Asesorías y Tutorías para la Investigación Científica en la Educación Puig-Salabarría S.C.
José María Pino Suárez 400-2 esq a Lerdo de Tejada. Toluca, Estado de México. 7223898475
RFC: ATI120618V12

Revista Dilemas Contemporáneos: Educación, Política y Valores.
http://www.dilemascontemporaneoseducacionpoliticayvalores.com/
Año: XIII Número: 3 Artículo no.:59 Período: 1 de mayo del 2026 al 31 de agosto del 2026

TÍTULO: Inferencia estadística integrada para la estimación del esfuerzo de software basada en datos históricos incompletos.

AUTORES:

1. Máster. Marco Antonio Guzmán López.
2. Dr. René Santaolaya Salgado.
3. Dr. Vitervo López Caballero.
4. Dra. Blanca Dina Valenzuela Robles.

RESUMEN: La estimación inexacta del esfuerzo en el desarrollo de software genera incumplimientos funcionales y desviaciones significativas en costo, tiempo y recursos previstos. Para atender esta problemática, el presente estudio propone un esquema novedoso de imputación de datos denominado *Inferencia Estadística Integrada (ISI)*, el cual combina la imputación por media o moda con un método propuesto basado en información estadística de proyectos de software (ISPI). El esquema ISI permitió imputar el 29% de los datos faltantes en una muestra aleatoria del repositorio ISBSG. Adicionalmente, se desarrolló un modelo de regresión para la predicción del esfuerzo, alcanzando un valor MMRE del 14,05%, considerado aceptable en la literatura.

PALABRAS CLAVES: estimación por analogía, selección de características, técnicas de imputación, datos faltantes, estadísticas de éxito de proyectos.

TITLE: Integrated statistical inference for software effort estimation based on incomplete historical data.

AUTHORS:

1. Master. Marco Antonio Guzmán López.
2. PhD. René Santaolaya Salgado.
3. PhD. Vitervo López Caballero.
4. PhD. Blanca Diva Valenzuela Robles.

ABSTRACT: Inaccurate estimation of effort in software development leads to functional non-compliance and significant deviations in planned cost, time, and resources. To address this problem, this study proposes a novel data imputation scheme called Integrated Statistical Inference (ISI), which combines imputation by mean or mode with a proposed method based on statistical information from software projects (ISPI). The ISI scheme allowed for the imputation of 29% of the missing data in a random sample from the ISBSG repository. Additionally, a regression model was developed to predict effort, achieving an MMRE value of 14.05%, considered acceptable in the literature.

KEY WORDS: estimation by analogy, feature selection, imputation techniques, missing data, project success statistics.

INTRODUCCIÓN.

La estimación del esfuerzo en el desarrollo de software (Software Development Effort Estimation, SDEE) constituye una actividad crítica que debe realizarse de manera temprana, precisa y confiable, ya que orienta la planificación, asignación de recursos y toma de decisiones a lo largo del ciclo de vida del proyecto; no obstante, durante la última década se han documentado desviaciones significativas en las estimaciones de esfuerzo, con sobrestimaciones que oscilan entre el 41% y el 258%, así como rebasamientos de la inversión prevista que van del 97% al 151%, afectando de forma directa el desempeño, la viabilidad y los resultados finales de los proyectos de software (Iok Kuan, 2017). Estas deficiencias han motivado el desarrollo de múltiples modelos de estimación orientados a mejorar la oportunidad y precisión de los cálculos.

Entre los enfoques tradicionales destacan modelos como COCOMO (Boehm) y aquellos basados en métricas de producto, tales como líneas de código o puntos de función; sin embargo, estos modelos presentan limitaciones relevantes: suelen ser poco interpretables para los diseñadores, requieren información disponible únicamente en etapas avanzadas del desarrollo, o dependen de la finalización de fases previas del ciclo de vida del software, lo que restringe su aplicabilidad en escenarios tempranos de planificación (Shah et al., 2019).

En contraste, en años recientes han surgido métodos de estimación más simples, precisos y ampliamente aceptados, particularmente aquellos fundamentados en técnicas de aprendizaje automático (Machine Learning, ML), como se reporta en diversos estudios (Abnane & Idri, 2018), (Abnane & Idri, 2016), (Shah et al., 2019), (Ezghari & Zahi, 2018), (Resmi & Vijayalakshmi, 2019). Dentro de este grupo, la Estimación por Analogía (Estimation by Analogy, EBA) ha recibido especial atención debido a su simplicidad, facilidad de interpretación y similitud con el razonamiento humano en la resolución de problemas (Wu et al., 2018). Este enfoque se apoya en información histórica de proyectos con esfuerzo previamente conocido (Huang et al., 2017), bajo el supuesto de que proyectos con características similares tienden a requerir esfuerzos comparables.

Una de las principales fortalezas de la EBA radica en su aplicabilidad durante las etapas iniciales del desarrollo del proyecto, cuando la información disponible es limitada (Abnane & Idri, 2016), (Hosni et al., 2017); no obstante, su principal limitación se relaciona con la gestión de datos faltantes (Missing Data, MD), una problemática frecuente en bases de datos históricas de proyectos de software (Amazal et al., 2019), (Idri et al., 2016).

La presencia de datos incompletos puede degradar significativamente la precisión de las estimaciones y conducir a sobrestimaciones o subestimaciones de recursos críticos (Abnane & Idri, 2016). En este sentido, la SDEE demanda bases de datos completas y de alta calidad, lo cual hace indispensable la

aplicación de técnicas de preprocesamiento de datos (Data Preprocessing, DPP), tales como reducción de dimensionalidad, proyección de atributos y tratamiento de valores faltantes.

En respuesta a esta problemática, el presente trabajo propone un esquema de imputación denominado Inferencia Estadística Integrada (ISI), concebido como una estrategia para identificar valores de reemplazo confiables en presencia de datos faltantes. El esquema ISI integra la imputación por media o moda (Imputation by Mean or Mode, IMEO) con un método propuesto denominado Imputación por Información Estadística de Proyectos (ISPI), el cual explota estadísticas derivadas de proyectos reales de software (McConnell, 2006). A diferencia de enfoques convencionales, este esquema incorpora información estadística histórica directamente en el proceso de imputación, con el objetivo de validar un método simple, consistente y útil para completar bases de datos de proyectos de software.

Las técnicas empleadas en el esquema ISI presentan una menor complejidad computacional en comparación con métodos de imputación más sofisticados, como la imputación basada en el K-nearest neighbors (KNNI) o en máquinas de soporte vectorial para regresión (SVR), reportados por Abnane & Idri (2018). En particular, el método ISPI permite imputar valores realistas fundamentados en datos históricos de proyectos similares, y ha demostrado generar estimaciones más precisas que la imputación por mediana del vecino más cercano propuesta por Shah et al. (2019); asimismo, los resultados obtenidos muestran una precisión comparable o superior a la alcanzada por métodos como KNNI, previamente reportados en la literatura (Idri et al., 2016).

Finalmente, esta investigación evidencia que la imputación de datos faltantes, la adecuada selección de características y la integración de información estadística de proyectos constituyen elementos clave para mejorar la precisión en la estimación del esfuerzo de desarrollo de software, en concordancia con lo señalado por Huang et al. (2017).

El modelo propuesto establece una relación entre variables de desarrollo y esfuerzo para predecir el esfuerzo en una muestra aleatoria del repositorio ISBSG, evaluando su desempeño mediante las métricas

MMRE, PRED() y MAE, de manera más sencilla y eficiente en comparación con estudios previos (Abnane & Idri, 2016).

DESARROLLO.

Trabajos relacionados.

La investigación sobre datos faltantes (Shah et al., 2019), (Abnane & Idri, 2018), (Abnane & Idri, 2016) reconoce tres técnicas de datos faltantes:

- *Ignorar.* Mediante esta técnica se borran los casos con datos faltantes. Aunque la técnica es muy popular debido a su sencillez, llega a presentar sesgo cuando se eliminan cantidades considerables de datos y no utiliza el dataset (Shah et al., 2019); por esta razón, se recomienda cuando hay un bajo nivel de datos faltantes (Azzeh & Elsheikh, 2017).
- *Tolerar.* Esta técnica tiene como estrategia tolerar los datos faltantes en el dataset y llevar a cabo un análisis completo y directo del mismo. Un enfoque empleado por la estrategia es asignar un valor especial *NULL* como valor de remplazo del dato faltante (Shah et al., 2019), (Abnane & Idri, 2018), (Resmi & Vijayalakshmi, 2019).
- *Imputar.* La técnica de imputación de datos faltantes rellena los huecos de información en los datos hasta completar todo el dataset para así permitir su análisis. Existen varias estrategias para implementar esta técnica, entre ellas la imputación por media, por mediana, por moda (Wu et al., 2018), (Abnane & Idri, 2018) y del vecino (KNNI) (Resmi & Vijayalakshmi, 2019).

Nuestro esquema ISI se compone de dos métodos de imputación de datos faltantes:

1) La popular técnica de imputación por media/moda IMEO.

2) Un método propio, la imputación por información estadística de proyectos ISPI.

Los dos métodos MD se emplean por los autores relacionados mostrados en la Tabla I.

Abnane & Idri (2018) demuestran que las técnicas de imputación basadas en *k-nearest neighbors* (KNNI) y en regresión mediante *Support Vector Regression* (SVR) alcanzan una mayor precisión en la estimación EBA en comparación con las técnicas tradicionales de tolerancia o eliminación de datos.

Tabla 1. Trabajos relacionados.

Tópico	**Técnica Propuesta**	**(Abnane & Idri, 2018)**	**(Shah et al., 2019)**	**(Idri et al., 2016)**	**(Huang et al., 2017)**	**(Abnane & Idri, 2016)**
Método de estimación de esfuerzo.	EBA	EBA	EBA	EBA, FA	EBA	FA
Mecanismos de datos faltantes.	MCAR	MCAR, MAR, NIM	MCAR, MAR, NIM	MCAR, MAR, NIM	N.A.	MCAR, MAR, NIM
Técnicas de datos faltantes.	ISI	Tolerance, Elimination, Knni, Svr.	Nc, Knni, Minn.	Tolerance, Elimination, Knni.	Knni, Mei.	Tolerance, Elimination, Knni.

Por su parte, Shah et al. (2019) demuestra que su método propuesto MINN maneja valores más realistas que los valores asignados mediante los métodos de imputación de limpieza numérica NC y del vecino KNNI.

Idri et al., (2016) encuentran que la Analogía Difusa produce estimados más precisos de esfuerzo que la Analogía Clásica, y que la imputación KNNI estima con mayor precisión con respecto a la tolerancia y la eliminación.

Huang et al. (2017) señalan que las técnicas DPP de tres etapas (imputación de datos faltantes, normalización de datos y selección de características) pueden ser clave para una estimación EBA más precisa.

Por último, Abnane & Idri (2016) utilizan más de una métrica para evaluar la precisión de predicción por la Analogía Difusa, y combinan la exactitud estandarizada (SA) con la exactitud de predicción PRED.

Aunque en las últimas dos décadas se han desarrollado varios modelos de estimación de esfuerzo, estos no han demostrado ser accesibles, confiables ni exactos. El modelo COCOMO es difícil de entender para los diseñadores; además, los valores de sus impulsores de costos corresponden al periodo comprendido entre los años 1960 y 1970, lo cual hace que el modelo sea incierto e inexacto para el complejo ambiente actual de desarrollo de software (Iok Kuan, 2017).

Por su parte, el método de Análisis de Puntos de Función presenta dificultades similares para los diseñadores del proyecto de software; por ejemplo, el conteo de puntos de función realizado individualmente por los contadores puede ser percibido en forma diferente por cada contador. Incluso si el conteo es efectuado por una casa de software, pueden existir diferencias y propiciar errores (Iok Kuan, 2017).

Preparación de los datos.

El pre-procesamiento *DPP* asegura la completez y calidad de los datos. Las técnicas *DPP* incluyen la reducción y la proyección de datos, y el tratamiento de datos faltantes (MDT). La reducción selecciona solo algunas características relevantes para la estimación y elimina el resto. La proyección de datos transforma a una nueva escala o dimensión la apariencia de los datos. Las técnicas *MDT* incluyen la eliminación de instancias con valores faltantes y/o su reemplazo por estimación de sus valores (Huang et al., 2017).

Nuestro esquema “Inferencia Estadística Integrada (*ISI*)” utiliza la reducción, así como las técnicas *MDT.* La Figura 1, ilustra nuestra preparación inicial de datos.

Nuestro proceso *DPP* se aplicó a una muestra aleatoria de 400 proyectos de software obtenida del reconocido repositorio *ISBSG* (Huang et al., 2017). Los datos faltantes existentes en la muestra aleatoria fueron remplazados por nuestro esquema *ISI* para permitir la experimentación.

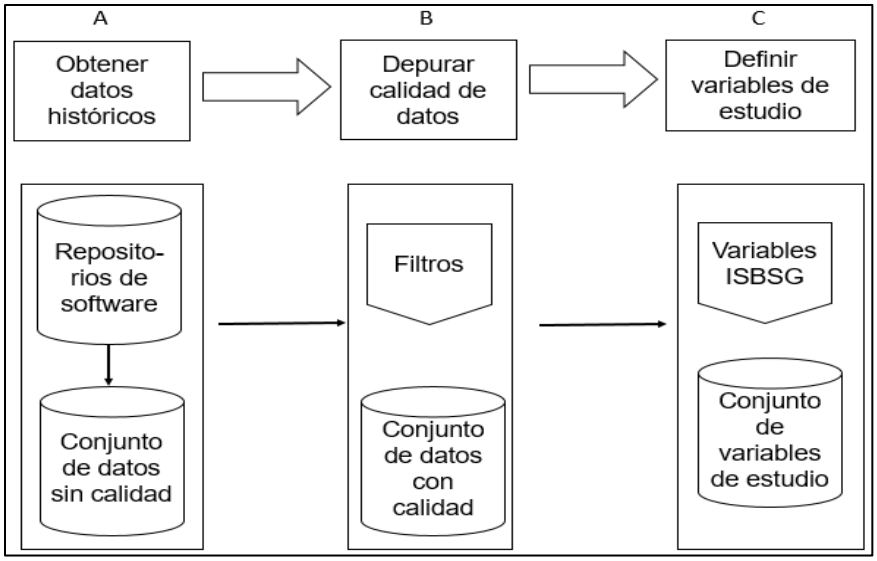

Figura 1. Pasos para la preparación de los datos.

a) Obtener datos históricos.

La predicción del esfuerzo de un nuevo proyecto requiere de datos históricos completos sobre el desarrollo. Se selecciona el repositorio de software ISBSG debido a su cantidad y diversidad de proyectos, la extensa gama de atributos de desarrollo, y la aportación de información de varios países, por lo que representa un panorama global del desarrollo de proyectos.

El repositorio de software ISBSG maneja datos de la historia de proyectos mediante aportaciones que recibe de varias asociaciones de software provenientes de 26 países. Su conjunto de proyectos es heterogéneo, variado en industrias y no contiene proyectos fallidos o abandonados; sin embargo, mantiene un par de carencias: una cantidad considerable de datos faltantes aunada a una baja calidad de datos.

Un proceso de filtrado de datos, sugerido por ISBSG a la comunidad de investigación, aumenta la calidad de los datos de experimentación; a su vez, el problema de los datos faltantes MD, debe ser resuelto por los propios investigadores mediante alguna técnica MDT, en nuestro estudio, el esquema ISI.

La base de datos del repositorio ISBSG corresponde al reléase de agosto, 2020 R1, el cual cuenta con 9,592 proyectos de software terminados con un formato de registro de 251 atributos de desarrollo (escenario de aplicación, contexto de desarrollo, dimensionamiento del proyecto, entre otros).

Dado que ISBSG es un repositorio grande y heterogéneo, se justifica realizar un tratamiento de sus datos como paso previo para cualquier análisis.

b) Depurar la calidad de los datos.

Son 4 los filtros sugeridos por ISBSG para aumentar la calidad de sus datos. La aplicación de los filtros a los 9,592 registros de la base de datos ISBSG genera un conjunto de 4,595 registros con calidad de datos para poder estimar el esfuerzo (Huang et al., 2017). La cantidad de registros omitidos por cada filtro se muestra en la Tabla 2.

Tabla 2. Cantidad de registros omitidos por filtro.

Filtro	**Nombre**	**Sugerencia ISBSG**	**Registros omitidos**	**Registros resultantes**
1	Calificación de la calidad de datos	"A" "B"	973	8619
2	Calificación de puntos de función no ajustados	"A" "B"	1351	7268
3	Enfoque de conteo	"IFPUG 4+"	1846	5422
4	Esfuerzo normalizado de trabajo de nivel 1	1	827	4595

c) Definir variables de estudio.

Los proyectos ISBSG están compuestos por 251 características. La literatura de investigación ha intentado identificar las características ISBSG más apropiadas para la predicción (Huang et al., 2017) del esfuerzo de software. Sus resultados guían nuestra selección de variables. Un estudio emprendido por González-Ladrón-de-Guevara et al. (2016) logró determinar las 20 variables ISBSG más utilizadas por estudios científicos para estimar el esfuerzo.

La investigación realizada por Wu et al. (2018) también se basa en el estudio realizado por González-Ladrón-de-Guevara et al. (2016) para seleccionar un subconjunto de variables de estudio.

De igual forma, nuestra investigación selecciona un subconjunto de 15 características ISBSG como variables de estudio para la experimentación, 9 son cualitativas y 6 cuantitativas. La Tabla 3 muestra nombre, tipo, porcentaje de valores faltantes, y método de imputación aplicado para la variable.

Tabla 3. Características seleccionadas.

No.	Variables	Identificador	Tipo de dato	Valores faltantes (%)	Técnica de imputación
1	Sector industria	IS	Categórica	19.75	Moda
2	Tipo de organización	OT	Categórica	19.75	Moda
3	Tipo de Aplicación	AT	Categórica	18.25	Moda
4	Tipo de desarrollo	DT	Categórica	0.00	N.A.
5	Plataforma de desarrollo	DP	Categórica	45.50	Moda
6	Tipo de lenguaje	LT	Categórica	26.50	Moda
7	Lenguaje primario de programación	PPL	Categórica	28.25	Moda
8	Tamaño funcional	FSZ	Continua	0.00	N.A.
9	Esfuerzo normalizado de trabajo, nivel 1	NWEL 1	Continua	0.00	N.A.
10	Resumen de esfuerzo de trabajo	SWE	Continua	0.00	N.A.
11	Tiempo del proyecto	PET	Continua	0.00	N.A.
12	Tamaño máximo del equipo	MTS	Continua	67.25	ISPI
13	Sistema de base de datos primario	1DBS	Categórica	65.50	Moda
14	Metodología utilizada	UM	Categórica	45.75	Moda
15	Líneas de código	LOC	Continua	97.50	ISPI

Hay 5 variables con datos completos, i.e. porcentaje de datos faltantes igual a cero. Entre las 10 variables restantes se encuentran 8 variables categóricas y 2 continuas. La imputación por moda se aplica a las 8 variables categóricas. En las variables continuas no es conveniente el uso de la imputación por media debido a su alto porcentaje de datos faltantes. En su lugar se aplica nuestro método ISPI.

Metodología de solución.

La predicción del esfuerzo de trabajo requiere contar con datos completos, de calidad y accesibles para la experimentación por la comunidad científica.

El repositorio de software *ISBSG* es utilizado con bastante frecuencia en trabajos de investigación sobre desarrollo de software. Nuestro estudio utiliza un conjunto de 4,595 registros *ISBSG* con calidad de datos, pero con datos faltantes en varios de sus atributos; de este conjunto, se seleccionó una muestra de 400 registros para entrenamiento del modelo de predicción propuesto.

Nuestro método ISI selecciona aleatoriamente 400 proyectos, formando rangos por número de proyecto, sin considerar las demás variables de estudio. La intención de esta forma de selección es probar que la simplicidad del método propuesto puede producir resultados validos de predicción de esfuerzo.

Se utilizó el número de proyecto para seleccionar 201 proyectos en el rango 10003 al 10998, y 199 proyectos en el rango 20000 al 21098 para completar los 400 proyectos del conjunto de entrenamiento. Bajo este criterio, no se clasificó los proyectos para su selección. Por simplicidad de nuestro método propuesto, se asumió que cada proyecto tenía una misma participación en el experimento.

Los datos MD existentes en la muestra de 400 registros requieren tratamiento para poder utilizarse por la experimentación. Entre las diversas técnicas de datos MD se encuentra la imputación de valores. Nuestro estudio propone el esquema ISI mostrado en la Figura 2 para la imputación de datos MD.

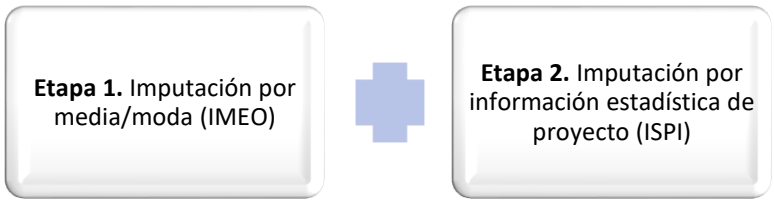

Figura 2. Inferencia estadística integrada (ISI).

El porcentaje de valores faltantes, columna 5 en Tabla 3, se refiere a la proporción de registros con valores faltantes en el conjunto de 400 proyectos de la muestra de entrenamiento; p. ej. la variable 12 MTS, en 269 de los 400 proyectos (67.25%), no contiene valor alguno. El método de imputación indicado en la última columna se refiere a nuestro método ISPI; por otra parte, las variables categóricas con datos

faltantes utilizan el conocido método de imputación por Moda, p. ej. la variable 5 DP con 45.50% de datos faltantes.

La completez de datos permite aplicar análisis estadísticos, de correlación y de regresión, entre otros. Mediante un análisis de correlación se puede determinar las variables de más alta relación y relevancia para la experimentación. Un análisis de regresión identifica el grado de contribución de cada variable al esfuerzo de desarrollo.

Ya cubiertos los datos MD de la muestra aleatoria de 400 registros, se tiene un conjunto de datos completos y con la calidad suficiente para llevar a cabo la experimentación de modelos de predicción de esfuerzo. Primero, se requiere convertir a valores numéricos los valores de las variables categóricas. Nuestro presente estudio toma la frecuencia de repetición de cada valor en la variable categórica para asignarla como valor numérico.

Luego, con ayuda del lenguaje de programación R, se normalizan los valores de la muestra aleatoria de 400 registros para que sean uniformes en escala. El lenguaje R declara el dataset ISBSG para la muestra y aplica la función normalizar usando los valores mínimo y máximo; además, el lenguaje R determina la correlación entre variables dependientes e independientes. A su vez, los índices de correlación identifican las variables que más influyen sobre la variable de estimación de esfuerzo.

En el repositorio ISBSG hay tres características que se refieren al esfuerzo. El Resumen del Esfuerzo de Trabajo (SWE) es la principal característica. Se mide en horas de personal. Es el esfuerzo total del proyecto reportado por la organización contribuyente; sin embargo, SWE no cubre todas las fases del ciclo de vida del proyecto.

La mayoría de los artículos acerca de estimación (89%) informan que el esfuerzo es la variable dependiente en los modelos de estimación del esfuerzo, aunque algunos utilizan la productividad (4,7%) o la tasa de entrega del proyecto (3,7%) (González-Ladrón-de-Guevara et al., 2016); sin embargo, el conjunto de datos de ISBSG tiene tres variables de esfuerzo:

1) El Resumen del Esfuerzo de Trabajo SWE.

2) El Esfuerzo Normalizado NE.

3) El Esfuerzo Normalizado de Nivel 1 NWEL1 (variable 9, Tabla 3).

Aunque es un hecho que algunos artículos indican la variable de esfuerzo, que utilizan como variable dependiente, muchos otros no lo hacen originando que el 19% de los artículos no sean repetibles, porque su variable dependiente es ambigua o desconocida.

El Resumen del Esfuerzo de Trabajo SWE, variable 10 en Tabla 3, es la variable de esfuerzo más usada. Simplemente, es el valor reportado del esfuerzo medido en horas staff. Los dos tipos de esfuerzo restantes sí distinguen las fases del ciclo de vida que se incluyen en SWE.

El Esfuerzo Normalizado NE (12,1 %) es la estimación de ISBSG del esfuerzo total cuando es necesario agregar las fases faltantes del ciclo de vida; sin embargo, a pesar de utilizar NE, aún puede haber cierta inconsistencia entre proyectos, ya que estos informan el esfuerzo que involucra a diferentes participantes identificados por la variable Nivel de Recurso RL, la cual considera 4 niveles.

Ascendiendo en cada nivel, se agrega el esfuerzo de más participantes en el esfuerzo. El nivel 1 indica que el esfuerzo se informa únicamente para el equipo de desarrollo; en consecuencia, el esfuerzo de trabajo normalizado nivel 1 (variable 9) es el esfuerzo normalizado únicamente para el equipo de desarrollo; por lo tanto, González-Ladrón-de-Guevara et al. (2016) recomiendan el uso de NWEL1 como variable dependiente, para asegurar la máxima coherencia. Nuestro estudio consideró también a la variable NWEL1 como la variable dependiente para estimar el esfuerzo. Métricas utilizadas y comprobadas como confiables por trabajos de investigación sirven para evaluar la precisión de nuestro modelo de predicción de esfuerzo.

Inferencia Estadística Integrada (ISI).

Las compañías de software no recopilan datos completos del desarrollo, por considerar que esa tarea podría incrementar el costo total del proyecto, o que otras tareas tienen mayor prioridad. Esta situación se

presenta durante el ciclo de desarrollo y origina el problema de datos faltantes MD en bases de datos históricas como en los repositorios ISBSG y PROMISE (Abnane & Idri, 2018), (Huang et al., 2017), (Sigweni, 2016), y además, repercute en estudios imprecisos de estimación de esfuerzo (Chinthanet et al., 2016).

En la muestra de 400 proyectos, se cuenta con 15 características por proyecto, resultando así en un total de 6,000 valores posibles. Al analizar el grupo de 400 proyectos, se obtiene un total de 1,736 valores faltantes que equivalen al 28.93% del total de valores.

Si se toma el criterio de eliminarlos, se tendría una muestra menor de proyectos; esto es 2/3 de la muestra original, la cual sería insuficiente para realizar las pruebas, o bien, provocaría resultados no confiables. Se hace necesario entonces utilizar una técnica de imputación de valores faltantes.

Nuestro esquema de imputación, Inferencia Estadística Integrada (ISI), puede ayudar a solucionar los datos faltantes. Los dos componentes del esquema son A) Imputación por Media/Moda IMEO y B) Imputación por Información Estadística de Proyectos ISPI.

a) Imputación por Media/Moda (IMEO).

Esta técnica de imputación utiliza los valores observados para remplazar cada valor faltante por la MEDIA de los valores numéricos o por la MODA de los valores categóricos. La técnica conserva la información de los datos y puede disminuir la varianza de las variables (Huang et al., 2017).

Como un ejemplo de imputación por moda, la variable 7 PPL (ver Tabla 3), contiene 28.25% de datos faltantes (113/400). La moda entre los 287 valores existentes obtiene una frecuencia de repetición para cada lenguaje. El lenguaje JAVA tiene como MODA la frecuencia más alta de 51 y es el valor de remplazo de los valores faltantes. La moda del lenguaje se asigna como el valor de la variable a fin de convertirla en numérica y así pueda intervenir en los análisis de correlación y regresión.

El alto porcentaje de datos faltantes en las variables numéricas ocasiona un sesgo muy alto al calcular la media de los valores existentes, lo cual no hace recomendable el uso de la imputación por media. En su lugar, nuestro presente estudio aplica el método propio de imputación ISPI.

b) Imputación por Información Estadística (ISPI).

Este método se utiliza para resolver el 97.50% de datos faltantes en la variable 15 (ver Tabla 3) "Líneas de Código Fuente", LOC, y el 67.25% de valores faltantes en la variable 12 (ver Tabla 3) "Tamaño Máximo del Equipo", MTS.

La Tabla 4 utiliza los resultados de la investigación sobre proyectos de software terminados mostrada por la versión 5, de la Tabla QSM de lenguajes y puntos de función PF, la cual contiene información estadística de 2192 proyectos y 126 lenguajes, donde cada lenguaje tiene asociado un factor de soporte por la relación Líneas de Código / Puntos de Función (LOC/PF).

Tabla 4. Estadísticas LOC por punto de función.

No. de proyecto	**Tamaño funcional no ajustado PF**	**Lenguaje de programación**	**QSM/SLIM promedio PromFS**	**Líneas de código fuente LOC**
10003	67	Java	53	3551
10007	51	Java	53	2703
10011	443	Access	37	16391
10012	76	COBOL	61	4636

Con ayuda de la Tabla QSM se relacionan puntos de función PF del Tamaño Funcional del proyecto (variable 8 en Tabla 3) con lenguajes de programación (variable 7 en Tabla 3) para asociarlos con factores de soporte correspondientes proporcionados por la Tabla de Lenguajes y Puntos de Función, en columna 4.

El promedio del factor soporte PromFS correspondiente al lenguaje, columna 4, permite obtener el valor faltante de LOC, columna 5 por la expresión:

$$LOC = PromFS * PF \qquad (1)$$

y tomarlo como valor de remplazo en los valores faltantes de la variable *LOC*.

La información estadística recopilada por McConnell (2006) sobre 500 proyectos de software terminados con un rango de entre 35.000 y 95.000 líneas de código permite imputar valores *MD* del tamaño máximo del equipo, variable 12 *MTS*.

Los proyectos se agruparon en cinco categorías de acuerdo con el tamaño de su equipo de desarrollo. La categoría 1 abarca entre 1,5 a 3 personas. La categoría 2 incluye de 3 a 5 personas. La categoría 3 considera de 5 a 7 personas. La categoría 4 oscila entre 9 a 11 personas, y la categoría 5 varía entre 15 y 20 personas.

Los resultados de McConnell (2006) reforzaron la creencia general respecto a que al crecer el tamaño del equipo, el esfuerzo de trabajo aumenta, y sin embargo, el tiempo calendario del proyecto aumenta. Equipos más allá del rango de entre 5 y 7 personas, categoría 3, aumentaban en esfuerzo pero también en el calendario.

Cuando se incrementa el rango 1, 5 – 3 al de 3 – 5, y de 3-5 a 5-7, el tiempo calendario se acorta y el esfuerzo se incrementa, pero cuando se pasa de 5-7 a 9-11 tanto el esfuerzo como el calendario se incrementan. En el rango 15 – 20 la situación es más grave.

Los equipos de más de 9 personas requieren de mayor coordinación y de más rutas de comunicación, lo cual crea más errores; por ello, es común aceptar de 5 a 9 personas como el tamaño máximo del equipo. Esto es, un tamaño mínimo de 5 personas y otro máximo de 9.

Nuestro estudio adopta el valor de 9 personas para remplazar los datos faltantes en la variable 12 tamaño máximo del equipo.

La aplicación de nuestro esquema de imputación *ISI* mediante la combinación de los dos métodos de imputación *IMEO* e *ISPI*, obtiene un conjunto de datos completos. Con ayuda del lenguaje R se determinan los coeficientes de correlación entre las variables dependiente e independientes y se calcula el valor del

estimado NWEL1 para compararlo con su valor actual en la muestra ISBSG y comprobar la exactitud de nuestro método ISI.

Adicionalmente, se emplea el método de imputación de datos faltantes por el KNN vecino más cercano disponible en el lenguaje R. El método KNN se aplica a la muestra de 400 proyectos con datos faltantes y se procede igualmente a su análisis para determinar el valor de su NWEL1 estimado y compararlo con su valor actual para conocer la exactitud de predicción. La comparación de la exactitud entre ambos métodos ISI y KNN identifica al método más exacto.

Resultados.

Durante el proceso de estimación, tomar en cuenta las variables con mayor correlación influyó para mejorar el MMRE y el PRED (0.25) del modelo de estimación. El conjunto de datos de estudio contiene 400 proyectos, cada uno con 15 características, lo que equivale a un total de 6,000 valores. Examinando los valores faltantes, se encuentra que 1,736 (28.93%) valores son cero ("0") o blancos (““). Solo ocho atributos tienen menos del 66% de valores faltantes. Hay cinco atributos sin valores faltantes.

La estadística de valores faltantes en el conjunto de 400 proyectos y 15 atributos es de los 15 atributos, 10 de ellos tuvieron valores faltantes (66.67%).

Para cubrir los valores faltantes, se aplicó el esquema de imputación propuesto ISI que combina el popular método de imputación por MODA en 8 atributos (80%), y nuestro método propuesto “Imputación por Información Estadística de Proyecto (ISPI)” en 2 atributos (20%). Esto señala que el 29% de los atributos resolvieron sus valores faltantes por nuestro esquema propuesto ISI.

CONCLUSIONES.

Nuestro estudio propone un método de imputación de valores faltantes basado en la combinación de la imputación por media/moda y un método novedoso: imputación por estadísticas de proyectos. El esquema

propuesto ISI demuestra una precisión y aplicabilidad dentro del umbral comúnmente considerado como aceptable por la literatura. El método considera los aspectos:

1) Manejo de atributos no cuantitativos.

2) Manejo de valores faltantes.

3) Define un modelo de correlación de variables.

4) Define un modelo de regresión lineal para determinar el valor real final del estimado de esfuerzo.

En la Tabla I se compara el esquema ISI contra otros 5 métodos desarrollados por la investigación relacionada. Tanto el estudio presente como otros métodos similares comparados no tienen valores óptimos del número N de proyectos históricos ni del número K de proyectos similares que den la precisión óptima esperada; en especial, para conjuntos de datos más grandes con cientos o incluso miles de objetos, donde el tamaño influye en la exactitud de predicción; en consecuencia, la analogía más cercana, como se supone en los métodos existentes, puede no producir la precisión esperada, particularmente en el caso de grandes conjuntos de datos.

En general, sería recomendable que los repositorios abiertos de software (ISBSG y PROMISE) llevarán a cabo un proceso de recopilación de datos más completo y preciso entre sus socios a fin de asegurar un nivel de calidad de datos óptimo y un mínimo de datos faltantes. Esto permitiría una mayor precisión de los estudios de estimación del esfuerzo y redundaría en un beneficio colectivo para la comunidad de software. Nuestro estudio recomienda como mínimo contar con todos los valores completos de las 15 variables de estudio mostradas en la Tabla 3.

Los resultados obtenidos por nuestro estudio no se pueden generalizar dado que sólo se exploró un conjunto de datos pequeño. Es deseable una exploración de más conjuntos de datos y de mayor tamaño a fin de obtener mayor conocimiento de otras técnicas de estimación y depurar a la vez el funcionamiento de la técnica analizada.

El valor de este presente proyecto consiste en augurar éxitos futuros en proyectos afines, en el uso de varios juegos de datos, en aplicación de clasificaciones por atributos de desarrollo, p.ej.., las primeras 4 variables de estudio, en muestras de datos de mayor tamaño e incluso en contrastar resultados entre varios repositorios de software y el empleo de métodos de imputación de mayor complejidad, p.ej. de aprendizaje automático.

REFERENCIAS BIBLIOGRÁFICAS.

1. Abnane, I., & Idri, A. (2016). Evaluating Fuzzy Analogy on incomplete software projects data. 1-8. https://doi.org/10.1109/SSCI.2016.7849922.
2. Abnane, I., & Idri, A. (2018). Improved Analogy-Based Effort Estimation with Incomplete Mixed Data. 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), 1015-1024. https://ieeexplore.ieee.org/document/8511226
3. Amazal, F. A., Idri, A., & Abran, A. (2019). Analysis of cluster center initialization of 2FA-kprototypes analogy-based software effort estimation. Journal of Software: Evolution and Process, 31(12), e2180. https://doi.org/10.1002/smr.2180.
4. Azzeh, M., & Elsheikh, Y. (2017). Learning best K analogies from data distribution for case-based software effort estimation (arXiv:1703.04567). arXiv. https://doi.org/10.48550/arXiv.1703.04567.
5. Chinthanet, B., Phannachitta, P., Kamei, Y., Leelaprute, P., Rungsawang, A., Ubayashi, N., & Matsumoto, K. (2016). A review and comparison of methods for determining the best analogies in analogy-based software effort estimation. Proceedings of the 31st Annual ACM Symposium on Applied Computing, 1554-1557. https://doi.org/10.1145/2851613.2851974.
6. Ezghari, S., & Zahi, A. (2018). Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method. Applied Soft Computing, 67, 540-557. https://doi.org/10.1016/j.asoc.2018.03.022.

7. González-Ladrón-de-Guevara, F., Fernández-Diego, M., & Lokan, C. (2016). The usage of ISBSG data fields in software effort estimation: A systematic mapping study. Journal of Systems and Software, 113, 188-215. https://doi.org/10.1016/j.jss.2015.11.040.
8. Hosni, M., Idri, A., & Abran, A. (2017). Investigating heterogeneous ensembles with filter feature selection for software effort estimation. Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, 207-220. https://doi.org/10.1145/3143434.3143456.
9. Huang, J., Li, Y. F., Keung, J. W., Yu, Y. T., & Chan, W. K. (2017). An empirical analysis of three-stage data-preprocessing for analogy-based software effort estimation on the ISBSG data. Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017, 442-449. https://doi.org/10.1109/QRS.2017.54.
10. Idri, A., Abnane, I., & Abran, A. (2016). Missing data techniques in analogy-based software development effort estimation. Journal of Systems and Software, 117, 595-611. https://doi.org/10.1016/j.jss.2016.04.058.
11. Iok Kuan, S. W. (2017). Factors on Software Effort Estimation. International Journal of Software Engineering & Applications, 8(1), 23-32. https://doi.org/10.5121/ijsea.2017.8103.
12. McConnell, S. (2006). Software estimation: Demystifying the black art. Microsoft Press.
13. Resmi, V., & Vijayalakshmi, S. (2019). Analogy-Based Approaches to Improve Software Project Effort Estimation Accuracy. Journal of Intelligent Systems, 29(1), 1468-1479. https://doi.org/10.1515/jisys-2019-0023.
14. Shah, M. A., Jawawi, D. N. A., Isa, M. A., Wakil, K., & Younas, M. (2019). MINN : A Missing Data Imputation Technique for Analogy-based Effort Estimation. International Journal of Advanced Computer Science and Applications, 10(2), 12. https://doi.org/10.14569/IJACSA.2019.0100230.

15. Sigweni, B. B. (2016). An investigation of feature weighting algorithms and validation techniques using blind analysis for analogy-based estimation [Thesis, Brunel University London]. http://bura.brunel.ac.uk/handle/2438/12797.

16. Wu, D., Li, J., & Bao, C. (2018). Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation. Soft Computing, 22(16), 5299-5310. https://doi.org/10.1007/s00500-017-2985-9.

DATOS DE LOS AUTORES.

1. **Marco Antonio Guzmán López.** Maestro en Sistemas Computacionales Administrativa por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM). Actualmente, es estudiante de Doctorado en Ciencias de la Computación, en la línea de investigación de Ingeniería de Software, en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Correo electrónico: marco.guzman17ce@cenidet.edu.mx, ORCID: https://orcid.org/0000-0002-1500-5635. Autor de correspondencia.

2. **René Santaolaya Salgado.** Doctor en Ciencias de la Computación por el Instituto Politécnico Nacional, en el Centro de Investigación en Computación. Actualmente se desempeña como profesor investigador de tiempo completo en el Centro Nacional de Investigación y Desarrollo Tecnológico Correo electrónico: rene.ss@cenidet.tecnm.mx, ORCID: https://orcid.org/0000-0003-3408-5818.

3. **Vitervo López Caballero.** Doctor en Ciencias Computaciones con la especialidad de Ingeniería de Software por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Actualmente se desempeña como profesor investigador de tiempo completo en el CENIDET. Correo electrónico: vitervo.lc@cenidet.tecnm.mx, ORCID: https://orcid.org/0000-0002-1942-9558.

4. **Blanca Dina Valenzuela Robles.** Doctora en Ciencias Computacionales con la especialidad de Ingeniería de Software por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).

Actualmente se desempeña como profesora investigadora de tiempo completo en el CENIDET. Correo electrónico: blanca.vr@cenidet.tecnm.mx, ORCID: https://orcid.org/0000-0002-7303-3052.

RECIBIDO: 26 de enero del 2026. **APROBADO:** 21 de febrero del 2026.