación	0.5
F7	Facilidad de operación	0.5
F8	Portabilidad	2
F9	Intercambiabilidad	1
F10	Concurrencia	1
F11	Aspectos especiales de seguridad	1
F12	Acceso de terceras partes	1
F13	Uso especial de instalaciones de entrenamiento	1

Fuente: (Karner, 1993)

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

Martin, M.

Tabla factores complejidad técnica.

$$TCF = 0.6 + 0.01 \sum_{i=1}^{13} F_i * W_i$$

Para el cálculo del factor ambiental EF se evalúan 8 factores:

*Tabla 4: Factores ambientales*

Fi	Factores que contribuyen a la eficiencia	Wi
F1	Familiaridad con Objectory	1.5
F2	Trabajadores medio tiempo	-1
F3	Capacidad de analistas	0.5
F4	Experiencia	0.5
F5	Experiencia orientada a objetos	1
F6	Motivación	1
F7	Dificultad de lenguaje de programación	-1
F8	Requerimientos de estabilidad	2

*Fuente: (Karner, 1993)*

$$EF = 1.4 - 0.03 \sum_{i=1}^8 F_i * W_i$$

En ambos casos (TCF y EF) los valores de  $F_i$  toman valores de 0 a 5 según su relevancia.

Finalmente, una vez calculado el UCP se lo multiplica por un valor medio de esfuerzo por punto. El autor propone utilizar 20 horas por UCP como estimación inicial.

#### 4.4.1.6 Price to win

La estrategia conocida como *Price to Win* (Precio para ganar) consiste en ajustar el esfuerzo estimado en función del presupuesto disponible o del precio que el cliente está dispuesto a pagar, en lugar de realizar una estimación técnica basada en el alcance funcional del software.

## 4.4.2 Técnicas de estimación en metodologías ágiles

### 4.4.2.1 *Planning poker*

Planning poker es una técnica de estimación diseñada para acelerar la planificación de historias de usuario <sup>7</sup>involucrando a todo el equipo. Se aplica principalmente durante la planificación de la versión para obtener una visión global del esfuerzo requerido sin detenerse en detalles excesivos. El objetivo no es obtener una estimación perfecta sino situarse en una posición en donde el valor de la estimación supere el costo de obtenerla (Cohn, 2005).

Primero cada participante recibe un mazo de cartas con valores no lineales (por ejemplo: 0, 1, 2, 3, 5, 8, 13, 20, 40 y 100), estos números representan unidades de tamaño. Luego el moderador presenta al equipo una historia de usuario y aclara las dudas esenciales. Cada participante elige una carta de su mazo que refleja la estimación que considera adecuada. Una vez que todos han seleccionado su carta, se revelan simultáneamente los valores; si coinciden, la estimación queda registrada y se avanza a la siguiente historia. En caso de existir diferencias los participantes con las estimaciones más altas y más bajas explican brevemente sus criterios. Luego se repite la votación hasta alcanzar una convergencia razonable.

### 4.4.2.2 T-shirt sizing

En la estimación por talla de camisetas se emplean categorías intuitivas —XS, S, M, L y XL— para asignar un tamaño relativo a cada historia o elemento a estimar. Es similar al Planning poker en cuanto a que el equipo revisa cada historia y, tras una breve discusión para aclarar su alcance y complejidad, acuerda de forma colectiva si resulta “Extra pequeña”, “Pequeña”, “Mediana”, “Grande” o “Extra grande”. A cada talla se le asocia un rango predefinido de esfuerzo (por ejemplo, XS = 1–2 puntos, S = 3–5, M = 8–13, L = 20–40, XL > 40), de modo que el equipo pueda convertir rápidamente esas categorías en estimaciones

---

<sup>7</sup> Historia de usuario: Descripción breve y en lenguaje natural de una característica de software, redactada desde la perspectiva del usuario final.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

cuantitativas cuando sea necesario. Al usar una escala de rangos amplios, el método facilita obtener estimaciones iniciales rápido ayudando al equipo a interiorizar la comparación relativa entre historias y evita el sesgo de apresurarse a definir cifras muy precisas antes de disponer de suficiente información (Mallidi & Sharma, 2021).

Luego del relevamiento de técnicas de estimación empleadas en la industria, se concluye que Planning Poker es la opción más factible para aplicar en la empresa. Los modelos paramétricos como COCOMO y los puntos de función fueron descartados por requerir calibración y series históricas detalladas que hoy la organización no dispone, lo que implica un mayor el esfuerzo inicial y el riesgo de sesgos de ajuste. Además, como se mencionó anteriormente no existe evidencia que indique una superioridad en cuanto a la precisión en la estimación de estos modelos. En contraste, Planning Poker no demanda datos históricos para comenzar y permite mejorar progresivamente a medida que se dispone de información.

La elección se basa en los siguientes aspectos:

- **Compatibilidad metodológica:** Es una técnica compatible con prácticas ágiles lo que favorece la estimación relativa por puntos de historia y su proyección mediante velocidad.
- **Bajo costo de adopción:** requiere una preparación mínima, se apoya en dinámicas ya presentes en los equipos y evita esfuerzos de modelado y calibración iniciales.
- **Aprovechamiento del conocimiento experto:** capitaliza la experiencia acumulada del personal que en la actualidad no es aprovechada.
- **Rapidez y flexibilidad:** posibilita ciclos de estimación ágiles y oportunos, adecuados a contextos con requisitos cambiantes o incertidumbre.

- Reducción de sesgos individuales: la revelación simultánea de estimaciones y la discusión guiada atenúan efectos de anclaje y jerarquía, mejorando la calidad del acuerdo alcanzado.
- Escalabilidad y aprendizaje: funciona tanto en proyectos nuevos como en mejoras y el equipo aprende en cada instancia de estimación mejorando progresivamente los resultados obtenidos.
- Trazabilidad y transparencia: deja registro de los supuestos relevantes emergentes en las discusiones (alcance, dependencias, riesgos), facilitando auditorías internas lo que apoya una cultura de mejora continua.

Dada la falta de datos calibrados, la necesidad de agilidad operativa y la disponibilidad de conocimiento del personal, Planning Poker ofrece la mejor relación costo-beneficio ya que habilita estimaciones rápidas, consistentes con el marco ágil de la empresa, y promueve un proceso de mejora continua soportado por métricas y retroalimentación sistemática.

#### **4.5 Diseñar una propuesta de mejora para el proceso de estimación de duración de proyectos de software, basada en los hallazgos del análisis.**

Los datos obtenidos en los capítulos previos muestran una falta de precisión en las estimaciones. El análisis realizado indica una desviación media de alrededor de un 35% lo cual refleja un bajo cumplimiento del estándar interno de la organización.

Estas evidencias justifican una intervención metodológica que aborde: (i) la reducción de sesgos cognitivos que afectan al juicio experto, (ii) la mejora en la calidad y completitud de los requerimientos tempranos, (iii) la institucionalización de métricas para aprender de la experiencia, y (iv) un ciclo de mejora continua.

En este capítulo se realizará una propuesta de mejora dividida en 4 pilares:

- Incorporación de una de las técnicas de estimación investigadas.
- Desarrollo de un listado de verificación previo a reuniones con clientes.

- Desarrollo y definición de *KPIs*<sup>8</sup> para tener una medición y control de las estimaciones.
- Implementación de ciclo *PDCA*<sup>9</sup> en búsqueda de la mejora continua.

#### 4.5.1 Checklist de datos a relevar

El *checklist*<sup>10</sup> se adopta como un instrumento estandarizado cuyo objetivo principal es asegurar la completitud mínima de requerimientos, la comparabilidad entre proyectos y la trazabilidad de supuestos al momento de estimar. Su aplicación responde a la importancia de tener requisitos claros y procesos repetibles para mejorar la planificación y la precisión de las estimaciones.

Se estructura en tres etapas:

- Preparación previa a la reunión, donde el equipo interno reúne antecedentes, plantillas y supuestos a validar.
- Aplicación durante la reunión con el cliente, destinada a relevar de forma ordenada los datos esenciales para la estimación.
- Generación de salidas, que consolidan el acta de alcance preliminar, el *backlog*<sup>11</sup> inicial y los compromisos pendientes.

Así, se busca instalar un proceso más uniforme que reduzca equivocaciones y omisiones al momento de la comunicación con el cliente.

##### 4.5.1.1 Preparación previa: conceptos y especificaciones a relevar

Antes de la reunión, el equipo debe determinar los aspectos necesarios a relevar para obtener información útil y completa. Se sugiere preparar:

- Lista de objetivos y métricas de éxito.

---

<sup>8</sup> *KPIs*: Indicadores clave de desempeño.

<sup>9</sup> *PDCA*: Proceso de mejora continua que consiste en planificar, hacer, verificar y actuar.

<sup>10</sup> *Checklist*: Listado de verificación.

<sup>11</sup> *Backlog*: Lista priorizada de tareas, requisitos o elementos de trabajo necesarios para el desarrollo de un producto.

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

- Listado de requisitos funcionales típicos por tipo de proyecto.
- Datos, interfaces y dependencias habituales para validar con el cliente.
- Requisitos no funcionales (rendimiento, seguridad, disponibilidad, auditoría).
- Criterios de *Definition of Ready*<sup>12</sup> (DoR) aplicables a las historias a estimar.

### 4.5.1.2 Checklist durante la reunión

- Objetivos.
- Restricciones regulatorias.
- Alcance funcional y criterios de aceptación.
- Actores/usuarios clave y su disponibilidad para validaciones.
- Procesos actuales y cambios esperados.
- Datos y calidad: fuentes, volumen, migraciones, retención, privacidad.
- Interfaces/dependencias y responsables externos.
- Requisitos no funcionales: rendimiento, seguridad, disponibilidad, usabilidad, auditoría.
- Entorno tecnológico: plataformas, versiones, políticas de cambios y congelamientos.
- Calendario.
- Riesgos conocidos, supuestos críticos e incertidumbres.

### 4.5.1.3 Salidas

La reunión debe producir:

- Acta de alcance preliminar.
- Backlog inicial con historias y criterios de aceptación.
- Inventario de interfaces y restricciones.
- Riesgos/supuestos priorizados.
- Plan de información pendiente.

---

<sup>12</sup> *Definition of Ready*: Criterios que una historia de usuario debe cumplir para empezar a trabajar.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

Una historia estará 'Lista' (DoR) si cuenta con objetivo claro, criterios de aceptación verificables, dependencias identificadas, y estimabilidad razonable.

La aplicación del checklist no solo mejora la recolección completa de información crítica para la estimación, sino que también promueve un proceso de aprendizaje organizacional. Al preparar y validar cada uno de los puntos, el equipo de trabajo debe revisar antecedentes, comprender con mayor detalle los procesos de negocio y familiarizarse con los requisitos técnicos y regulatorios más relevantes. Esta práctica fortalece el conocimiento y favorece una comprensión más integral del proyecto antes de su estimación, lo que reduce la probabilidad de interpretaciones erróneas en etapas posteriores.

### **4.5.2 Planning Poker**

Se propone la incorporación de la técnica de estimación Planning Poker. Esto se debe a una clara necesidad de superar las limitaciones detectadas en las prácticas actuales de juicio experto caracterizadas por la presencia de sesgos individuales. Su aplicación permitirá introducir un enfoque estructurado, participativo y transparente buscando reducir los sesgos y aumentar la confiabilidad de las estimaciones.

Para implementar esta técnica es necesario introducir cambios en la dinámica de trabajo del equipo. En primer lugar, hay que establecer un entorno de estimación seguro y colaborativo, en el cual los integrantes puedan expresar libremente sus valoraciones. La estimación debe entenderse como una actividad técnica y no como una negociación. Esto debe lograrse para minimizar el riesgo de que los participantes ajusten sus números para cumplir expectativas o evitar cuestionamientos.

Asimismo, se deberá trabajar sobre los sesgos más comunes que afectan a los estimadores. Uno de los más frecuentes es la sobreestimación defensiva, que surge cuando los participantes inflan deliberadamente los valores con el objetivo de protegerse de retrasos o dificultades imprevistas. Este comportamiento genera distorsiones significativas y reduce la utilidad de la estimación. Para contrarrestar este sesgo, se propone:

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

*Martin, M.*

- Separar claramente la instancia de estimación de la de compromiso. La primera refleja la magnitud del esfuerzo mientras que la segunda define cuánto puede asumirse en función de la capacidad disponible.
- Registrar y discutir explícitamente los supuestos y riesgos identificados durante la estimación, de manera que las preocupaciones de los participantes se incorporen a la planificación sin necesidad de inflar las cifras.
- Capacitar al equipo en sesgos cognitivos y técnicas de estimación colaborativa, brindando ejemplos prácticos de cómo estos sesgos impactan en los resultados.

La introducción de Planning Poker no es solo un cambio metodológico, sino también como un cambio cultural y organizacional orientado a fomentar la transparencia, reducir sesgos individuales y consolidar un aprendizaje colectivo.

### 4.5.3 KPIs

Para fomentar el aprendizaje y poder llevar un registro del proceso de estimación y sus resultados se propone el desarrollo de una serie de indicadores.

El primero de ellos es el MRE (magnitud del error relativo).

$$MRE = \frac{|Esfuerzo\ real - Esfuerzo\ estimado|}{Esfuerzo\ real} * 100\%$$

Mide la diferencia entre lo real y lo estimado. Un valor bajo en este indicador significa una buena estimación mientras que un valor alto indica errores al estimar. Se utiliza para tener una visión global de la desviación en las estimaciones.

El segundo indicador propuesto es similar al anterior, pero toma en cuenta el signo de la desviación para identificar tendencias a subestimar o sobrestimar. Se utiliza para identificar tendencias. Si el valor histórico obtenido en una empresa es de - 30% entonces se concluye que la empresa tiene a subestimar y se pueden realizar correcciones como por ejemplo agregar horas extra en cada entrega. El indicador tiene el nombre de RE (error porcentual relativo).

$$RE = \frac{\text{Esfuerzo estimado} - \text{Esfuerzo real}}{\text{Esfuerzo real}} * 100\%$$

- Si el  $RE > 0$  → sobreestimación.
- Si el  $RE < 0$  → subestimación.

Al igual que el MRE se busca que este indicador se encuentre lo más cerca del 0% posible lo cual implica poca diferencia entre lo estimado y lo real.

El tercer indicador se conoce como Pred(25). Es el porcentaje de estimaciones cuya desviación entra dentro del  $\pm 25\%$  del valor real.

$$Pred(25) = \frac{\text{Número de estimaciones con } \left| \frac{E_i - \hat{E}_i}{E_i} \right| \leq 0.25}{n} * 100$$

Con el objetivo de analizar la eficiencia del checklist se propone la medición del índice de omisiones críticas (IOC). Mide la proporción de incidentes, riesgos o problemas detectados en retrospectiva o en fases posteriores del proyecto que deberían haber sido relevados en el checklist inicial pero no lo fueron.

$$IOC = \frac{\text{Número de omisiones}}{\text{Número de ítems críticos definidos en el checklist}} * 100$$

Se entiende por ítem crítico a todo elemento de información cuya ausencia o error en la etapa de recolección inicial compromete de manera significativa la validez de la estimación. Pueden ser historias, funcionalidades clave, requisitos no funcionales esenciales, dependencias externas, riesgos, supuestos o restricciones.

Un IOC alto indica que, aunque el checklist se completó formalmente, no se abordan con suficiente rigor los puntos realmente críticos.

## 4.5.4 Ciclo PDCA para la mejora del proceso de estimación

### 4.5.4.1 Plan (planificar)

El objetivo de esta etapa es preparar las condiciones para estimar con mayor precisión, sustentándose en tres pilares: checklist, Planning Poker y KPIs.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

1. Preparación del checklist: Se adopta como instrumento estandarizado para asegurar la completitud mínima de requerimientos, habilitar comparabilidad entre proyectos y trazar supuestos de estimación. Se estructura en tres momentos: preparación previa, aplicación en reunión con el cliente y generación de salidas formales. En la preparación se definen: objetivos y métricas de éxito; requisitos funcionales típicos; datos, interfaces y dependencias a validar; requisitos no funcionales; y criterios de Definition of Ready (DoR) aplicables a las historias a estimar.
2. Planificar la sesión de Planning Poker. Se selecciona esta técnica ágil como sustituto del juicio experto individual para reducir sesgos, fomentar la discusión y separar explícitamente estimación de esfuerzo de compromiso de plazo.
3. Definir objetivos de KPIs. Se establecen las métricas a utilizar (MRE, RE, Pred(25) e IOC) y sus metas.

### **4.5.4.2 Do (hacer)**

1. Aplicación del checklist en la reunión con el cliente. Durante la reunión se relevan los aspectos considerados en el checklist. Las salidas son: acta de alcance preliminar, backlog inicial con historias y criterios de aceptación, inventario de interfaces/restricciones, riesgos/supuestos priorizados y plan de información pendiente.
2. Documentación y criterio DoR. Cada historia queda "Lista (DoR)" cuando posee objetivo claro, criterios de aceptación verificables, dependencias identificadas y estimabilidad razonable. Esto reduce equivocaciones y omisiones tempranas de información.
3. Uso del checklist como insumo de Planning Poker. Las historias que cumplen DoR, construidas a partir de la información consolidada por el checklist, se llevan a la sesión

## Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

*Martin, M.*

de Planning Poker para estimación colaborativa, mitigando sesgos del juicio experto y acelerando la planificación.

### 4.5.4.3 *Check* (verificar)

1. Cálculo de KPIs del proyecto. Concluida la ejecución (o en cortes definidos del proyecto), se calculan los KPIs definidos previamente.
2. Parámetro de aceptabilidad. Se considera que un proyecto cumple el estándar interno si  $MRE \leq 15\%$  y  $|RE| \leq 15\%$ . Pred(25) y IOC se evalúan comparando contra su línea base histórica para verificar mejoras.

### 4.5.4.4 *Act* (actuar)

1. Estudio de causa raíz ante desvíos. Si algún KPI es no aceptable se inicia un análisis de causa raíz utilizando las categorías ya identificadas mediante la herramienta de Ishikawa.
2. Acciones de mejora focalizadas. En función del diagnóstico, se actúa sobre los aspectos correspondientes: reforzar el checklist donde aparecieron omisiones, ajustar la dinámica de Planning Poker o introducir ajustes operativos que reduzcan fuentes de variabilidad detectadas.
3. Documentación de resultados positivos. Cuando los KPIs alcanzan o superan las metas (reducción de MRE/|RE| hacia 15% y mejora de Pred(25) y IOC), se documenta lo actuado para consolidar aprendizaje organizacional y retroalimentar el proceso de planificación futura.

## 5. DISCUSIÓN

La evaluación de las prácticas de estimación en la empresa EMMSA, realizada en el marco de este trabajo, ha arrojado resultados que confirman el desafío recurrente que enfrenta el sector de desarrollo de software: la estimación precisa de la duración de los proyectos. Los hallazgos del Objetivo 2 revelaron una magnitud promedio de desvío absoluto cercana al 35% en las series limpias de proyectos. Este índice está significativamente por debajo del estándar interno de la organización, que aspira a una desviación menor al 15%. Solo alrededor de un 22% a 24% de los proyectos cumplen con este estándar de buena estimación.

El principal hallazgo de la investigación (el desvío promedio del 35%) se contextualiza mediante el análisis de la técnica de estimación utilizada actualmente por EMMSA: el juicio de expertos. La literatura especializada, citada en el Marco Teórico, reconoce que el juicio de expertos es la estrategia más utilizada en el software, especialmente cuando los expertos poseen conocimientos relevantes. Sin embargo, el mismo marco teórico advierte que los expertos pueden estar fuertemente sesgados y tender al exceso de optimismo (*wishful thinking*).

El análisis de la mediana del desvío en los proyectos de mejora (+23%) muestra una clara tendencia a subestimar el esfuerzo, lo que es consistente con el sesgo optimista que Jørgensen (2004) identifica como una debilidad inherente al juicio experto no estructurado. Este sesgo lleva a la omisión de tareas o a la minimización de la complejidad real. Además, la dispersión observada no se concentra en unos pocos proyectos atípicos (solo el 30% de los proyectos explican el 80% del impacto total, lo que no cumple estrictamente con el principio de Pareto), sino que los errores de estimación se distribuyen a lo largo de los proyectos. Esto sugiere que el problema es de naturaleza sistémica y metodológica, y no solo un evento aislado atribuible a proyectos de alta incertidumbre, aunque el marco teórico señala

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

que las estimaciones de modelos son más precisas que las de expertos cuando la incertidumbre es alta.

La propuesta de mejora diseñada (Objetivo 5) aborda los cuatro factores principales que, según el diagrama de Ishikawa y la experiencia de los referentes, causan la variabilidad: Factores funcionales (requisitos), Factores de gestión (planeación), Factores tecnológicos y Factores organizacionales.

La identificación de factores clave (Objetivo 3) reveló que la falta de involucramiento temprano de usuarios clave, la mala comunicación, la omisión de tareas no previstas y los cambios tecnológicos/normativos son causas recurrentes de desvío.

Para contrarrestar esto, se diseñó el checklist de datos a relevar. Esta herramienta responde al principio de estimación por expertos que recomienda usar listas de verificación para recordar tareas y considerar eventos inesperados, mejorando la consistencia y fomentando la consideración de alternativas. Al estandarizar la recolección de información crítica (objetivos, alcance, dependencias, riesgos, entorno tecnológico), el checklist busca asegurar la completitud mínima de los requisitos tempranos, una práctica crucial para la precisión de la estimación.

Se eligió la técnica Planning Poker, una técnica ágil investigada en el Objetivo 4, como sustituto del juicio de expertos individual. Si bien el juicio de expertos es la técnica actual, es propensa a sesgos individuales. Planning Poker se justifica porque:

1. Está diseñada para acelerar la planificación y reducir los sesgos individuales al forzar la discusión y el consenso grupal simultáneo (evitando el efecto anclaje que ocurre cuando las estimaciones se ven influenciadas por otras, como sucede en métodos Top-down y Bottom-up independientes antes de la comparación).

2. Promueve la transparencia y la colaboración, separando explícitamente la instancia de estimación (esfuerzo) de la instancia de compromiso (plazo). Esto es vital

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

para mitigar el sesgo optimista generado por el wishful thinking que se busca evitar al establecer metas de estimación en conflicto.

3. La técnica se alinea con el modelo de trabajo de EMMSA, que ya integra principios de metodologías ágiles para el desarrollo de software.

Finalmente, la propuesta de incluir KPIs (MRE, RE, Pred(25) e IOC) y el ciclo PDCA tiene como objetivo instaurar un proceso de mejora continua basado en la experiencia, como lo recomienda el marco teórico. La medición del MRE y el RE permite identificar errores sistemáticos (como la subestimación) y aplicar correcciones. El Índice de Omisiones Críticas (IOC) se propone específicamente para medir la eficiencia del checklist en la detección temprana de problemas, lo cual es fundamental para el feedback y el aprendizaje organizacional.

La propuesta busca transformar un proceso de estimación basado en un juicio experto, históricamente propenso a errores del 35% por causas de sesgos y requisitos deficientes, en un proceso estructurado, colaborativo y medible, utilizando herramientas de Ingeniería Industrial (checklist, KPIs, PDCA) aplicadas al dominio de la Ingeniería de Software.

## 6. CONCLUSIONES

El objetivo general de este Trabajo Final, Evaluar y analizar las técnicas de estimación utilizadas por una empresa tecnológica marplatense y diseñar una propuesta de mejora en el proceso, ha sido logrado mediante un estudio de caso que combinó el diagnóstico empírico con el diseño de una solución integral.

La investigación cumplió exitosamente con los cinco objetivos específicos, generando las siguientes conclusiones clave:

1. Relevamiento de la Técnica Actual (Objetivo 1): Se constató que la empresa EMMSA utiliza actualmente una técnica de estimación basada en el juicio de expertos, la cual se apoya en la vasta trayectoria profesional de su equipo técnico. Esta técnica, aunque flexible, mostró ser susceptible a errores sistémicos.

2. Análisis de Desvíos Significativos (Objetivo 2): El análisis de 77 proyectos finalizados reveló que la precisión de las estimaciones es inadecuada. La magnitud promedio del desvío absoluto se mantiene cercana al 35%. Este resultado es significativamente superior al estándar interno de la empresa, que aspira a una desviación menor al 15%. Solo entre el 22% y el 24% de los proyectos cumplieron con este estándar de buena estimación. Adicionalmente, se observó una tendencia a subestimar el esfuerzo en los proyectos de mejora (mediana de desvío de +23%).

3. Determinación de Factores Clave de Variabilidad (Objetivo 3): Los principales factores que contribuyen a los desvíos se determinaron como la falta de involucramiento temprano de usuarios clave, la modificación de plataformas tecnológicas, el incumplimiento de plazos por parte del cliente, cambios normativos y la omisión de tareas no previstas. Estos factores refuerzan la necesidad de estandarizar la entrada de información y mitigar el sesgo optimista.

## **Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

4. Investigación y Comparación de Metodologías (Objetivo 4): Se investigaron y compararon diversas técnicas, incluyendo modelos algorítmicos (COCOMO, COCOMO II, Puntos de función) y métodos ágiles (Planning Poker, T-shirt sizing). La revisión se centró en evaluar su aplicabilidad al contexto de EMMSA, orientando la decisión hacia técnicas que mitigaran los sesgos inherentes al juicio experto.

5. Diseño de la Propuesta de Mejora (Objetivo 5): Se diseñó una propuesta de mejora estructurada en cuatro pilares para abordar las limitaciones encontradas.

- Se propuso la incorporación de Planning Poker como técnica colaborativa para estructurar el juicio de expertos y reducir los sesgos individuales que se observaron en los resultados históricos.

- Se diseñó un checklist de datos a relevar para asegurar la completitud mínima de los requerimientos tempranos, abordando directamente el factor de variabilidad relacionado con la omisión de tareas y la inadecuada definición de requisitos.

- Se definieron KPIs (MRE, RE, Pred(25) e IOC) para permitir la medición continua y el aprendizaje organizacional.

- Incorporación de un ciclo PDCA que englobe los 3 pilares previos para fomentar la mejora continua.

### Implicaciones y Trabajo Futuro

El trabajo aportó una solución diseñada utilizando herramientas de Ingeniería Industrial, como el análisis de causa raíz y la definición de métricas de control (KPIs), aplicadas al sector de software.

Es fundamental señalar que este estudio se centró en el análisis de datos históricos y en el diseño formal de una propuesta de mejora. Por lo tanto, el alcance del trabajo no incluyó la fase de implementación ni la validación empírica de la propuesta.

Como trabajo futuro, se recomienda la ejecución de la propuesta diseñada:

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

1. Implementar la técnica *Planning Poker* junto con el checklist de datos a relevar en una serie de proyectos piloto.

2. Medir los KPIs propuestos (MRE, RE, Pred(25) e IOC) para cuantificar la reducción de la desviación promedio del 35% y validar la efectividad de las nuevas prácticas en la mejora de la precisión de las estimaciones.

## 7. BIBLIOGRAFÍA

- Clemmons, R. K. (2006). *Project Estimation With Use Case Points*. Diversified Technical Services. [https://www.researchgate.net/profile/Roy-Clemmons/publication/200036324\\_Project\\_Estimation\\_With\\_Use\\_Case\\_Points/links/00b7d518401cd4fd9b000000/Project-Estimation-With-Use-Case-Points.pdf](https://www.researchgate.net/profile/Roy-Clemmons/publication/200036324_Project_Estimation_With_Use_Case_Points/links/00b7d518401cd4fd9b000000/Project-Estimation-With-Use-Case-Points.pdf)
- Cohn, M. (2005). *Agile estimating and planning*. Prentice Hall Professional Technical Reference.
- Deming, W. E. (1986). *Out of the Crisis*. MIT Press.
- Gómez, A., López, M. d. C., Migani, S., & Otazú, A. (s.f.). *UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*. [https://www.academia.edu/4853590/UN\\_MODELO\\_DE\\_ESTIMACION\\_DE\\_PROYECTOS\\_DE\\_SOFTWARE](https://www.academia.edu/4853590/UN_MODELO_DE_ESTIMACION_DE_PROYECTOS_DE_SOFTWARE)
- Jørgensen, M. (2004). A Review of Studies on Expert Estimation of Software Development Effort. *Simula Research Laboratory*, 27. <https://web-backend.simula.no/sites/default/files/publications/SE.4.Joergensen.2004.c.pdf>
- Jørgensen, M. (2005). *Practical Guidelines for Expert-Judgment-Based Software Effort Estimation*. Simula Research Laboratory. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1438330>
- Jørgensen, M., & Teigen, K. H. (2002). *Uncertainty Intervals versus Interval Uncertainty: An Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects*. Simula Research Laboratory. <https://web-backend.simula.no/sites/default/files/publications/files/promac-final2.pdf>
- Karner, G. (1993). *Resource Estimation for Objectory Projects*. Objective Systems SF AB. [https://teraits.com/anhanguera/marcio/gcm/p\\_Karner\\_ResourceEstimationForObjectoryProjects.pdf](https://teraits.com/anhanguera/marcio/gcm/p_Karner_ResourceEstimationForObjectoryProjects.pdf)

**Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.**

*Martin, M.*

- Mallidi, R. K., & Sharma, M. (2021). *Study on Agile Story Point Estimation Techniques and Challenges*. International Journal of Computer Applications. [https://d1wqtxts1xzle7.cloudfront.net/119199895/mallidi-2021-ijca-921014-libre.pdf?1730097735=&response-content-disposition=inline%3B+filename%3DStudy\\_on\\_Agile\\_Story\\_Point\\_Estimation\\_Te.pdf&Expires=1746279563&Signature=YuTDEq1kda6lKddbBK1volHa8ge3A0VF8fnN](https://d1wqtxts1xzle7.cloudfront.net/119199895/mallidi-2021-ijca-921014-libre.pdf?1730097735=&response-content-disposition=inline%3B+filename%3DStudy_on_Agile_Story_Point_Estimation_Te.pdf&Expires=1746279563&Signature=YuTDEq1kda6lKddbBK1volHa8ge3A0VF8fnN)
- Pressman, R. S. (2010). *INGENIERIA DE SOFTWARE*. McGraw-Hill Interamericana de España S.L.
- Pressman, R. S. (2010). *INGENIERIA DE SOFTWARE*. McGraw-Hill Interamericana de España S.L.
- Ramos Cardozzo, D. (2016). *Desarrollo de Software: Requisitos, Estimaciones y Análisis. 2 Edición* (I. T. Campus Academy, Ed.). CreateSpace Independent Publishing Platform.
- Rodríguez Soria, P., & Martínez Herráiz, J. J. (2010). *Modelos Paramétricos de Estimación de Esfuerzo dentro del Proceso de Planificación de Proyectos Software*. Revista de procesos y métricas de las tecnologías de la información. <https://portalcientifico.uah.es/documentos/61567cdaf4a2be562344f59c?lang=gl>
- Santos, S. A. (2014). Estimación de proyectos de software pequeños basada en el juicio de expertos [Tesis de maestría]. In *Universidad Nacional de La Plata*.
- Shepperd, M., & Schofield, C. (1997). *Estimating Software Project Effort Using Analogies*. IEEE Transactions on Software Engineering. [https://www.researchgate.net/publication/3187904\\_Estimating\\_software\\_project\\_effort\\_using\\_analogies](https://www.researchgate.net/publication/3187904_Estimating_software_project_effort_using_analogies)
- Sommerville, I. (2011). *Software Engineering*. Pearson.

Evaluación y propuesta de mejora en las técnicas de estimación de la duración de proyectos de software: Estudio de caso en una empresa marplatense.

Martin, M.

## 8. ANEXO

Tabla 5: Tabla de proyectos

Descripción proyecto	Clase Proyecto	Fecha desde Plan	Fecha hasta Plan	Hs Plan	Fecha desde Real	Fecha hasta Real	Hs Real
App CEM	Proyecto Original	2019-05-01 00:00:00.000	2019-08-30 00:00:00.000	160	2019-02-25 00:00:00.000	2019-11-28 00:00:00.000	109
AdInsurance - Integracion con	Proyecto de Mejoras	2023-04-15 00:00:00.000	2023-10-30 00:00:00.000	1200	2023-04-12 00:00:00.000	2023-11-22 00:00:00.000	114.5
U-Sign - Historico	Proyecto de Mejoras	2021-09-01 00:00:00.000	2022-01-31 00:00:00.000	216	2022-03-30 00:00:00.000	2022-07-01 00:00:00.000	120.5
Gea OT Mobile	Proyecto Original	2018-02-01 00:00:00.000	2018-12-31 00:00:00.000	120	2018-02-23 00:00:00.000	2019-01-08 00:00:00.000	127
Portal matriculados	Proyecto de Mejoras	2018-08-08 00:00:00.000	2018-10-31 00:00:00.000	270	2018-08-18 00:00:00.000	2019-01-17 00:00:00.000	133
AMS FX Manager - Jun2017-Ma	Proyecto de Mejoras	2017-06-01 00:00:00.000	2018-05-31 00:00:00.000	240	2017-09-08 00:00:00.000	2018-06-08 00:00:00.000	133.5
Gaucion - Fase II	Proyecto de Mejoras	2020-06-12 00:00:00.000	2020-09-30 00:00:00.000	384	2020-07-16 00:00:00.000	2021-03-19 00:00:00.000	146
Spider FC Electrónica El Salva	Proyecto de Mejoras	2023-09-01 00:00:00.000	2023-10-31 00:00:00.000	160	2023-09-18 00:00:00.000	2023-10-27 00:00:00.000	151
Fx Manager - Im plementacion	Proyecto Original	2017-03-01 00:00:00.000	2017-09-29 00:00:00.000	200	2017-03-02 00:00:00.000	2017-07-05 00:00:00.000	160
Cambio ONLine	Proyecto de Mejoras	2019-06-15 00:00:00.000	2020-01-02 00:00:00.000	800	2019-06-10 00:00:00.000	2020-06-08 00:00:00.000	161
Iniciativas Sharepoint	Proyecto Original	2023-11-01 00:00:00.000	2025-01-17 00:00:00.000	500	2023-11-03 00:00:00.000	2024-06-28 00:00:00.000	180
U-Sign - Mobility Brasil - Sincro	Proyecto de Mejoras	2021-05-01 00:00:00.000	2021-07-30 00:00:00.000	264	2021-06-02 00:00:00.000	2021-09-09 00:00:00.000	183
VitaAire Mobile Uruguay	Proyecto Original	2022-04-01 00:00:00.000	2022-11-30 00:00:00.000	288	2022-04-18 00:00:00.000	2022-12-14 00:00:00.000	191.5
Scanning Tool - FECD	Proyecto de Mejoras	2017-12-18 00:00:00.000	2018-08-30 00:00:00.000	304	2018-01-02 00:00:00.000	2019-01-14 00:00:00.000	192
FX Manager - Implementación	Proyecto Original	2017-06-15 00:00:00.000	2017-08-15 00:00:00.000	350	2017-06-12 00:00:00.000	2018-01-04 00:00:00.000	216
Transporte de Carga - Adicion	Proyecto de Mejoras	2021-08-15 00:00:00.000	2021-12-31 00:00:00.000	200	2021-08-23 00:00:00.000	2021-11-04 00:00:00.000	220
START.NET (Fase 4)	Proyecto de Mejoras	2021-09-01 00:00:00.000	2021-12-31 00:00:00.000	360	2021-08-10 00:00:00.000	2021-12-13 00:00:00.000	238
Plataforma ON LINE	Proyecto de Mejoras	2020-08-15 00:00:00.000	2021-12-31 00:00:00.000	320	2020-08-25 00:00:00.000	2020-11-06 00:00:00.000	245
CFDI de Pago MAM	Proyecto de Mejoras	2018-02-01 00:00:00.000	2018-09-28 00:00:00.000	300	2018-02-20 00:00:00.000	2018-06-27 00:00:00.000	249.5
Implementación CESP en Aka	Proyecto Original	2023-05-01 00:00:00.000	2023-07-28 00:00:00.000	152	2023-05-15 00:00:00.000	2023-12-29 00:00:00.000	263
Mejoras en Sitio RRHH	Proyecto de Mejoras	2018-03-01 00:00:00.000	2018-05-30 00:00:00.000	240	2018-03-12 00:00:00.000	2018-05-21 00:00:00.000	266.5
START.NET - Fase II	Proyecto de Mejoras	2019-09-06 00:00:00.000	2020-12-31 00:00:00.000	3193	2019-09-11 00:00:00.000	2020-04-06 00:00:00.000	267.5
D. Financiera May-Sep 2022 (F	Proyecto Original	2022-04-01 00:00:00.000	2022-09-30 00:00:00.000	1608	2022-04-08 00:00:00.000	2022-07-11 00:00:00.000	308
Mejoras Hytech DMS	Proyecto de Mejoras	2021-12-09 00:00:00.000	2022-10-28 00:00:00.000	382	2021-11-18 00:00:00.000	2022-08-12 00:00:00.000	310.5
FX-Implementacion	Proyecto Original	2019-07-01 00:00:00.000	2020-07-02 00:00:00.000	2000	2019-08-02 00:00:00.000	2020-04-28 00:00:00.000	348.5
Proyecto Home Intranet	Proyecto de Mejoras	2017-08-01 00:00:00.000	2017-09-08 00:00:00.000	142	2017-07-05 00:00:00.000	2017-10-19 00:00:00.000	364
Nuevo engine Einstein	Proyecto de Mejoras	2019-06-01 00:00:00.000	2019-11-29 00:00:00.000	400	2018-08-23 00:00:00.000	2019-05-08 00:00:00.000	366.5
VitaAire (Chile)	Proyecto Original	2019-02-01 00:00:00.000	2019-06-28 00:00:00.000	396	2018-10-22 00:00:00.000	2020-01-21 00:00:00.000	374.5
SECOEXPO - COM "A" 3493	Proyecto de Mejoras	2017-11-01 00:00:00.000	2018-01-31 00:00:00.000	760	2017-07-10 00:00:00.000	2018-02-15 00:00:00.000	378
Informe de control de gestión	Proyecto Original	2019-04-01 00:00:00.000	2019-06-28 00:00:00.000	408	2019-03-29 00:00:00.000	2019-07-05 00:00:00.000	409.5
Plan Tool - SF Cost	Proyecto de Mejoras	2015-11-20 00:00:00.000	2017-10-31 00:00:00.000	432	2015-12-18 00:00:00.000	2017-09-28 00:00:00.000	417.5
Start.Net - Cta. Cte. de interes	Proyecto de Mejoras	2024-07-01 00:00:00.000	2024-09-30 00:00:00.000	840	2024-07-01 00:00:00.000	2024-10-23 00:00:00.000	427.5
Informe de control de gestión	Proyecto de Mejoras	2019-07-01 00:00:00.000	2019-09-30 00:00:00.000	408	2019-07-10 00:00:00.000	2019-09-30 00:00:00.000	430
PCR Intranet	Proyecto Original	2020-09-01 00:00:00.000	2021-02-26 00:00:00.000	1632	2020-08-28 00:00:00.000	2021-05-28 00:00:00.000	463
Timesheet Etapa 2	Proyecto de Mejoras	2018-10-01 00:00:00.000	2018-10-31 00:00:00.000	160	2018-12-26 00:00:00.000	2019-10-03 00:00:00.000	490.5
Digitalización Financiera (Mar	Proyecto Original	2021-03-01 00:00:00.000	2021-08-31 00:00:00.000	1367	2020-10-28 00:00:00.000	2021-08-31 00:00:00.000	495
Sistema de calidad	Proyecto de Mejoras	2017-10-02 00:00:00.000	2018-09-28 00:00:00.000	1076	2016-08-11 00:00:00.000	2018-08-27 00:00:00.000	524
Sistema de control presupues	Proyecto Original	2015-10-05 00:00:00.000	2017-10-30 00:00:00.000	1300	2015-12-22 00:00:00.000	2018-01-31 00:00:00.000	555.5
Sigco EM Fase 2.5	Proyecto de Mejoras	2017-07-01 00:00:00.000	2018-12-28 00:00:00.000	1000	2017-06-26 00:00:00.000	2018-11-02 00:00:00.000	567
Timesheet Tool	Proyecto Original	2017-11-01 00:00:00.000	2018-08-29 00:00:00.000	440	2017-10-03 00:00:00.000	2018-09-13 00:00:00.000	599.5
Start.Net - Cta. Cte. de interes	Proyecto de Mejoras	2022-07-01 00:00:00.000	2023-12-29 00:00:00.000	798	2022-06-23 00:00:00.000	2024-02-16 00:00:00.000	608.25
Gea BI	Proyecto Original	2017-11-01 00:00:00.000	2018-08-29 00:00:00.000	548	2017-09-22 00:00:00.000	2018-08-03 00:00:00.000	616
Unificación factura	Proyecto de Mejoras	2021-01-11 00:00:00.000	2022-08-31 00:00:00.000	981	2021-01-26 00:00:00.000	2022-09-02 00:00:00.000	663
Preven Fase III	Proyecto de Mejoras	2022-02-01 00:00:00.000	2022-08-30 00:00:00.000	860	2022-01-05 00:00:00.000	2022-08-26 00:00:00.000	668.5
Sistema de Gestión de Bienes	Proyecto Original	2019-03-15 00:00:00.000	2019-03-15 00:00:00.000	512	2019-03-11 00:00:00.000	2019-06-12 00:00:00.000	676
Timesheet - ResourceManaje	Proyecto Original	2020-02-01 00:00:00.000	2020-09-30 00:00:00.000	1000	2020-02-04 00:00:00.000	2020-06-01 00:00:00.000	677
APP Transporte	Proyecto Original	2017-05-29 00:00:00.000	2018-08-30 00:00:00.000	632	2017-05-29 00:00:00.000	2018-04-19 00:00:00.000	705
Preven (Fase 2)	Proyecto de Mejoras	2021-09-01 00:00:00.000	2022-03-15 00:00:00.000	1500	2021-01-25 00:00:00.000	2022-01-14 00:00:00.000	709
Start.Net - Cta. Cte. de interes	Proyecto de Mejoras	2024-01-01 00:00:00.000	2024-06-28 00:00:00.000	1234	2024-02-15 00:00:00.000	2024-06-28 00:00:00.000	748.25
Preven fase IV	Proyecto de Mejoras	2022-09-15 00:00:00.000	2023-02-15 00:00:00.000	645	2022-09-07 00:00:00.000	2023-03-30 00:00:00.000	764
BAS - EEUU - Ago2018-Jun202	Proyecto de Mejoras	2018-08-01 00:00:00.000	2020-01-02 00:00:00.000	2328	2018-08-07 00:00:00.000	2020-05-21 00:00:00.000	845.5
SMART.NET	Proyecto Original	2021-02-01 00:00:00.000	2021-12-31 00:00:00.000	1904	2019-09-09 00:00:00.000	2022-01-19 00:00:00.000	861.5
SMART.NET Fase II	Proyecto Original	2022-02-01 00:00:00.000	2022-09-30 00:00:00.000	3000	2022-01-03 00:00:00.000	2022-09-16 00:00:00.000	993.5
BAS - Field Service (Migración	Proyecto de Mejoras	2021-06-11 00:00:00.000	2021-10-29 00:00:00.000	752	2021-06-11 00:00:00.000	2021-11-25 00:00:00.000	1094.5
Registro Firmantes - HeathCa	Proyecto de Mejoras	2016-09-15 00:00:00.000	2017-10-31 00:00:00.000	216	2016-10-03 00:00:00.000	2017-11-23 00:00:00.000	1109.5
AMS SMG Oct - Dic	Proyecto de Mejoras	2017-10-01 00:00:00.000	2017-12-29 00:00:00.000	1700	2017-10-02 00:00:00.000	2017-12-29 00:00:00.000	1117
Planning - Reingeniería	Proyecto de Mejoras	2019-03-01 00:00:00.000	2019-09-30 00:00:00.000	1376	2019-02-26 00:00:00.000	2020-01-29 00:00:00.000	1124
START.NET (Fase 3)	Proyecto de Mejoras	2021-04-01 00:00:00.000	2021-07-30 00:00:00.000	915	2021-01-07 00:00:00.000	2021-09-10 00:00:00.000	1129.5
Sistema de Notificaciones	Proyecto de Mejoras	2017-08-01 00:00:00.000	2017-12-29 00:00:00.000	500	2017-08-01 00:00:00.000	2018-04-05 00:00:00.000	1171.5
START.NET (Fase 2)	Proyecto de Mejoras	2020-08-01 00:00:00.000	2021-01-15 00:00:00.000	863	2020-08-03 00:00:00.000	2020-12-23 00:00:00.000	1453
Dig. Financiera (Sep 2021-Ma	Proyecto Original	2021-09-01 00:00:00.000	2022-03-31 00:00:00.000	3136	2021-09-01 00:00:00.000	2022-03-31 00:00:00.000	1687
AML Anti Lavado de Dinero	Proyecto de Mejoras	2017-05-15 00:00:00.000	2017-09-29 00:00:00.000	1128	2017-03-01 00:00:00.000	2017-12-14 00:00:00.000	1717
Reingeniería Spider MX	Proyecto de Mejoras	2020-05-01 00:00:00.000	2020-09-30 00:00:00.000	2000	2020-04-07 00:00:00.000	2020-11-05 00:00:00.000	2081.5
Desarrollos Sharepoint 2017	Proyecto Original	2017-01-01 00:00:00.000	2017-12-29 00:00:00.000	1816	2016-11-04 00:00:00.000	2017-12-29 00:00:00.000	2258
BAS - EEUU	Proyecto de Mejoras	2017-09-18 00:00:00.000	2018-09-28 00:00:00.000	1580	2017-09-04 00:00:00.000	2018-09-27 00:00:00.000	2356.5
Desarrollo en SAP BTP 2'S 20	Proyecto Original	2024-07-01 00:00:00.000	2025-01-05 00:00:00.000	2448	2024-06-28 00:00:00.000	2025-01-06 00:00:00.000	2535.5
Transporte de Carga	Proyecto Original	2020-07-01 00:00:00.000	2020-11-30 00:00:00.000	660	2020-09-23 00:00:00.000	2021-11-25 00:00:00.000	2572.5
Desarrollo en SAP BTP 1'S 20	Proyecto Original	2024-01-22 00:00:00.000	2024-07-15 00:00:00.000	2448	2023-10-25 00:00:00.000	2024-07-01 00:00:00.000	2997.25
SMART.NET Fase III	Proyecto de Mejoras	2022-08-01 00:00:00.000	2023-03-03 00:00:00.000	3100	2022-02-15 00:00:00.000	2023-04-21 00:00:00.000	3130
Reingeniería Registro Firmant	Proyecto Original	2019-01-01 00:00:00.000	2019-07-31 00:00:00.000	1216	2018-10-16 00:00:00.000	2020-06-03 00:00:00.000	3180.5
Circulación de Especificación	Proyecto Original	2021-12-01 00:00:00.000	2023-03-30 00:00:00.000	3300	2021-12-06 00:00:00.000	2023-06-05 00:00:00.000	4045
AdInsurance - Reportes Norm	Proyecto Original	2022-01-01 00:00:00.000	2022-12-30 00:00:00.000	4000	2021-12-07 00:00:00.000	2022-12-29 00:00:00.000	4947.5
Portal de productores & B2B	Proyecto de Mejoras	2020-10-01 00:00:00.000	2022-03-31 00:00:00.000	7000	2020-10-08 00:00:00.000	2021-12-28 00:00:00.000	8693