MONTEVIDEO, URUGUAY / DICIEMBRE, 2014

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA

Calidad de Datos en Experimentos de Ingeniería de Software

Tesis de Maestría: Carolina Valverde

TUTORES DE TESIS: ADRIANA MAROTTA / DIEGO VALLESPIR
PROGRAMA: MAESTRÍA EN INGENIERÍA DE SOFTWARE

“In God we trust. All others must bring data.”

W. Edwards Deming

Resumen

En la Ingeniería de Software se realizan experimentos con el objetivo de demostrar si ciertas teorías o creencias son reales. Durante la ejecución de estos experimentos, los sujetos humanos registran ciertos datos. Estos datos son almacenados en repositorios, y serán utilizados por los investigadores y por la comunidad empírica en general para obtener resultados y conclusiones. Si los datos a partir de los cuales se generan los resultados del experimento no tienen los niveles adecuados de calidad y no reflejan la realidad, entonces las decisiones que tome la comunidad en Ingeniería de Software a partir de los mismos pueden ser equivocadas.

La Calidad de Datos es un área de investigación cuyo foco está en definir diferentes aspectos que permitan obtener el máximo valor de los datos en su contexto de uso, así como en proponer técnicas, métodos y metodologías para medir y tratar problemas a nivel de la calidad de los datos. Los problemas de calidad de datos se generan cuando, al utilizar los datos, éstos no tienen los valores adecuados de calidad. En particular, la aplicación de técnicas y actividades de la calidad sobre datos de experimentos en ingeniería de software no ha sido aún muy desarrollado.

En este trabajo se propone un modelo de calidad de datos y una metodología de trabajo para evaluar y mejorar la calidad de los datos resultante de la ejecución de experimentos en Ingeniería de Software. El modelo de calidad de datos que están basados en los conceptos más referenciados y utilizados por los principales autores del área de Calidad de Datos, y que ya han sido aplicados de forma exitosa en otros dominios. La metodología es propuesta para que pueda ser instanciada y utilizada por los investigadores en Ingeniería de Software.

El modelo y metodología propuestos son aplicados sobre los datos de cuatro experimentos en Ingeniería de Software para validarlos en casos reales. Gracias a las aplicaciones, fue posible encontrar que estos datos contienen problemas de calidad que deben ser atendidos antes de obtener los análisis estadísticos de los experimentos.

La aplicación de la metodología propuesta sobre los datos de los cuatro experimentos muestra que tanto la metodología como el modelo de calidad definidos contribuyen en la evaluación y mejora de la calidad de los datos analizados. Es posible su aplicación sobre casos concretos de este dominio particular y la misma resulta en un beneficio importante para el experimento, el experimentador, y para la comunidad en ingeniería de software empírica en general. Los resultados obtenidos permiten también posicionar nuestra propuesta de forma tal que pueda ser aplicada sobre los datos de otros experimentos en ingeniería de software que involucran sujetos humanos.

Observamos también que el esfuerzo invertido durante la aplicación no es excesivo, principalmente en relación al importante beneficio obtenido. Los experimentadores también demostraron su amplia aceptación respecto al enfoque propuesto, considerando que es aplicable y ampliamente beneficioso en este dominio.

Por todo esto, el trabajo realizado constituye un aporte importante tanto para la comunidad en Ingeniería de Software Empírica como para la de Calidad de Datos. En particular, muestra un avance en el tema de calidad de datos para experimentos en ingeniería de software. Contribuye a la evaluación y mejora de la calidad de los datos utilizados por los experimentadores, así como de los mecanismos de recolección y almacenamiento de los datos. El modelo y metodología propuestos sistematizan este trabajo de evaluación y mejora, facilitando su incorporación en futuras experiencias. Esto impacta en la calidad, confianza y validez de los resultados obtenidos en los experimentos, y que serán utilizados por los profesionales e investigadores para avanzar en sus trabajos, investigaciones y en la toma de decisiones clave.

Índice General

Índice de tablas

Índice de ilustraciones

Capítulo 1: Introducción

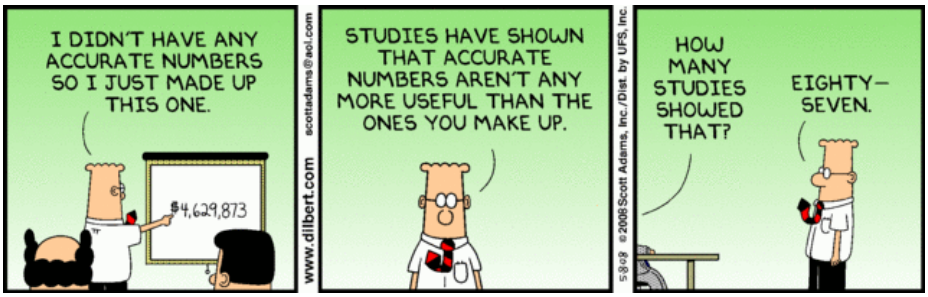

Ilustración 1: Motivación

Durante la experimentación en Ingeniería de Software se generan una gran cantidad de datos. Sin embargo, la problemática sobre si el nivel de calidad que contienes esos datos es adecuada, no ha sido atendida con la importancia que se merece en este contexto. Sorprendentemente, en esta comunidad no se presentan ni aplican protocolos o métodos sistemáticos ni explícitos para evaluar y mejorar la calidad de los datos que se utilizan [1]. Sólo una minoría de los trabajos en ingeniería de software empírica discuten la problemática de la calidad de datos, e incluso menos trabajos aún toman acciones para tratar los problemas de calidad sobre los datos. Estos problemas pueden deberse, por ejemplo, a valores faltantes o incorrectos, inconsistencias en los datos, duplicaciones o incluso representaciones y formatos en la estructura de datos no adecuadas a la realidad. Si no conocemos el nivel de calidad de los datos en los cuales se basan los resultados, su validez resultará cuestionable pudiendo llegar a ser incluso ignorados por el resto de la comunidad [2].

En la investigación empírica se utilizan datos recolectados para realizar análisis y obtener conclusiones acerca de algún hecho de la realidad. Estos resultados son utilizados por la comunidad científica para adaptar y mejorar sus propios trabajos e investigaciones. La calidad de los datos en los cuales se basan los análisis empíricos afecta directamente a los resultados y conclusiones de la investigación [2], [3].

La Ingeniería de Software (IS) es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, mantenimiento y operación del software [4]. Los principios y prácticas de esta disciplina son esenciales para el desarrollo de sistemas grandes, complejos y confiables [5].

La Ingeniería de Software Empírica busca, mediante el empirismo, mostrar hechos de la realidad. Hay 3 tipos principales de técnicas o estrategias para la investigación empírica: las encuestas, los estudios de casos y los experimentos [6]. En todos ellos se recolectan datos, a partir de los cuales se realizan análisis y se obtienen conclusiones. Resulta fundamental conocer el nivel de calidad de los datos que están siendo utilizados. Si los datos en los cuales se basan las investigaciones en la ingeniería de software no son de buena calidad, entonces los resultados que se obtengan a partir de los mismos pueden no ser correctos.

En particular, la experimentación se refiere a la correspondencia de las suposiciones, asunciones, especulaciones y creencias acerca de algo con hechos de la realidad. En la Ingeniería de Software, la experimentación permite confirmar teorías, conocer los factores que hacen a un software bueno o mejor que otro, así como las técnicas, métodos y herramientas más apropiadas para desarrollar software bajo determinadas situaciones. Cierto conocimiento será considerado científicamente válido, siempre y cuando su validez haya sido demostrada, y además exista una comprobación de este conocimiento respecto a la realidad.

Un experimento controlado es una investigación empírica en la cual se manipula uno de los factores o variables del estudio, manteniendo el resto de las variables constantes. Se aplican diferentes tratamientos a los diferentes sujetos, y se mide el efecto en las variables resultantes. Los experimentos son ejecutados en un ambiente de laboratorio, lo cual provee una mayor nivel de control [6].

Cada experimento controlado en Ingeniería de Software es ejecutado en las siguientes fases: Definición, Planificación, Operación, Análisis e Interpretación, Presentación y Empaquetado [7]. Los datos que son reco-

lectados, generados y almacenados durante la ejecución de un experimento son de suma importancia y valor, ya que constituyen la base para realizar los análisis estadísticos y obtener los resultados del experimento.

Para estudiar la calidad que tiene un conjunto de datos es necesario identificar una serie de criterios conocidos como dimensiones que son usualmente definidas como propiedades o características de calidad. Para tratar la calidad de un sistema de información o conjunto de datos particular, es necesario definir un modelo de calidad que represente el conjunto de dimensiones y sus relaciones [8].

En este trabajo se presenta un Modelo de Calidad de Datos que desarrollamos específicamente para el dominio de Experimentos en Ingeniería de Software que involucran sujetos humanos. Este modelo tiene como objetivo evaluar y mejorar la calidad de los datos generados durante la ejecución de este tipo de experimentos. El modelo define métricas de calidad para ser aplicadas en casos específicos, y que están basadas en los conceptos propuestos y acordados por los principales autores del área de Calidad de Datos [9]–[13]. También desarrollamos una metodología repetible que define los pasos y guías para aplicar el modelo de calidad sobre los datos de un experimento particular en ingeniería de software. El modelo y la metodología de calidad propuestos fueron aplicados sobre los datos de 4 experimentos en ingeniería de software.

En el resto del capítulo definimos la motivación y el contexto de la investigación en las secciones 1.1 y 1.2. Luego se presentan en la sección 1.3 los objetivos del trabajo. Los resultados obtenidos a partir del trabajo realizado se describen en la sección 1.4. Finalmente se presentan las publicaciones realizadas y el contenido del documento en las secciones 1.5 y 1.6.

1.1 Contexto de investigación

La presente investigación fue realizada en el marco de la Maestría en Ingeniería de Software de la Facultad de Ingeniería (FIng), Universidad de la República, Uruguay. Integra al área de Ingeniería de Software (más específicamente Ingeniería de Software Empírica) del Grupo de Ingeniería de Software (GrIS), con el área de Sistemas de Información (más específicamente de Calidad de Datos) del grupo Concepción de Sistema de Información (CSI), ambos pertenecientes al Instituto de Computación (InCo) de la FIng.

Durante el transcurso de la investigación se realizaron dos pasantías en el exterior. La primera fue de 1 mes de duración (Julio 2013) junto a la Dra. Natalia Juristo, en el Grupo de Investigación en Ingeniería de Software Empírica (GrISE) de la Universidad Politécnica de Madrid (UPV). La segunda pasantía se realizó en el Centro de Investigación en Métodos de Producción de Software (PROS) de la Universitat Politècnica de València (UPV), en conjunto con el Dr. Óscar Pastor. Esta última pasantía tuvo una duración de 10 meses (Noviembre 2013-Agosto2014).

Ambas pasantías permitieron y facilitaron la ejecución de una parte importante del trabajo que se realizó junto con los diferentes experimentadores de UPV y UPM.

1.2 Motivación

La Ingeniería de Software ha ganado importancia en los últimos tiempos, debido principalmente al incremento sustancial en la complejidad de los productos de software. La información acerca de la evolución e historia de proyectos de software es recolectada y almacenada en repositorios de datos, de forma de ser utilizada por la comunidad empírica para hacer análisis y obtener conclusiones. Algunos ejemplos de esto son reportes de defectos y cambios en el software, o datos sobre la gestión de configuración [3].

Durante los últimos años, la Ingeniería de Software Empírica ha tomado más relevancia. Su insumo más importante son los datos, al ser utilizados para realizar predicciones, nuevos descubrimientos, tomar decisiones acerca de nuevas técnicas o estrategias, o incluso determinar la efectividad de las técnicas y herramienta utilizadas, o impacto de uso [14]. Los resultados de los experimentos son generalmente considerados verdaderos y confiables, y son utilizados para tomar decisiones y obtener conclusiones. La comunidad en Ingeniería de Software utilizará estos resultados para contribuir, mejorar o ajustar sus propios procesos e investigaciones. Esto impactará el trabajo realizado no solo por los investigadores, sino por los profesionales en Ingeniería de Software que basan sus decisiones en estos resultados [3]. Sin embargo, la calidad de los datos en los cuales se basan estos resultados no es comúnmente cuestionada o analizada. A pesar de que la problemática de calidad de datos no es nueva en el área de Ingeniería de Software Empírica, no ha recibido la atención e importancia que merece [15].

Entonces, si los datos en los cuales se basan los resultados de los experimentos contienen errores, las conclusiones obtenidas pueden no ser confiables. Si la calidad de los datos utilizados es desconocida o contienen errores, entonces los supuestos planteados por la comunidad empírica serán inciertos, cuestionados o ignorados por el resto de la comunidad [2], [3].

El área de investigación de Calidad de Datos se ha enfocado en definir diferentes técnicas, métodos y metodologías para medir y tratar los problemas de calidad de datos [9], [10], [13]. La importancia de considerar y atender la problemática de calidad de datos ha sido reconocida por los productores y consumidores de datos en varias áreas de investigación, como Sistemas de Información y Minería de Datos [11], [16], debido a su impacto en los resultados obtenidos. Los problemas de calidad de datos pueden afectar a diferentes contextos, como compañías, gobiernos, universidades, clientes, organizaciones [16], [17].

Para la academia la buena calidad de los datos resultará en investigaciones de mejor y mayor calidad así como en la obtención de resultados más confiables [18]. Afortunadamente, en los últimos tiempos se ha comenzado a tomar conciencia sobre la importancia de la calidad de los datos en diferentes contextos [19]. En particular, desde la perspectiva de la Ingeniería de Software, la calidad de datos también se considera una temática de interés y que merece atención [18], [20]–[22].

La importancia de la calidad de los datos utilizados por los estudios empíricos en la Ingeniería de Software también ha comenzado a ser reconocida y evaluada en los últimos años [2], [3], [14], [15], [23]–[26], principalmente por su impacto en las decisiones tomadas. Algunos trabajos enfatizan explícitamente la importancia de la calidad de los datos para repositorios de datos [2], [15], [27].

Bachman analizó las características de calidad de repositorios de datos abiertos y cerrados de proyectos de software, y encontró que todos ellos contenían problemas de calidad [3], [23], [28]. Estos problemas podrían tener un impacto importante en los resultados de las investigaciones empíricas en ingeniería de software. Bachman define un *framework* y métricas de calidad de datos para evaluar y analizar la calidad de los datos de proyectos de software; sin embargo, no establece si pueden ser aplicadas a datos experimentales.

Según *Liebchen*, parece haber un incremento en el tiempo en la cantidad de trabajos que consideran la problemática de Calidad de Datos, sugiriendo que la comunidad le está presentado más atención a este temática [2]. Sin embargo, no encontramos en la literatura ningún estudio que analice específicamente la calidad de los datos cuya fuente sea un experimento controlado en Ingeniería de Software. Tampoco encontramos trabajos que propongan protocolos, metodologías o guías que puedan ser aplicadas de forma sistemática, disciplinada y estructurada sobre este dominio.

1.3 Objetivos

El objetivo general de nuestro trabajo es proponer un Modelo de Calidad de Datos que permita evaluar y mejorar la calidad de los datos recolectados durante la ejecución de experimentos controlados en Ingeniería de Software que involucran sujetos humanos. Específicamente buscamos desarrollar un enfoque sistemático, disciplinado y estructurado que utilice dicho modelo, y que provea las pautas y guías para que la aplicación del modelo de calidad pueda ser repetible y generalizable sobre los datos de cualquier experimento en Ingeniería de Software. Otro objetivo planteado es mostrar que tanto el modelo como la metodología de trabajo propuesta (que utiliza ese modelo) son instanciables a datos de experimentos particulares.

El modelo y metodología de calidad de datos son propuestos para ser aplicados de forma posterior a la ejecución del experimento (una vez se han recolectado todos los datos), pero antes de que los análisis estadísticos sean llevados a cabo. De esta forma será posible medir y mejorar la calidad de datos antes de que sean utilizados, para así obtener resultados experimentales en base a datos que contienen un nivel de confianza definido y conocido, y que sean realmente representativos del experimento.

Nuestra pregunta general de investigación es la siguiente: *¿Es posible y brinda algún beneficio a los investigadores en ingeniería de software y a la comunidad empírica en general, aplicar un método sistemático, disciplinado y estructurado para analizar la calidad de datos de un experimento controlado en Ingeniería de Software cuyos sujetos son humanos?* A continuación se presentan los objetivos de la tesis, que apuntan a responder la pregunta de investigación planteada.

1. Objetivo uno: Conocer si existen modelos, metodologías o protocolos de calidad de datos ya propuestos y que puedan ser aplicados sobre los datos de experimentos en ingeniería de software. De ser así, conocer de qué forma se realiza y cuál es el marco conceptual en el cual se basan.
2. Objetivo dos: Proponer y definir un modelo de calidad de datos que pueda ser aplicado específicamente en el dominio de experimentos en ingeniería de software, y una metodología de trabajo sistemática, disciplinada y estructurada que utilice dicho modelo para evaluar y mejorar la calidad de sus datos.
3. Objetivo tres: Aplicar el modelo y metodología de calidad propuestos sobre datos de experimentos particulares. A partir de este objetivo se definen tres sub-objetivos: mostrar que es posible su uso en casos (experimentos) reales; evaluar y mejorar la calidad de los datos de los experimentos analizados; conocer la relación costo-beneficio (esfuerzo invertido vs. mejora en la calidad de los datos) de su aplicación.

1.4 Trabajo realizado y resultados obtenidos

Para cumplir con el *primer objetivo: conocer si existen modelos, metodologías o protocolos de calidad de datos ya propuestos y que puedan ser aplicados sobre los datos de experimentos en ingeniería de software*, se realizó una revisión de la literatura. Se tomaron como punto de partida las tres revisiones de la literatura llevadas a cabo en el dominio particular de calidad de datos para ingeniería de software empírica [2], [3], [25]. Los resultados obtenidos muestran que existe interés y preocupación por cómo los investigadores están tratando la problemática de calidad de datos. Todos ellos concluyen que la comunidad en ingeniería de software empírica debería prestar más atención al análisis y mejora de la calidad de los datos recolectados y utilizados en este contexto, ya que los resultados muestran que esta temática ha sido ignorada.

Los resultados de la revisión sistemática de la literatura llevada a cabo por *Liebchen* y *Shepperd* [14], [15] muestran que solamente un 1% de los trabajos analizados consideran explícitamente el "ruido" (*noise*) o la calidad de datos como una problemática, sin necesariamente proponer soluciones. El ruido se define como "información incorrecta, falta de información o información no confiable" [2]. A pesar de que la mayoría de los trabajos reconocen su importancia (138 de 161), muy poco se ha hecho para tratar los problemas de calidad de datos. *Liebchen* sugiere que se debería desarrollar un protocolo de calidad de datos unificado para la comunidad en Ingeniería de Software Empírica, ya que ninguno de los trabajos encontrados aplican o proponen uno.

El *segundo objetivo: proponer y definir un modelo de calidad de datos que pueda ser aplicado en el dominio de experimentos en ingeniería de software, y una metodología de trabajo que utilice dicho modelo para evaluar y mejorar la calidad de sus datos*, apunta justamente a cubrir la carencia planteada en el párrafo anterior. Para esto, desarrollamos un modelo de calidad de datos y una metodología sistemática, disciplinada y estructurada que utiliza dicho modelo, con el objetivo de evaluar y mejorar la calidad de los datos de experimentos en ingeniería de software que involucran humanos como sujetos. Definimos métricas de calidad de datos que están basadas en los conceptos y técnicas propuestas por el área de Calidad de Datos [9]–[13].

Para la construcción y desarrollo de los productos de investigación (modelo de calidad de datos y metodología de aplicación del modelo) utilizamos una metodología de investigación iterativa e incremental, basada en la mejora continua [29]. Luego de cada aplicación particular, el modelo puede ser refinado y ajustado de acuerdo a las lecciones aprendidas y/o nuevas necesidades identificadas.

El modelo de calidad y la metodología de aplicación del modelo, fueron instanciados sobre los datos de 4 experimentos controlados en ingeniería de software. De esta forma, cumplimos con el *tercer objetivo: aplicar el modelo y metodología de calidad propuestos sobre datos de experimentos particulares*. Presentamos como fue llevada a cabo la aplicación del método y del modelo, y los resultados obtenidos en cada caso.

Encontramos que los datos utilizados por estos experimentos presentan problemas de calidad que deben ser tratados antes de realizar los análisis estadísticos. Podría ser necesario tener que tomar acciones correctivas y preventivas con el fin de mejorar la calidad de los datos utilizados, y de esta forma aumentar la confianza en los resultados obtenidos. Mostramos que mediante la aplicación de la metodología y modelo propuesto se encuentran problemas de calidad que podrían ser de otra forma ignorados.

1.5 Publicaciones

En el transcurso de esta tesis se publicó un artículo en el *Workshop Quality of Models and Models of Quality (QMMQ 2014)*, parte de la *33rd International Conference on Conceptual Modeling (ER 2014)*. Esta es una de las conferencias más reconocidas a nivel internacional sobre modelado conceptual. En este artículo se presenta la aplicación del modelo de calidad de datos definido en los dos experimentos de la UPV.

Valverde Carolina, Vallespir Diego, Marotta Adriana, Panach Jose Ignacio: Applying a Data Quality Model to Experiments in Software Engineering. ER Workshop 2014, LNCS 8823, pp.168-177, 2014. Springer International Publishing Switzerland 2014 [30].

También se realizaron otras dos publicaciones en conferencias latinoamericanas. Ambos presentan la aplicación del modelo de calidad de datos definido (en versiones anteriores) y los resultados obtenidos. El primero de ellos es sobre los datos del experimento de UdelaR, mientras que el segundo es sobre los datos recolectados en la ejecución de un proceso de desarrollo de software [31].

Valverde, C., Vallespir, D., Marotta, A.: Análisis de la Calidad de Datos en Experimentos en Ingeniería de Software. En Proceedings CACIC 2012, pp. 794-803, Argentina (2012) [32]

Valverde, C., Grazioli, F., Vallespir, D.: Un Estudio de la Calidad de los Datos Recolectados durante el Uso del Personal Software Process. En Proceedings JIISIC 2012. Lima, Peru (2012) [33]

1.6 Estructura

El informe consta de 8 capítulos además del actual. En el Capítulo 2 *"Calidad de Datos"* y Capítulo 3 *"Experimentación en Ingeniería de Software"*, se presentan los conceptos y características fundamentales sobre el área de Calidad de Datos e Ingeniería de Software Empírica respectivamente. Luego se presentan los principales trabajos que existen en el dominio bajo estudio, en el Capítulo 4 titulado *"Trabajos Relacionados"*. En dicho capítulo se presenta de qué manera se han aplicado diferentes técnicas, herramientas y conceptos de la Calidad de Datos a repositorios que contienen datos de Ingeniería de Software, y más específicamente de Ingeniería de Software Empírica.

En el Capítulo 5 *"Modelo de Calidad de Datos y Metodología de Aplicación para Experimentos en Ingeniería de Software"* se presenta el modelo y métricas de calidad definidas para el contexto bajo estudio, así como la metodología de trabajo seguida para su aplicación sobre los datos de Experimentos en Ingeniería de Software. La metodología de investigación que se siguió para la construcción y el desarrollo del modelo de calidad de datos se presenta luego, en el Error: Reference source not found bajo el título *"Metodología de Investigación"*.

Las aplicaciones del modelo y metodología propuesta sobre datos de experimentos particulares se presentan en el Capítulo 7 *"Aplicación de la Metodología y Modelo de Calidad sobre los Datos de Experimentos de Métodos de Desarrollo"* y Capítulo 8 *"Aplicación de la Metodología y Modelo de Calidad sobre los Datos de Experimentos de Técnicas de Verificación"*. En el primer caso se describe de forma detallada la aplicación de la metodología y del modelo de calidad (por cada fase de la metodología seguida) y los resultados obtenidos, mientras que en el segundo caso se presenta la aplicación solamente de forma resumida.

El análisis y discusión de los resultados obtenidos considerando los 4 casos de aplicación se presenta en el Capítulo 9 de *"Resultados y Discusión"*. Finalmente, el Capítulo 10 *"Conclusiones y Trabajos a Futuro"* presenta las conclusiones acerca del trabajo, los logros obtenidos, las limitaciones identificadas, y las tareas que serían de interés abordar en futuros trabajos.

La tesis contiene también 4 Anexos. En el Anexo A *"Aplicación de la Metodología y Modelo de Calidad sobre los Datos del Experimento de UdelaR" y* Anexo B *"Aplicación de la Metodología y Modelo de Calidad sobre los Datos del Experimento de UPM"*, se presentan ambos casos de aplicación de forma completa. La instanciación detallada del modelo de calidad de datos y los resultados detallados obtenidos a partir de las 4 aplicaciones se incluyen como parte del Anexo C "*Modelo de Calidad de Datos y Aplicaciones*". Finalmente, el Anexo D *"Cuestionario de Satisfacción y Respuestas de los Experimentadores"* muestra el cuestionario de satisfacción diseñado y las respuestas recibidas por parte de los experimentadores.

Capítulo 2: Calidad de Datos

En este capítulo se presenta el contexto general de Calidad de Datos, incluyendo su definición, principales conceptos e importancia. También se presentan diferentes metodologías para la evaluación y mejora de la calidad de datos y modelos de calidad existentes. Por último se trata la temática de la limpieza de datos.

2.1 Introducción

Los datos representan objetos del mundo real. Están presentes en muchas de las actividades que llevamos a cabo en nuestro día a día, siendo generados, registrados, procesados y utilizados en diferentes contextos.

En particular, los datos constituyen un recurso muy valioso así como un activo de importancia estratégica para las organizaciones, llegando en algunos casos a garantizar la sobrevivencia y éxito de las mismas. Los datos son también generados y utilizados por las comunidades para la toma de decisiones y para proveer resultados a la comunidad entera.

El problema de la calidad de datos ha sido objeto de estudio desde varias perspectivas y por diferentes áreas a lo largo de los años, tal es el caso de la Estadística, Gestión o Computación. A medida que su importancia se hace más evidente a los ojos de estas y otras áreas, se incrementan también las investigaciones e intenciones de mejora en este sentido.

El objetivo de este capítulo es introducir los principales conceptos y técnicas del área de Calidad de Datos, ya que los mismo son de gran relevancia para nuestro trabajo. Para esto nos basamos fuertemente en el libro de *Batini y Scannapieco* [9], así como en las principales publicaciones que existen en la literatura de Calidad de Datos [10]–[13], [16].

A través de este análisis bibliográfico podremos:

- Conocer el estado del arte de Calidad de Datos: comprender la teoría y las principales propuestas que existen en esta área de investigación, así como su importancia.
- Proveer las bases para seleccionar el marco conceptual sobre el cual se sustentará nuestro trabajo, y poder justificar los motivos de su elección.
- Este capítulo está estructurado como sigue. Se introducen los principales conceptos de Calidad de datos, luego se describen algunas metodologías y modelos propuestos en esta área, y por último se trata la limpieza de datos, cuyo objetivo final es la mejora de su calidad.

2.2 ¿Qué es la Calidad de Datos?

De forma general, la calidad de los datos se expresa mediante un conjunto de dimensiones que son usualmente definidas como propiedades o características de calidad. Para tratar la calidad de un sistema de información (SI) o conjunto de datos particular, es necesario definir un modelo de calidad (ver sección 2.6) que se basará en cierto conjunto de dimensiones.

La definición de Calidad de Datos utilizada comúnmente es la de "adecuación al uso" (*fitness for use/purpose*) [1], [2], [13], [14], [34]. Es el grado de beneficio o valor percibido por el usuario al utilizar ciertos datos en un determinado contexto [35]. De manera similar, la buena calidad de datos se puede definir como *"fit for their intended purpose in operations, decision making, and planning"* [20], [36].

Esta definición, de naturaleza puramente subjetiva, indica que el "uso" o "propósito" de los datos será adecuado para cada usuario/consumidor dependiendo principalmente del contexto en cual se encuentra inmerso, las características que se consideren relevantes del mismo, así como de sus necesidades, requerimientos u objetivos [37]. En particular, el contexto de los datos (para qué van a ser utilizados, por quiénes, en qué organización, de qué forma) es fundamental para definir si los mismos su nivel de calidad [14]. La perspectiva del consumidor de los datos es central para el análisis de su calidad, y considera el concepto de Calidad de

Datos desde un punto de vista más amplio: no sólo intrínseco al dato [10], [12], sino que también dependiendo de su diseño, contexto y proceso de producción [17].

La calidad de los datos también se puede definir como "la distancia que existe entre los datos que son presentados en un sistema de información y los mismos datos en el mundo real" [13]. Cuáles son los datos del mundo real que son de relevancia para cada usuario/consumidor, dependerá de sus necesidades, expectativas, requerimientos y características, entre otros.

Por otra parte, así como se contempla la percepción subjetiva de los consumidores de los datos, no hay que dejar de lado la medición objetiva, ya que ambas visiones resultan ser complementarias [11]. En este sentido, es posible definir métricas que permitan evaluar de forma objetiva y subjetiva la calidad de los datos.

Volviendo a la naturaleza subjetiva de la definición, la gestión de la calidad de los datos requiere un entendimiento de cuáles son las dimensiones de calidad importantes para el usuario de los mismos [8]. De la misma forma en que el "uso" o "propósito" al que se refiere la definición presentada es subjetivo, también lo son las dimensiones de calidad de datos, ya que no pueden ser evaluadas de manera independiente a quienes hacen uso de los datos y su contexto [2].

2.3 La importancia de la Calidad de Datos

La mala calidad de los datos influye de manera significativa y profunda en la efectividad y eficiencia de las organizaciones así como en todo el negocio, llevando en algunos casos a pérdidas multimillonarias. Cada día se hace más notoria la importancia y necesidad en distintos contextos de un nivel de calidad adecuado para los datos. Por esto, es importante lograr identificar las causas por las cuales ciertos datos son de mala calidad, para eliminar, o en su defecto mejorar, la problemática de raíz [9].

En el contexto de un negocio, la falta de datos críticos o la presencia de valores incorrectos en los mismos, podría afectar negativamente los procesos de negocio clave, causando su detención u otros resultados no deseados. Por otra parte, una mala calidad en los datos podría resultar en toma de decisiones inapropiadas para la estrategia de negocio [35]. Existen reportes sobre el costo en el que incurren las organizaciones por utilizar datos de mala calidad, no solo económico sino también de reputación.

Para las organizaciones contar con datos que gocen de buena calidad en su contexto de aplicación determinado es fundamental para que sus procesos de toma de decisiones resulten certeros, ya que serán la clave para la sobrevivencia de las mismas. Por otra parte, para la academia la buena calidad de los datos resultará en investigaciones de mejor y mayor calidad así como en la obtención de resultados más confiables [18]. En particular, los problemas asociados con los valores de los datos son de especial interés para la ingeniería de software empírica [2].

Afortunadamente, en los últimos tiempos se ha comenzado a tomar conciencia sobre la importancia de la calidad de los datos en diferentes contextos [19]. En particular, desde la perspectiva de la Ingeniería de Software la Calidad de Datos también se considera una temática de interés y que merece atención [18], [20]–[22]. Para que los datos que se utilizan aporten valor (ya sea a nivel de la industria o de la academia), es fundamental que exista confianza en los mismos.

Por otra parte, no debe dejarse de lado el balance entre el costo y el beneficio de invertir en la calidad de datos. El costo de la calidad de datos se define como la suma del costo de la evaluación más de las actividades de mejora [19], también referido como el costo asociado a la mala calidad. Organizaciones encuestadas por Garner reportaron perder un promedio de $8.2 millones por año debido a la mala calidad de datos. El 22% de las compañías encuestadas (de un total de 140) estiman pérdidas anuales por $20 millones, mientras que un 4% llegarían a los $100 millones [20].

En cualquier dominio la fase de recolección de datos es crítica para su calidad. Esto puede darse mediante alguna herramienta, o ser registrados directamente por humanos. Los errores que se pueden introducir en cada caso varían, ya que la naturaleza de cada fuente de datos es diferente. Las herramientas pueden introducir errores sistemáticos (por ejemplo si fue mal construida o calibrada), que típicamente llevan a cadenas de *outliers* asociados. Por otra parte, los humanos pueden introducir errores sistemáticos (por falta de formación) o aleatorios (por descuidos), que típicamente llevan a *outliers* asilados. Algunos ejemplos de errores en los datos introducidos por humanos incluyen: lectura incorrecta de escalas, ingreso incorrecto de datos a partir de un instrumento o herramienta, transposición de dígitos o ingreso de valores en lugares incorrectos. Estos errores son en general difíciles de detectar. Incluso si se verificara, se requeriría del conocimiento de un experto para saber si es razonable o absurdo [37].

2.4 Dimensiones de Calidad de Datos

Como se mencionó en la sección anterior, existen distintos aspectos que componen la calidad de datos. Estos aspectos son conocidos normalmente como dimensiones de calidad.

En los trabajos del área de Calidad de Datos, existen variadas definiciones y clasificaciones sobre las dimensiones de calidad de datos que son planteadas por los diferentes autores. Sin embargo, existen diferencias en la interpretación de las definiciones y en el diseño de métricas para las mismas dada la naturaleza subjetiva y contextual de la calidad de datos [2], [17], [19]. De todas formas, analizando las clasificaciones de dimensiones de calidad más importantes, existe un núcleo de dimensiones que es compartido por la mayoría de los autores presentada en esta sección [9], [13], [19], [38]–[40]. Estas son: correctitud, unicidad, completitud, consistencia y frescura. También presentamos las dimensiones representación e interpretabilidad, ya que luego las utilizamos en nuestro trabajo.

En [40], [41] se presenta una abstracción de la calidad de datos como la que se muestra en la Ilustración 2, donde además de las dimensiones se definen otros conceptos para la clasificación y el manejo de la misma. Estos conceptos son el de factor, métrica y método de medición.

Mientras que una dimensión de calidad captura una faceta (a alto nivel) de la calidad de los datos, un factor de calidad representa un aspecto particular de una dimensión de calidad. Una dimensión puede ser entendida como un agrupamiento de factores que tienen el mismo propósito de calidad.

Las medidas cuantitativas de la calidad de los datos se obtienen mediante las métricas. Una métrica es un instrumento que define la forma de medir un factor de calidad. Un mismo factor de calidad puede medirse con diferentes métricas. A su vez, un método de medición es un proceso que implementa una métrica. Se pueden utilizar distintos métodos de medición para una misma métrica. Generalmente los métodos de medición son los que hay que particularizar para cada contexto.

Se pueden considerar distintos niveles de granularidad para evaluar la calidad de los datos. Por ejemplo, en una base de datos relacional las granularidades posibles serían: celda, tupla, columna, tabla, e incluso la base de datos entera. En una planilla electrónica o cualquier estructura de datos tipo tabla, las granularidades podrían definirse de forma análoga. Sin embargo en otros modelos (orientados a objetos, xml, texto plano, etc.) las granularidades definidas serán diferentes. Por esto se definen funciones de agregación, las cuales calculan un valor de calidad para un conjunto de datos a partir de valores de calidad medidos para cada elemento de ese conjunto, es decir permiten pasar de un nivel de granularidad de datos a otro, obteniendo la calidad resumida para ese nuevo nivel. Por ejemplo, es posible obtener una medida de calidad de una tupla a partir de las medidas de calidad de cada una de sus celdas. El ratio es una de las posibles funciones de agregación que pueden utilizarse,

la cual consiste en identificar la cantidad de valores sin problemas de calidad sobre la cantidad de valores totales. Otros ejemplos de funciones de agregación son los promedios y promedios ponderados.

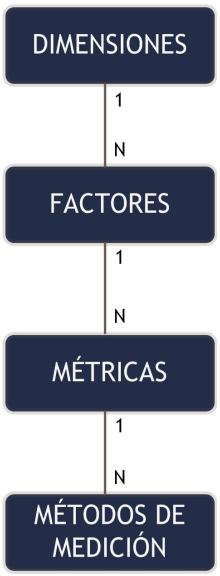

Ilustración 2: Abstracción de los principales conceptos de Calidad de Datos

A continuación se presentan las dimensiones de calidad de datos que son consideradas para el presente trabajo.

2.4.1 Exactitud *(Accuracy)*

La exactitud se puede definir como la cercanía que existe entre un valor v del mundo real, y su representación v'. Se refiere a la correcta y precisa asociación entre los estados del SI y los objetos del mundo real.

Existen tres factores de exactitud: exactitud semántica, exactitud sintáctica y precisión.

La exactitud sintáctica se refiere a la cercanía entre un valor v y los elementos de un dominio D. Esto es, si v corresponde a algún valor válido de D (sin importar si ese valor corresponde a uno del mundo real).

Para poder medir la exactitud sintáctica se puede utilizar la comparación de funciones, métrica que mide la distancia entre un valor v y los valores en el dominio D. Otras alternativas posibles son la utilización de diccionarios que representen fielmente el dominio, o el chequeo de los datos contras reglas sintácticas.

La exactitud semántica se refiere a la cercanía que existe entre un valor v y un valor real v'. Esta dimensión se mide fundamentalmente con valores booleanos (indicando si es un valor correcto o no), para lo cual es necesario conocer cuáles son los valores reales a considerar.

En este caso, interesa medir qué tan bien se encuentran representados los estados del mundo real. En general la exactitud semántica es más compleja de medir que la exactitud sintáctica (ya que se requieren conocer los valores del mundo real). Una forma de chequear la exactitud semántica es comparar diferentes fuentes de datos (referenciales considerados válidos), y encontrar a partir de estas el valor correcto deseado.

Por último, la precisión se refiere al nivel de detalle de los datos.

2.4.2 Unicidad *(Uniqueness)*

La Unicidad se refiere al nivel de duplicación que tienen los datos. La duplicación ocurre cuando un objeto del mundo real se encuentra presente más de una vez (más de un registro representa exactamente el mismo objeto). Existen diferentes situaciones que pueden llevar a la duplicación de datos: cuando la misma entidad se identifica de diferentes formas, cuando ocurren errores en la clave primaria de una entidad, o cuando la misma entidad se repite con diferentes claves.

Distinguimos dos factores de la dimensión Unicidad.

- Duplicación: la misma entidad aparece repetida de manera exacta.
- Contradicción: la misma entidad aparece repetida con contradicciones.

2.4.3 Completitud *(Completeness)*

La completitud se puede definir como la medida en que los datos son de suficiente alcance y profundidad. Se refiere a la capacidad del SI de representar todos los estados significativos de una realidad dada. Existen dos factores de la completitud: cobertura y densidad.

La cobertura se refiere a la porción de datos de la realidad que se encuentran contenidos en el SI. Al igual que para la exactitud semántica, la cobertura involucra una comparación con el mundo real, por lo que un referencial es también requerido. Debido a que suele ser difícil obtenerlo, otra alternativa es estimar el tamaño de tal referencial.

La densidad se refiere a la cantidad de información contenida y faltante acerca de las entidades del SI. En un modelo relacional, la densidad puede caracterizarse por los valores nulos. Un valor nulo puede indicar que dicho valor no existe en el mundo real, que el valor existe pero no se conoce, o que no se sabe si el valor existe o no en el mundo real.

2.4.4 Dimensiones relacionadas con el tiempo

Los cambios y actualizaciones de los datos son un aspecto importante de la calidad de datos a tener en cuenta. Se describen los siguientes dimensiones relacionados con el tiempo.

- Actualidad (*Currency*): trata sobre la actualización de los datos y su vigencia. Puede ser medida de acuerdo a la información de “última actualización".
- Volatilidad (*Volatility*): se refiere a la frecuencia con que los datos cambian en el tiempo. Una medida es la cantidad de tiempo que los datos permanecen siendo válidos.
- Oportunidad (*Timeliness*): especifica que tan actuales/viejos son los datos para la tarea/evento en cuestión. Para medirla es necesario considerar una métrica de actualidad, y verificar que los datos se encuentren dentro del límite establecido por la tarea/evento en cuestión.

2.4.5 Consistencia *(Consistency)*

Esta dimensión hace referencia al cumplimiento de las reglas semánticas que son definidas sobre los datos. La inconsistencia de los datos se hace presente cuando existe más de un estado del SI asociado al mismo objeto de la realidad. Una situación que podría ocasionar inconsistencias en los datos es la incorporación de datos externos o con otros formatos.

Las restricciones de integridad definen propiedades de consistencia que deben ser cumplidas por los datos. Se distinguen tres tipos de restricciones, que corresponden a los factores de calidad:

- Restricciones de dominio: satisfacción de reglas sobre el contenido de los atributos de una relación.
- Restricciones intra-relacionales: satisfacción de reglas sobre uno o varios atributos de una relación.
- Restricciones inter-relacionales: satisfacción de reglas sobre atributos de distintas relaciones.

2.4.6 Interpretabiliad *(Interpretability)*

Además de las dimensiones ya presentadas, que forman parte de la propuesta consensuada, existen otras varias dimensiones que han sido planteadas por diferentes autores.

En particular, la interpretabilidad [9] hace referencia a la documentación y metadata que se encuentran disponibles para poder interpretar correctamente el significado y propiedades de las fuentes de datos. De forma de maximizar la interpretabilidad, se debería disponer de la siguiente información:

- esquema conceptual de los archivos o bases de datos
- restricciones de integridad que existen entre los datos
- conjunto de metadatos para los diferentes dominios de información
- información de la historia, origen y trazabilidad de los datos

La interpretabilidad [12], [13] está relacionada con el formato en que los datos son especificados, incluyendo el lenguaje y unidades, así como la claridad (no ambigüedad) de las definiciones de los datos.

Esta dimensión está estrechamente vinculada con la facilidad de entendimiento, que se define como [39] el grado en que la información es capaz de ser entendida e interpretada, o [12] el alcance en que los datos son claros, sin ambigüedades y fácilmente comprensibles. A partir de estos conceptos se consideran dos factores para la dimensión interpretabilidad: facilidad de entendimiento y metadata.

2.4.7 Representación *(Representation)*

La calidad de datos representacional [12] incluye aspectos relacionados con el formato de los datos, esto es, que la representación de los datos sea concisa y consistente. En esta misma línea se define también la consistencia representacional [13], que es el alcance en que los datos son siempre representados en el mismo formato. A partir de estos conceptos se consideran dos factores para la dimensión representación: estructura de datos y formato de datos.

2.5 Metodologías para la evaluación y mejora de la Calidad de Datos

Las metodologías proveen guías de acción para el tratamiento de la calidad de datos. Se presentan a continuación algunas de las metodologías propuestas.

En [19] se comparan diferentes metodologías para la evaluación y mejora de la calidad de datos para diferentes contextos, pero ninguna de ellas es para datos de ingeniería de software o ingeniería de software empírica. La metodología se define como un conjunto de guías y técnicas que componen un proceso para la evaluación y mejora de la calidad de datos.

Algunas de las fases que tienen en común las diferentes metodologías propuestas para la evaluación y medición de la calidad de datos son: captura de requisitos de datos, captura de requisitos de calidad de datos, análisis de los datos, definición de áreas críticas, modelado de procesos, medición de la calidad, y definición de metadata. Para la mejora de la calidad, se distinguen entre otras:

evaluación de costos, identificación de las causas de errores, selección de estrategias y técnicas, diseño de soluciones de mejora, y monitoreo.

En todas las metodologías, las dimensiones de calidad y métricas para evaluar la misma es una actividad crítica. En general, existen múltiples métricas asociadas a cada dimensión de calidad.

Se destaca el trabajo planteado por [35], ya que tiene algunas similitudes con nuestra propuesta. Proponen y aplican una metodología de trabajo previamente definida, y utilizan como base el mismo marco conceptual [9], [12] que el nuestro. Las diferencias fundamentales están en que se presenta el método y el proceso para definir métricas de calidad, pero sin mostrar resultados (esto es, la propias métricas ni su aplicación). Además, el foco del trabajo está en el negocio y no en la academia.

En [42] se propone la aplicación de una ontología de calidad de datos, para la evaluación y mejora de la calidad de los datos utilizados como base para la toma de decisiones en el área de finanzas. La ontología es una especificación explícita de un modelo abstracto sobre un dominio de conocimiento particular, con el fin de resolver problemas de calidad de datos. Hace mención a la disciplina de calidad de datos y sus dimensiones. Define problemas de calidad de datos, describe cómo identificarlos y presenta un caso de aplicación.

Otra metodología más genérica y no específica de Calidad de Datos que es aplicada también por algunos autores es GQM (*Goal Question Metric*) [43]–[45]. A partir del planteo de objetivos y preguntas, se obtienen métricas que constituyen el punto de partido para el análisis sistemático de la calidad de datos.

2.6 Modelos de Calidad de Datos

Un modelo de calidad de datos representa el conjunto de dimensiones y otros aspectos de la calidad de datos (tales como factores, métricas y mediciones) y las relaciones que existen entre ellos, y que proporciona la base para especificar y evaluar la calidad de las entidades. Para cada caso particular, se podría llegar a definir un modelo de calidad válido [46].

No se encontraron trabajos que definan modelos de calidad que resulten replicables a otros estudios. Por otra parte, mientras que la literatura ofrece diferentes dimensiones y *frameworks* para identificar los aspectos de calidad de datos que son importantes, así como técnicas para su evaluación y mejora, son pocos los que describen la metodología utilizada o proponen métricas de calidad concretas [19], [35].

En [47] se presenta un modelo de calidad extendido enfocado en mejorar la calidad de datos en las organizaciones, incluyendo causas de imperfecciones en los datos desde el punto de vista práctico (calidad en el valor y en representación de los datos) y estructural (calidad en el modelo y en arquitectura de los datos).

En [48] se propone un modelo de calidad de datos para portales web (PDQM). Aunque no utiliza el enfoque de dimensiones y factores de calidad, el modelo incluye atributos de calidad tomados de la literatura de Calidad de Datos (que podrían ser mapeados con las dimensiones y factores), así como expectativas de los consumidores de datos y funcionalidades ofrecidas al usuario en el contexto de un portal web. Las fases propuestas por el modelo son las siguientes: Identificación de atributos de calidad de datos en la web, Definición de matriz de clasificación, Clasificación de atributos de calidad en la matriz, Validación, y PDQM (Portal Data Quality Model – Modelo de Calidad de Datos para Portales).

2.6.1 Métricas

En líneas generales las métricas propuestas por los trabajos relacionados son genéricas y no describen cómo aplicarlas, o se refieren a dominios específicos o casos particulares. Raramente se presentan mediciones de calidad de datos concretas, o al menos la forma de cuantificar el impacto de la calidad de datos.

Actualmente, la mayoría de las métricas de calidad de datos son generadas de manera ad-hoc de forma de resolver problemas específicos. Es por esto que algunos trabajos describen guías que pueden contribuir con las organizaciones en este sentido. La necesidad de definir métricas dependientes de cada contexto particular es inevitable dada que la definición de calidad de datos ("*fitness for use*") es sensible al contexto [11], [35].

La experiencia ha demostrado que no es posible proponer un conjunto de métricas que sean aplicables en toda situación. Por el contrario, la definición de métricas de calidad es un esfuerzo continuo que requiere conocer el contexto y de los principios de la disciplina [11]. La interrogante planteada es entonces [35]: *¿cómo pueden las compañías identificar las métricas de calidad de datos que son relevantes para sus procesos de negocio?* De la misma manera, *¿cómo puede la academia identificar las métricas de valor para analizar la calidad de los datos que utilizan?*

Las métricas son el instrumento que permiten obtener valores de calidad concretos para las dimensiones de calidad de datos que son de interés [9]. Se define "métrica de calidad de datos" como la medición cuantitativa del grado en que los datos poseen un determinado atributo de calidad. Por cada métrica, se debe especificar el método de medición (dónde), el objeto sobre el cual se aplica (qué), la herramienta (cómo) y la escala de medición. Concluyen que el proceso para identificar métricas de calidad de datos relevantes es una tarea compleja que debe ser soportada no sólo por un proceso que sirva de guía, sino que también por un repositorio que contenga métricas de calidad ya implementadas y que sirvan de buenas prácticas.

Algunas métricas propuestas, que operan como funciones de agregación [11] incluyen: ratio simple (cantidad de valores sin error en el total), operaciones de máximo y mínimo, promedios ponderados.

Otra de las propuestas [8] plantea incluir la perspectiva de calidad de datos como parte del diseño de la base de datos mediante la definición de "*tags*". El objetivo es desarrollar una metodología para determinar qué aspectos de la calidad de datos son importantes y qué etiquetas deben contener los datos. Cada etiqueta está compuesta por parámetros (dimensión subjetiva, necesidades) e indicadores (dimensión objetiva, medible) de calidad.

2.6.2 Modelos de Calidad de Datos para dominios específicos

Algunos trabajos proponen modelos de calidad de datos para contextos particulares.

El estudio en [49] muestra una encuesta sobre cómo se maneja la Calidad de Datos en organizaciones de Australia. Para ello se basan en las propuestas de *Wang & Strong's* [12] de Calidad de Datos, teniendo en cuenta el concepto multifacético que la define. La aplicación de calidad de datos está dada durante el armado de la encuesta. Los resultados muestran que a pesar de que la calidad de los datos se considera una temática de suma relevancia, crítica para la toma de decisiones efectiva dentro de las organizaciones, poco se hace para mejorarla. Resultados similares a estos encuentra *Liebchen* [2] cuando hace la revisión sistemática en el contexto de la ingeniería de software empírica (esto se trata en detalle en la sección ...). Parte de la motivación radica en que aquellos datos de mala calidad que sean identificados pero no corregidos, pueden generar destrozos económicos y tener un impacto social importante en la organización. Además, se muestra que existe una desconexión en

referencia a la calidad de datos entre los custodios, los productores y los dueños de los datos. Concluyen que la mayor parte de las organizaciones en Australia no tienen planes de análisis ni mejora la calidad de los datos [49].

Existen algunos ejemplos de modelos de calidad de datos también aplicados al campo de la medicina. En el contexto de la disciplina biomédica [45] se propone un modelo para evaluar la calidad de datos para el dominio específico de estudios de genomas (*GWAS – Genome Wide Association Studies*), y se presentan los problemas de calidad con los cuales se enfrentan. Este modelo de calidad fue desarrollado considerando los requerimientos para la técnica estadística de meta-análisis, que es utilizada para construir nuevos *GWAS*.

Por otra parte, en [50] se define un modelo de calidad específico para datos provenientes de sensores utilizados para el monitoreo de la salud de personas mayores que permanecen en su hogar. Para esto, se definen un conjunto de dimensiones de calidad de datos que son de interés en este dominio de estudio particular, y se especifican un conjunto de métricas de calidad que permitan medir las dimensiones definidas sobre los datos que corresponde.

Existen también estudios aplicados al campo de empresas y negocios financieros. En [51] se experimenta en una aplicación real de un área de negocio financiera, sobre el dominio de aplicación *CRM* (*Costumer Relationship Management*). Para medir la calidad se pone en práctica una metodología en la que las métricas de calidad se obtienen refinando las metas de calidad de la organización. Como resultado se obtiene una biblioteca de métodos de medición de calidad y una base de datos con las medidas tomadas para la aplicación financiera. Los métodos propuestos son parametrizables y extensibles, pudiendo ser utilizados en otras aplicaciones.

Otro estudio [52] planteado en el contexto de la gestión del riesgo de crédito, propone un conjunto de dimensiones que son aplicables en este dominio particular y que son obtenidas en base a entrevistas. Sugiere la aplicación de *TDQM* (Programa para la Gestión de la Calidad de Datos Total) que consiste en las fases de: Definición de Calidad de Datos, Medición, Análisis, y Mejora. Este artículo realiza un estudio empírico mediante la distribución de un cuestionario en instituciones financieras de forma de identificar las dimensiones de calidad más importantes. Luego se evalúa el nivel de calidad de sus repositorios de datos a partir de estas dimensiones, se analizan los problemas de calidad existentes y se sugieren acciones de mejora. El cuestionario está estructurado según la propuesta de *Wang & Strong's*, e incorpora tres nuevas dimensiones que se consideran importante incluir en este contexto de aplicación.

Finalmente, existen estudios para la calidad de datos de experimentos pero alejados del contexto de la ingeniería de software. En particular, en [53] se analiza la calidad de datos que son recolectados para experimentos biométricos. Es interesante ver cómo en otras áreas científicas también es relevante que los datos gocen de buena calidad. El foco está en la correctitud, consistencia y disponibilidad de los datos.

2.7 Limpieza y gestión de Calidad de Datos

La limpieza de datos (*data cleaning* o *data cleansing*) [40], [54] intenta resolver la problemática de la detección y corrección de errores en los datos, con el fin de mejorar su calidad. Estas actividades son de mayor importancia en las bases de datos en las cuáles la información se ingresó de alguna manera que deja lugar a la aparición de errores. Por ejemplo, cuando la ingresan personas desde el teclado, cuando se obtiene de fuentes no muy confiables o cuando se integran diferentes fuentes de información.

Existen variadas herramientas que dan soporte a la limpieza de datos. Sin embargo, además de la utilización de herramientas, se requiere de un arduo trabajo manual o de programación de bajo nivel para su resolución.

Alguno de los problemas que enfrenta la limpieza de datos en las fuentes de información tienen sus orígenes en las restricciones de integridad y el esquema de datos en el cual se encuentran inmersos. Por ejemplo, las bases de datos relacionales tienen menor probabilidad de poseer errores e inconsistencias en los datos, a diferencia de los archivos de texto plano o planillas de cálculo en los cuales no existe ningún tipo de reglas ni restricciones con respecto a los datos ni sus valores.

Un proceso de limpieza de datos consta básicamente de las siguientes fases.

1. *Análisis de datos*: consiste en determinar los errores e inconsistencias que deberán eliminarse. Para ello se realiza una inspección manual y se utilizan programas de análisis de datos.
2. *Definición de transformaciones de datos y reglas de mapeo*: consiste en un conjunto de pasos durante los cuales se llevan a cabo transformaciones a nivel del esquema y de las instancias. Para ello se pueden utilizar herramientas de *ETL (Extraction, Transformation, Loading)*, sentencias *SQL (Standar Query Language)* o funciones definidas por el usuario *(UDF - User Defined Functions)*.

2.7.1 Actividades de la Calidad de Datos

Las actividades relativas a la calidad de datos se refieren a cualquier proceso (o transformación) que se aplica a los datos con el objetivo de mejorar su calidad. Para llevar a cabo dichas actividades, se hace uso de distintas técnicas [9], [40]. A continuación se describen las principales actividades relativas a la calidad de los datos, algunas de las cuales se detallan a lo largo de esta sección.

- *Obtención de nueva información*: es el proceso de refrescar la información almacenada en la base con datos de mayor calidad (por ejemplo ingresar datos más precisos, de mayor actualidad).
- *Estandarización*: es el proceso de "normalizar" los datos almacenados, de manera que queden almacenados respetando cierto formato (por ejemplo todos los números de teléfono deben incluir el código de región, el sexo debe contener los valores F/M).
- *Identificación de Objetos*: es el proceso por el cual se identifican registros (del mismo o diferentes repositorios) que hacen referencia al mismo objeto de la realidad. Podría suceder que los datos que representan al mismo objeto resulten ser complementarios, duplicados o contradictorios.
- *Integración de datos*: hace referencia a la actividad de unificar datos provenientes de distintas fuentes, resolviendo los problemas que esto trae aparejados (redundancias, problemas de consistencia, duplicación).
- *Confiabilidad de las fuentes*: implica "calificar" a las distintas fuentes de información de acuerdo a la calidad de los datos que proveen (por ejemplo en un sistema P2P).
- *Composición de calidad*: hace referencia a la definición de un álgebra para calcular la composición (o agregación) de las medidas de las dimensiones de calidad de datos. Por ejemplo, calcular la completitud de una unión de relaciones, a partir de la completitud de cada relación.
- *Detección de errores*: dadas una o más tablas, y ciertas reglas que los registros de dichas tablas deben cumplir, este es el proceso de detectar qué registros no cumplen con dichas reglas.
- *Corrección de errores*: luego de la detección, esta actividad es responsable de corregir los registros con errores, de manera que se respeten todas las reglas correspondientes.

- *Optimización de costos*: implica obtener la mejor relación costo-beneficio al aplicar procesos de mejora de la calidad de los datos.

2.7.2 Detección y corrección de errores

Dependiendo de la naturaleza del error encontrado, el foco puede estar en:

- *Detectar y corregir inconsistencias*. Se trata de detectar registros que no cumplan con determinadas reglas, y luego modificar los datos para que las cumplan. Esta tarea incluye asegurar que la información se encuentra consistente (sin contradicciones) y libre de redundancias.
  Una técnica para la detección de inconsistencias es la llamada *Data editing*, la cual consiste en la definición de reglas (*edits*) que deben ser respetadas por cierto conjunto de datos. Los *edits* representan condiciones de error, por lo cual deben ser consistentes y no redundantes. Los datos de un registro deben ser ajustados de manera tal que cumplan con las reglas, pero minimizando la cantidad de modificaciones a los datos.
  Existen varias formas de corregir los errores detectados: refrescando la base de datos con nuevos datos (a partir de la obtención de nueva información), o utilizando los edits definidos de manera tal que cuando no se cumple una regla, se imputa un valor que haga que la misma sea verdadera.
- *Detectar y corregir datos incompletos*. Si se considera una base de datos relacional, el caso de incompletitud más claro son los valores nulos. Si bien es muy simple detectar los datos incompletos, corregirlos puede ser una tarea difícil (si no se puede obtener la información faltante).
- *Detectar y corregir anomalías*. Este es el caso de datos cuyo valor difiere en gran medida con respecto a los demás datos. La situación puede ser que: el valor fue mal medido, o mal ingresado en la base; el valor corresponde a una "muestra" distinta a la de todos los demás; o el valor es correcto y simplemente corresponde a algún suceso inusual de la realidad.
  Existen varias técnicas para identificar anomalías. Una de ellas calcula el valor promedio y la desviación estándar de cierto conjunto de datos, para identificar aquellos valores que se desvíen "demasiado" del valor promedio. Se podría definir por ejemplo un valor límite a partir del cual el dato es sospechoso de ser incorrecto. Otras técnicas utilizan también el factor tiempo para identificar datos anómalos, partiendo de la base que datos medidos o registrados en cierto lapso de tiempo pueden estar altamente relacionados, y también teniendo en cuenta posibles ciclos donde aparezcan "picos" en los valores.
  Lidiar con estas anomalías implica un doble esfuerzo: primero hay que identificarlas, y luego decidir si corresponden a datos correctos de sucesos de la realidad poco comunes, o si corresponden a datos incorrectos y deben ser corregidos.

2.7.3 Prevención de errores

Consiste en evitar que ocurran errores en los datos a futuro. Para ello es necesario identificar cuáles son las causas de los errores, para intentar eliminarlos de manera permanente.

En general, la localización y corrección de errores se lleva a cabo para datos cuya creación y actualización es poco frecuente. Sin embargo, la prevención de errores a través del manejo de procesos es utilizada en mayor medida cuando los datos son actualizados y creados de manera frecuente. Se incluyen controles a los procesos en los cuales los datos son creados y/o actualizados para evitar que sucedan inconsistencias.

Los *edits* también pueden ser utilizados para la prevención de errores y la mejora de procesos, evitando la ocurrencia de ciertas inconsistencias en la base.

Otra forma de prevención de errores consiste en identificar cuáles con las actividades manuales en las cuales suelen ocurrir la mayor cantidad de errores, y buscar su automatización.

La mejor forma de mejorar la calidad de los datos consiste en perfeccionar la fase de recolección [55]. Sin duda lo ideal es prevenir la ocurrencia de errores antes que corregirla, ya que el costo e impacto asociado será significativamente menor en el primer caso.

Capítulo 3: Experimentación en Ingeniería de Software

En este capítulo se presenta la experimentación de forma general, cómo es aplicada a la Ingeniería de Software, y por qué es importante en esta área. Finalmente se describen los conceptos básicos que hacen al diseño de un experimento. Este capítulo se basa fuertemente en [7].

3.1 Introducción

La experimentación se refiere a la correspondencia de las suposiciones, asunciones, especulaciones y creencias acerca de algo con hechos de la realidad. En general sucede que si ciertas ideas son tomadas como verdaderas y utilizadas por una gran cantidad de personas, entonces pueden llegar a convertirse en certeras para toda la sociedad. Por el contrario, si una idea es descartada por la sociedad perderá validez y dejará de ser utilizada.

Esta metodología "natural" de selección acerca de las ideas verdaderas no es apropiada para las disciplinas de la Ingeniería, entre ellas para la Ingeniería de Software. Esta última requiere pruebas que tengan sus raíces en hechos de la realidad (no en supuestos), que establezcan si un determinado enfoque o técnica es realmente mejor o peor que otra. A pesar de esto, son pocas las ideas de la Ingeniería de Software que se prueban con hechos de la realidad, con datos empíricos, y menos aún las que siguen un enfoque formal para la experimentación. Aún así, muchas de estas ideas no probadas mediante experimentación son asumidas como válidas y utilizadas por toda la comunidad científica.

La experimentación permite confirmar teorías, conocer los factores que hacen a un software bueno o mejor que otro, así como las técnicas, métodos y herramientas más apropiadas para desarrollar software bajo determinadas situaciones. Sin embargo, en la actualidad, la Ingeniería de Software no cuenta con un nivel de desarrollo en materia de experimentación formal comparable con otras disciplinas de la Ingeniería.

3.1.1 Tipos de estudios empíricos

A nivel general, se identifican dos enfoques para la investigación empírica:

- Estudios cuantitativos. Su objetivo es obtener una relación numérica entre las variables o alternativas en cuestión. Los datos obtenidos para este tipo de estudio son siempre valores numéricos.
- Estudios cualitativos. Su objetivo es intentar explicar las formas en que determinados objetos (tales como las personas) manejan sus comportamientos en entornos particulares. Se enfoca en obtener una visión integral del contexto bajo estudio. Los datos obtenidos para este tipo de estudio son textos, gráficos o imágenes.

Por otro lado, existen estudios cuantitativos subjetivos, cuando las personas brindan los datos desde su punto de vista, y cuantitativos objetivos, cuando los datos son obtenidos por ejemplo a partir de una herramienta. Análogamente existen estudios cualitativos subjetivos y objetivos.

Los estudios son llevados a cabo desde un enfoque cualitativo o cuantitativo, o incluso ambos, dependiendo de la realidad que se examina. Si bien las investigaciones cuantitativas pueden llegar a resultados más formales y justificables, las cualitativas complementan a estas para definir el cuerpo del conocimiento de cualquier disciplina. De esta manera, ambos enfoques resultan complementarios.

3.1.2 Amplitud de los estudios experimentales

Existen tres tipos de roles que requieren cierto grado de experiencia, y que deben formar parte del proceso de prueba de una idea o teoría para alcanzar su validez:

- Investigadores. Llevan a cabo experimentos en sus laboratorios, bajo condiciones controladas y sin presiones del mercado, para comprobar la validez de su propuesta. Las teorías y resultados son luego publicados, para que otros investigadores puedan también replicar el experimento y publicar nuevos resultados.
- Innovadores. Llevan a cabo el experimento en proyectos reales, y se hacen responsables por los riesgos que corren a cambio de poder experimentar con lo último en innovación. Deben luego publicar sus resultados, estableciendo cuándo falló el experimento y cuándo no, y qué mejoras deberían realizarse al mismo.
- Desarrolladores rutinarios. Luego de superados los dos niveles anteriores, y a la luz del riesgo que se asume y de las fallas y mejorías requeridas por el experimento, los usuarios podrán aplicarlo a proyectos reales. Se publica luego su comportamiento indicando bajo qué circunstancias fue sometido el experimento.

En particular, en este trabajo se definen tres roles principales: el *Analista en Calidad de Datos*, responsable de ejecutar las actividades referentes a la calidad de datos; el *Responsable del Experimento*, quien tiene el conocimiento profundo y necesario sobre el experimento; y el *Responsable del Modelo de Calidad de Datos*, responsable del mantenimiento del modelo de calidad. Estos roles se presentan en el Capítulo 5 y Capítulo 6.

3.2 ¿Por qué experimentar?

La investigación es una actividad llevada a cabo de manera voluntaria y consciente con el fin de encontrar un conocimiento certero sobre alguna cuestión en particular. Cierto conocimiento será considerado científicamente válido, siempre y cuando su validez haya sido demostrada, y además exista una comprobación de este conocimiento contra la realidad. El conocimiento probado es de suma importancia ya que permite predecir el comportamiento de los elementos en juego (por ejemplo, a partir de las leyes de *Newton* es posible calcular la fuerza neta de un objeto). De manera contraria a las opiniones que son meramente subjetivas, las investigaciones científicas son estudios objetivos basados en la observación del mundo real y la experimentación con éste.

3.2.1 El factor humano en la Ingeniería de Software

Un elemento crucial que es necesario considerar en el área de la Ingeniería de Software es la influencia del factor humano, como ser la experiencia, conocimiento y capacidad de las personas, en el uso de sus artefactos (métodos, herramientas, paradigmas). El elemento humano es importante para esta disciplina, ya que la misma se ve influenciada por las relaciones entre las personas (tales como el equipo de un proyecto), así como por su contexto social (la cultura organizacional, por ejemplo).

El aspecto social se convierte entonces en una dificultad a la hora de llevar a cabo experimentos, haciendo que los mismos resulten más complejos. Más que convertirse en una excusa para no experimentar, debería servir de impulso para adquirir mayor conocimiento en este aspecto, y lograr así realizar experimentos que consideren al factor social y su influencia de manera apropiada.

3.2.2 El método científico

Las actividades que se llevan a cabo en toda investigación científica son las que siguen:

- Interacción con la realidad. Esta actividad puede realizarse mediante la observación (pasiva), caso en el que los investigadores perciben cosas de la realidad sin interferir ni tener control

sobre ella. O también mediante la experimentación (activa), en cuyo caso los investigadores someten al objeto en cuestión a nuevas condiciones y observan sus reacciones, interfiriendo y controlando la realidad.

- Especulación. Los investigadores formulan hipótesis acerca de su percepción del mundo real. En el campo de la Ingeniería de Software, esta actividad comprende encontrar relaciones entre las variables en juego para predecir las consecuencias en el proceso de desarrollo mismo y en el producto resultante.
- Confrontación con la realidad. Consiste en el chequeo de las especulaciones teóricas (ideas) contra la realidad (hechos).

En este trabajo se aplican técnicas de calidad de datos a los datos generados durante la aplicación del método científico en el contexto de la Ingeniería de Software.

3.2.3 Replicación en experimentos

Comprobar las ideas contra la realidad no es suficiente para la experimentación. Es necesario además proveer a la comunidad con los datos requeridos para que el experimento pueda ser replicado por agentes externos y verificar sus resultados.

Sin embargo, en el área de la Ingeniería de Software la replicación se convierte en similitud: resulta prácticamente imposible replicar dos experimentos de manera exacta (mismas personas, misma experiencia, mismo proceso de desarrollo, mismos productos.) Es necesario definir entonces las características de un proyecto de desarrollo que harán que dos proyectos similares se conviertan en "idénticos". Una vez más, es el factor humano el que hace esta tarea más compleja por ser el más variado e impredecible.

Se distinguen dos objetivos a la hora de llevar a cabo la replicación de un experimento. Si la replicación se realiza bajo condiciones similares al original, entonces el objetivo resulta ser la confirmación de la hipótesis del primer experimento. Si por el contrario una variable es modificada durante la replicación, entonces el objetivo es chequear si dicha variable podría ser generalizada para ciertos valores.

3.3 ¿Cómo experimentar?

El objetivo de todo experimento que estudia determinado fenómeno, es establecer (o descubrir) las relaciones que existen entre las variables involucradas en dicho fenómeno, con el fin de lograr predecir cuál será el comportamiento de las mismas bajo determinadas circunstancias.

Se distinguen tres categorías de relaciones entre variables, dependiendo de cuánto se conoce sobre la relación:

- Relaciones descriptivas. Son relaciones donde sólo se conoce cierto patrón de comportamiento, pero sin tener medida de cuánto afecta una variable a otra.
- Correlaciones. Este tipo de relación sucede cuando se conoce que cierta(s) variable(s) afectan a una tercera, de acuerdo a determinada función conocida. No necesariamente hay una teoría de fondo, por lo que no se puede distinguir entre causa y efecto.
- Relaciones causales. Existe este tipo de relación cuando, por ejemplo, se sabe que las variables A y B causan todos los cambios en la variable C. O sea, la variable C varía dependiendo solamente de los valores de A y B. Es el grado máximo de conocimiento que se puede tener sobre una relación. A este tipo de relación se lo conoce como causalidad determinista, pues dada la causa siempre se obtiene el efecto esperado. Existe también la causalidad probabilís-

tica, donde dada la causa se sabe que se obtendrá el efecto esperado con una probabilidad menor a 1.

3.3.1 Fases de un experimento

Se definen cuatro fases para un experimento:

- Definición de objetivos. Se transforma la hipótesis general en términos de las variables del fenómeno a examinar. Es importante definir un procedimiento cuantitativo con el fin de evaluar la hipótesis.
- Diseño. Implica definir un plan para la ejecución del experimento. Se deben definir todas las condiciones bajo las cuales se llevará a cabo el experimento, como ser las variables que lo afectarán, quiénes van a participar, cuántas veces se repetirá el experimento, entre otras. Un buen diseño apunta a obtener la mayor cantidad de conocimiento posible, en la menor cantidad de experimentos.
  Durante esta etapa se definen qué datos serán recolectados, por quiénes (experimentadores o sujetos), de qué forma (manual o automática), y dónde quedarán almacenados (base de datos, planillas de cálculo, entre otros).
- Ejecución. Se ejecuta el experimento de acuerdo al diseño.
  Luego de diseñar y planificar el experimento, éste debe ser ejecutado para recolectar los datos que se quieren analizar. La operación del experimento consiste en las tres etapas siguientes.
  - Preparación: selección de sujetos y preparación de artefactos utilizados.
  - Ejecución: durante la ejecución se recolectan los datos necesarios, según lo definido en el diseño experimental. Los datos pueden ser recolectados de las siguientes formas: manualmente mediante el llenado de formularios por parte de los sujetos; manualmente soportado por herramientas; mediante entrevistas, automáticamente por herramientas.
  - Validación de los datos: cuando se obtienen los datos, se debe chequear que fueron recolectados correctamente y que son razonables. Es importante revisar que el experimento sea ejecutado en la forma en que fue planificado. De lo contrario los resultados podrían ser inválidos.
- Análisis del resultado. Se analizan los datos obtenidos durante el experimento, en busca de relaciones entres las variables consideradas.
  Luego de que finaliza la ejecución del experimento y se cuenta con los datos recolectados, comienza la fase de análisis de los mismos conforme a los objetivos planteados. Después de obtener los datos es necesario interpretarlos para llegar a conclusiones válidas. En esta última fase, se aplican conceptos estadísticos para analizar los datos. Si los datos que son utilizados para obtener los resultados de los experimentos contienen errores o no reflejan la realidad, entonces los análisis resultantes podrían ser incorrectos.
  Para identificar correlaciones o relaciones causales, hace falta aplicar técnicas de análisis de datos, las que implican realizar análisis estadísticos sobre los datos. Uno de los estudios más comunes que se realiza sobre los datos se conoce como “test de significancia”, el cual tiene como objetivo establecer si las variaciones observadas en los datos recolectados tienen significado estadístico.

3.3.2 La importancia de los datos en el contexto de un experimento

Los datos que son recolectados, generados y almacenados durante la ejecución de un experimento son de suma importancia y valor, ya que constituyen la base para realizar los análisis estadísticos y obtener los resultados del experimento.

A pesar de que se propone incorporar una etapa validación de datos durante la fase de ejecución del experimento, no encontramos en la literatura procedimientos, enfoques sistemáticos o protocolos sobre cómo llevar a cabo estas validaciones o chequeos en el contexto de experimentos en ingeniería de software. El análisis, evaluación y mejora de la calidad de datos resultantes de experimentos en el contexto de la ingeniería de software es un tema de gran relevancia pero que no ha tenido la atención que se merece [2], [3]. Esto se presenta en profundidad en el Capítulo 4.

3.4 Conceptos básicos sobre el diseño de experimentos

A continuación se introducen los principales conceptos sobre el diseño de experimentos, y se presenta un ejemplo para introducirlos. Se considera un experimento que consiste en la evaluación de la efectividad de un analgésico en personas de entre 25 y 40 años de edad, llamado Efec-Analgésico.

- *Unidad experimental (experimental unit)*: son los objetos sobre los cuales es llevado a cabo un experimento.
  En Efec-Analgésico la unidad experimental es el grupo de personas entre 25 y 40 años.
- *Sujetos del experimento (experimental subjects)*: son las personas que llevan a cabo el experimento. Debido a que el factor humano influye de manera significativa en los resultados de un experimento dentro de la Ingeniería de Software (el resultado será distinto dependiendo de la persona que lo aplique), es imprescindible considerar el efecto de esta variable en el diseño de los experimentos.
  Los sujetos en Efec-Analgésico son quienes administran los analgésicos a los pacientes, por ejemplo enfermeros.
- *Variable de respuesta (response/dependent variable)*: es el resultado cuantitativo de un experimento.
  En Efec-Analgésico podría ser el grado en que el analgésico calma el dolor, o la rapidez con la actúa.
- *Parámetros (parameters)*: son características que se mantienen invariadas durante el experimento, y por lo tanto no se desea que afecte el resultado del mismo. Los parámetros pueden ser cualitativos o cuantitativos. De esta manera, los resultados arrojados por un experimento serán ciertos sólo bajo las condiciones impuestas por los parámetros.
  En el caso de Efec-Analgésico un parámetro a considerar es el rango de edades (25 a 40 años).
- *Factores (factors, predictor/independent variables)*: son características variadas intencionalmente durante el experimento, y por lo tanto afectan el resultado del mismo.
  El factor en Efec-Analgésico es el analgésico que se utilice.
- *Alternativas (alternatives, levels, treatment)*: son los valores posibles que pueden tomar los factores durante un experimento.
  Las alternativas en el caso de Efec-Analgésico serían los distintos analgésicos que se utilicen (Perifar, Zolben,).
- *Interacciones (interactions)*: suceden cuando el efecto de un factor depende del valor de otro. Debido a que influencian el resultado del experimento, deben ser considerados en su diseño.

- *Variaciones no deseadas (undesired variations, blocking variables)*: son variables inevitables que afectan el resultado del experimento.
  En Efec-Analgésico podrían provocarse este tipo de variaciones debido a la distinta respuesta que presenta cada persona a los fármacos.
- *Experimento unitario (elementary/unitary experiment)*: cada aplicación de una combinación de alternativas de factores llevada a cabo por sujetos sobre una determinada unidad experimental.
  Para el caso de Efec-Analgésico un experimento unitario involucra el analgésico x, aplicado por el enfermero i al paciente j.
- *Replicación externa (external replication)*: replicaciones de un experimento llevadas a cabo por otros investigadores y con diferentes muestras. En la Ingeniería de Software la replicación exacta de un experimento no es posible, por lo tanto nos referimos a "la mayor similitud posible".
- *Replicación interna (internal replication)*: repetición de uno o más experimentos unitarios. La cantidad de repeticiones que serán llevadas a cabo en un experimento debe ser establecido durante el diseño del mismo.
- *Error experimental (experimental error)*: se refiere a las variaciones inevitables que ocurren entre repeticiones, tales como errores en la medición de los resultados, o variables no consideradas. Estas últimas pueden llegar a invalidar el resultado de un experimento.

La Ilustración 3 muestra cómo se relacionan los principales conceptos sobre el diseño de un experimento.

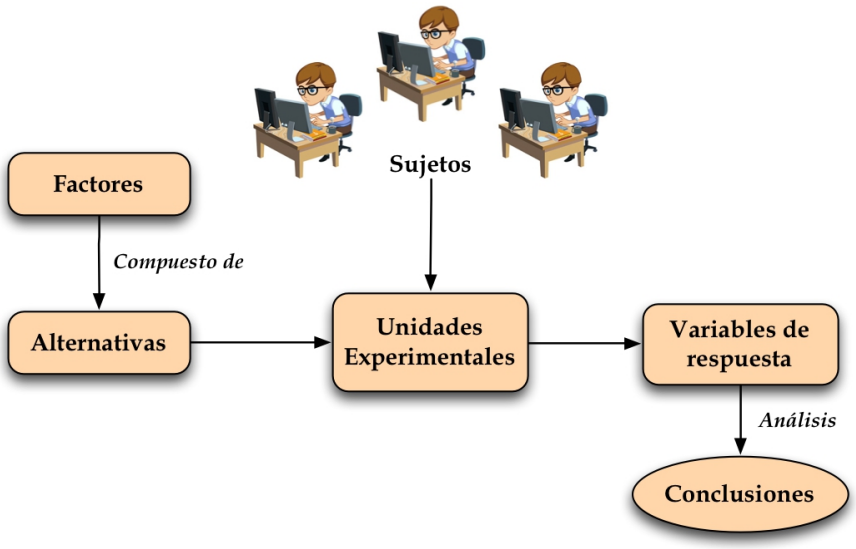

Ilustración 3: Diseño Experimental

Capítulo 4: Trabajos Relacionados sobre Calidad de Datos aplicada a Ingeniería de Software Empírica

En este capítulo se presentan los principales trabajos relacionados que fueron encontraron y analizados durante la revisión de la literatura. Este capítulo contiene dos secciones que presentan cómo se han aplicado técnicas y conceptos de Calidad de Datos a repositorios que contienen datos de Ingeniería de Software, y más específicamente para datos de Ingeniería de Software Empírica. En la última sección también se presentan los resultados encontrados por las principales revisiones de la literatura que fueron realizadas por otros autores en este mismo dominio.

4.1 Calidad de Datos aplicada a Ingeniería de Software

En esta sección se describe de qué manera se han aplicado diferentes técnicas, herramientas y conceptos de Calidad de Datos a repositorios que contienen datos de Ingeniería de Software.

Los datos en Ingeniería de Software provienen básicamente de tres fuentes primarias [55]:

1. Datos recolectados mediante la experimentación, observación y estudios retrospectivos.
2. Métricas del software o datos reportados por la gestión del proyecto incluyendo esfuerzo, tamaño y estimación de hitos del proyecto.
3. Artefactos del software incluyendo requerimientos, diseño, inspección de documentos, código fuente y la historia de sus cambios, seguimiento de errores, bases de datos de testeo.

Dichas fuentes de datos de Ingeniería de Software, por otra parte, se encuentran almacenadas de forma de ser utilizadas para algún propósito dado. De acuerdo a los trabajos analizados, se realiza la siguiente clasificación según su uso o aplicación: predicciones, estimaciones y heurísticas; procesos y proyectos de desarrollo de software; modelos de predicción y máquinas de aprendizaje; análisis estadísticos; experimentos; estudios empíricos (en general).

La tabla 1 muestra a modo de resumen cómo se distribuyen los artículos analizados según las clasificaciones antes mencionadas (para el dominio de aplicación de datos de Ingeniería de Software).

El foco de nuestro trabajo está, dentro de la fuente primaria 1), en los datos recolectados mediante experimentos en Ingeniería de Software que involucran seres humanos, y que serán utilizados para obtener análisis, conclusiones y los resultados que se definan a partir del diseño experimental.

Desde hace varios años, la comunidad de Ingeniería de Software ha demostrado interés y preocupación por que los datos que utiliza sean de buena calidad [56]. Existen varios trabajos en el área de Ingeniería de Software que analizan la calidad de los datos que utilizan, o al menos algún aspecto de esta (en general la completitud o exactitud de los datos). Sin embargo, sólo algunos mencionan de manera explícita que se trate de un estudio de calidad de datos, o utilizan la disciplina Calidad de Datos como marco conceptual. Tampoco se plantea un proceso, metodología o modelo que muestre de forma sistemática cómo fue llevado a cabo el análisis, y que facilite su replicación en otros estudios.

El artículo de *Basili y Weiss* [56] publicado en 1984, tiene como objetivo mostrar cómo obtener datos válidos que serán utilizados para conocer acerca del proceso de desarrollo, así como para evaluar metodologías de desarrollo de software. Plantea la necesidad de incluir un proceso de validación en la recolección de datos. Sin este proceso, hasta un 50% de los datos podrían contener errores. Establece que los datos recolectados en ingeniería de software deberían ser correctos, consistentes y completos. Desde el punto de vista de la calidad de datos, estos atributos se refieren a dimensiones o aspectos de calidad, y en definitiva, lo que se plantea es que los datos gocen de buena calidad.

Fuente Primara / *Uso/Aplicación*		**Experimentos, observación, estudios retrospectivos**	**Métricas de SW, gestión de proyectos**	**Artefactos del SW (gestión de cambios, defectos, configuración, ...)**	**Revisión de Literatura**
Ing de SW	Predicciones, estimaciones y heurísticas		[60], [61]	[56], [74], [75]	
	Procesos y proyectos de desarrollo de software			[76], [77]	
Ing de SW Emp	Modelos de predicción y máquinas de aprendizaje		[62]–[68]	[62], [73], [78]–[80]	[25], [26]
	Análisis estadísticos		[69]–[73]	[69]–[71]	
	Experimentos	[57]–[59]		[24], [80]	
	Estudios empíricos (en general)		[2], [15], [55]	[3], [23], [28], [55], [81]	[1], [2], [14]

Tabla 1: Clasificación de artículos analizados

La importancia de la calidad aplicada a datos de ingeniería de software se ha ido incrementando a lo largo de los años. Esto lo demuestra la cantidad de estudios y publicaciones que existen en esta área por cada año [2].

Aún en tiempos corrientes, se continúan realizando publicaciones en el área indicando que el análisis de la calidad de los datos sigue siendo un tema de gran relevancia [62]. El trabajo presentado en [62] clasifica la información de ciertos repositorios de ingeniería de software, analiza la calidad de sus datos, y presenta los problemas de calidad con los que se enfrentan los investigadores al aplicar técnicas de estadísticas o de máquinas de aprendizaje sobre los datos. Estos problemas de calidad tienen un foco estadístico y en minería de datos (*data mining*), y no en la calidad de datos como tal.

Existen estudios que analizan la completitud de los datos y el impacto que tienen los valores faltantes en la ingeniería de software. El trabajo en [55] consiste en revisar, proponer y aplicar diferentes técnicas para lidiar con la problemática de incompletitud, y de esta manera mejorar la calidad de los mismos. Uno de los objetivos es concienciar a quienes realizan estudios en ingeniería de software, sobre la importancia de conocer y aplicar técnicas para valores faltantes. Establece que los investigadores en ingeniería de software en general no tienen en cuenta esta problemática, resultando en análisis que tienen desviaciones respecto a la realidad. Una vez más se plantea que la calidad de los datos recolectados y del proceso de recolección influirá significativamente en los resultados obtenidos.

Hay otros trabajos [68], [82] cuyo foco está en determinar si ciertos métodos de imputación de datos mejoran la calidad de los repositorios de ingeniería de software que contienen valores nulos. Se plantea la falta de guías metodológicas y técnicas para que los investigadores en ingeniería de software traten la problemática de incompletitud. Concluye que los datos sobre los cuales se realizan análisis deben ser cuestionados antes de someterlos a un análisis.

También se proponen diferentes técnicas para no considerar los valores faltantes (eliminarlos o imputarlos), y se muestra que de esta manera hay una mejora en los resultados obtenidos [63], [64]. Se establece que los valores faltantes son una problemática conocida para la construcción de modelos

de estimación de costos, y que pueden impactar en los factores de productividad que se obtienen a partir de la información histórica.

Otro conjunto de artículos analizan la calidad de los datos contenidos en bases de datos de *bugs* y cambios (generados durante el proceso de desarrollo de software). El foco está en la falta de vínculos (*links*) que existen entre los reportes de *bugs* y cambios aplicados al software [23], [28], [74], [75], [81]. La importancia de que los *logs* de cambios y reportes de errores gocen de buena calidad, radica en que estos datos son la base para construir las heurísticas que permiten encontrar los *links* de forma automática [75], realizar predicciones para futuros proyectos [28], [74], así como para el seguimiento, estimación y planificación de proyectos [74].

En particular en [75] el foco está en la completitud, ya que analizan los casos de *links* faltantes. Desarrollan un algoritmo automático de recuperación de *links*, y analizan el impacto en métricas del mantenimiento de software y de predicción de defectos, resultando en una mejora significativa. La calidad de datos se menciona como parte de la motivación del trabajo, pero no como el marco conceptual base.

Por otra parte en [28] se establece que la desviación (*bias*) en los datos podría afectar la validez estadística de las hipótesis que se basan en dichos datos. Por un lado, no todos los defectos son reportados en la herramienta (foco en la completitud). Por otro lado, la mayoría de los cambios no incluyen la referencia hacia el reporte de *bug* correspondiente (62,9%).

En particular, algunas publicaciones analizan la calidad de los datos sobre los cuales se basan para obtener predicciones o estimaciones, con foco principalmente en los análisis estadísticos sobre los datos [78]. La información histórica sobre los reportes de *bugs*, son la base para obtener modelos estadísticos basados en predicciones así como para probar/refutar hipótesis acerca de la efectividad del proceso. Se busca entonces conocer el efecto que tiene las "desviaciones" (*bias*) en los datos con respecto a los valores reales, sobre los resultados obtenidos (corresponde a un problema de correctitud). Se reconoce que es un tema crítico que debe ser tratado, ya que afecta la generalización de las hipótesis que se basan en dichos datos.

De la misma manera, otros estudios analizan la calidad de los datos de proyectos de software a partir de los cuales se obtienen predicciones de costos y esfuerzo. Mientras que algunos estudian específicamente los valores faltantes y la completitud de los datos [63], [64], otros se enfocan en analizar el "ruido" (o correctitud) [65]–[67], [73], o en la identificación y eliminación de *outliers* [61].

De los trabajos que se enfocan en el problema de "*noisy data*", algunos utilizan los datos como base para la predicción de defectos [73], y otros para aprendizaje (*machine learners*) [65]–[67] El foco está en estudiar el impacto que tienen los datos con "ruido" en las predicciones resultantes. La correctitud de la predicción de defectos depende fuertemente de la calidad de los datos. Si los datos de defectos son de mala calidad (con "ruido"), la validez de las predicciones obtenidas es cuestionable. Se propone un algoritmo para identificar y eliminar el ruido, estableciendo que existe una mejora en la exactitud de la predicción luego de eliminado el mismo. Se concluye que es una temática muy relevante, y que a pesar de que se han desarrollados varios modelos para la predicción de defectos, ninguno de ellos toma en cuenta el problema de los datos con ruido [73].

Mientras que la comunidad en ingeniería de software ha querido mejorar la precisión de las estimaciones de esfuerzo, no se ha considerado la calidad de los datos que impactan en los resultados obtenidos. En [61] se analizan diferentes métodos de eliminación de *outliers*. La motivación está en que los *outliers* influyen en la correctitud de las estimaciones de esfuerzo de software. Los resultados muestran que la aplicación de métodos de eliminación de *outliers* mejora las estimaciones de esfuerzo.

En algunos casos se muestra la aplicación de métricas de calidad de datos específicas, pero sin basarse en la disciplina de calidad de datos [23], [74]. Las métricas se definen sobre la calidad de los datos que analiza y sobre el proceso en sí mismo. Todas las métricas son definidas como ratios y promedios. A partir de la aplicación de las métricas y los resultados obtenidos, se muestra por qué los profesionales en ingeniería de software deben tener en cuenta la calidad de los datos y del proceso, motivando a mejorar en ese sentido. También muestra la relación que existe entre la calidad en el proceso (incluyendo la calidad de los datos) y en el producto. Estas publicaciones forman parte de la tesis doctoral de *Bachamn* [3], que será tratada en detalle más adelante.

En particular, hay estudios sobre la calidad de los datos específicamente en reportes de *bugs* [76], [77], [81], y en reportes de cambios [24]. En [81] el objetivo está en obtener la mayor cantidad de datos de la realidad sobre cada *bug*, siendo el análisis y limpieza de datos realizado de forma manual. Esto se debe a la dependencia que existe entre los datos de reportes de *bugs* y el conocimiento social, organizacional y técnico de quienes los reportan, que no puede extraerse solamente mediante repositorios de datos y herramientas automatizadas. La metodología propuesta consiste en, a partir de una base de datos que contiene reportes de *bugs*, intentar obtener la mayor cantidad de información electrónica posible. Mediante entrevistas a las personas relevantes que han tenido contacto o relación con el bug, se obtiene la información deseada o con quién se podría obtener, y así se continúa la cadena de información. El proceso finaliza una vez que se haya logrado la total reconstrucción de la información, o cuando ya no es posible obtener más. Se detallan diferentes tipos de errores de calidad de datos, concluyendo que los datos contenidos en los repositorios de reportes de *bugs* son incorrectos e incompletos (esto es, de mala calidad).

En [76] el análisis de la calidad de reportes de *bugs* se realiza desde el punto de vista del desarrollador, que es diferente al de los usuarios. Se mencionan cuáles son los principales errores que contienen los datos de los reportes de *bugs* (completitud, correctitud, duplicación), y se propone una herramienta que mide la calidad de los reportes, con el objetivo de mejorarla mediante recomendaciones propuestas.

En [77] se introducen técnicas de minería de datos para mejorar la calidad de los datos sobre reportes de *bugs*. Se enfoca en los formularios de descripción de texto libre, utilizando técnicas de PLN.

Respecto al análisis de la calidad de datos de reportes de cambios, en [24] se analizan los registros de cambios para evaluar la completitud y correctitud de los datos. La motivación para este análisis es que cada día estos datos son más utilizados como base para la experimentación, cuestionando de esta manera la utilidad de los datos para la investigación. Los problemas de completitud en los datos presentan porcentajes de omisión que van entre los 3,7 y 78,6%. Se concluye que previo a la ejecución de cualquier experimento basado en estos datos, los mismos deberían ser completos y correctos (esto es, de buena calidad).

Otras publicaciones que ponen foco en la calidad de datos destacan su importancia mediante la evaluación y aplicación de técnicas para la limpieza de datos [27], [78]. Estas publicaciones forman parte de la tesis doctoral de *Liebchen* [2], detallada en la próxima sección.

En [60] se desea analizar datos de proyectos para obtener métricas de productividad, pero como paso previo se hace una limpieza (exclusión de datos sucios) de los errores de calidad más visibles que puedan afectar los resultados. Se definen errores de calidad particulares que son tenidos en cuenta, y que se corresponderían con las dimensiones de completitud, correctitud, consistencia y unicidad. Los repositorios contienen varios valores en cero y otros que parecen corresponder al ingreso de datos incorrectos. Estos valores quedan por fuera de los análisis que se realizan con los datos, ya que afectarán el resultado obtenido.

Por otra parte, en [27] se comparan tres técnicas de limpieza de datos: ignorar los datos erróneos, eliminarlos, o corregirlos. El objetivo es reportar el impacto que tiene el "ruido" en los datos sobre la correctitud de las predicciones. Se desea evaluar la calidad de los datos antes de que sean utilizados para análisis, y de ser posible eliminar aquellos datos que se consideren sospechosos o de baja calidad. Para ello se realiza el análisis manual de los datos buscando errores de calidad (*noise*). Estas imperfecciones son un problema para los investigadores que utilizan datos de la realidad ya que pueden tener impactos no deseados en los análisis y conclusiones obtenidas.

4.2 Calidad de Datos aplicada a Ingeniería de Software Empírica

En esta sección se muestra de qué manera se ha aplicado la Calidad de Datos a diferentes repositorios que contienen datos resultantes de los Procesos de Ingeniería de Software, y que serán utilizados luego como base para obtener análisis y conclusiones.

A pesar de que estos trabajos tienen foco en los datos utilizados para la Ingeniería de Software Empírica, no se encuentra en la revisión de la literatura el análisis de la calidad de los datos resultantes, en particular, de experimentos en Ingeniería de Software.

4.2.1 Calidad en experimentos en Ingeniería de Software

Existen algunas propuestas que evalúan la calidad de experimentos en ingeniería de software, aunque el foco no es específicamente en la calidad de sus datos [57]–[59].

El objetivo en [57], [58] es evaluar la calidad de experimentos en Ingeniería de Software "centrados en humanos", de manera de identificar si la misma ha mejorado con el tiempo. Para ello se analizan 70 artículos mediante la utilización de un cuestionario (*checklist*) que contiene 9 preguntas, de las cuales 2 consideran los siguientes aspectos que refieren a la calidad de los datos:

- ¿Se utilizan métodos de control de calidad para asegurar la consistencia, completitud y correctitud de los datos recolectados?
- Si los outliers son mencionados y excluidos del análisis, ¿se hace de forma justificada?

Sin embargo, no especifica cómo se debe controlar cada ítem de la checklist, por lo tanto no se conoce cómo se determinan estos aspectos de la calidad de los datos. Establece además que en artículos relacionados no encuentran discusión sobre el uso de criterios para determinar la calidad de los estudios que se utilizan para meta-análisis.

Por otra parte, en [59] el objetivo es analizar si existe una correlación entre la validez interna y "*bias*" en experimentos en ingeniería de software. Mientras que la validez interna mide qué tan bien fue planificado, ejecutado y analizado el experimento, el "*bias*" mide qué tanto se alejan los resultados obtenidos de los resultados "reales". Corresponde a un error sistemático que representa la desviación del resultado del experimento con respecto a su valor verdadero.

4.2.2 Principales trabajos en Calidad de Datos para Ingeniería de Software Empírica

Los principales trabajos identificados sobre Calidad de Datos aplicada a Ingeniería de Software Empírica son las tesis de doctorado de *Liebchen* [2], [37] y de *Bachman* [3], y sus publicaciones relacionadas [1], [14], [15], [23], [27], [28], [60], [74], así como los artículos publicados recientemente por *Bosu* y *MacDonell* [25], [26].

En su tesis de doctorado *Liebchen* [2] propone la aplicación y análisis de tres técnicas de limpieza para repositorios de datos de ingeniería de software (r*obust filtering, predictive filtering, filte-*

ring and polishing), con el fin de conocer su efectividad y eliminar el "ruido" en los datos. Entre un 60 y 95% del esfuerzo dedicado a análisis de los datos se invierte en la limpieza de los mismos [37]. Su objetivo es analizar la calidad de los datos (en particular el "ruido") en repositorios de ingeniería de software empírica, dejando por fuera los *outliers*.

La falta de trabajos que investiguen de manera independiente la evaluación de la calidad de los datos en ingeniería de software empírica es otra de sus motivaciones. En vez de proponer nuevos algoritmos para calidad de datos en ingeniería de software empírica, se debería verificar si los ya existentes son apropiados. Por este motivo, realiza una revisión sistemática de la literatura con el objetivo de mostrar de qué manera la comunidad en ingeniería de software empírica ha considerado la calidad de los datos que utilizan (poniendo foco en los datos con ruido).

Por otra parte, la mayor contribución de la tesis de *Bachman* [3] es la evaluación de la calidad de los datos y las características de un conjunto de repositorios de datos usados frecuentemente en ingeniería de software empírica, analizando el efecto que tienen los problemas de calidad de datos en los profesionales e investigadores en ingeniería de software empírica.

Para ello, *Bachman* [3] presenta un procedimiento que permite recolectar, convertir e integrar los datos de los procesos de ingeniería de software, con el fin de mejorar su calidad. Define un *framework* de calidad de datos que incluye métricas de calidad de datos y características específicas para procesos de ingeniería de software. Este *framework* se aplica sobre datos de proyectos de software con el fin de evaluar su calidad. Como resultado se encuentra que todos los proyectos analizados contienen problemas de calidad de datos. En particular, el ratio de vinculación entre reportes de *bugs* y sus *commits* es en todos los casos menor al 55%, y en el peor caso de 7,43%. Esto constituye una amenaza sobre los resultados de las investigaciones que se basan en estos datos.

Algunas de las métricas de calidad de datos (en ratios) que se definen son: reportes de *bugs* corregidos, reportes de bugs duplicados, reportes de *bugs* inválidos, mensajes de *commit* vacíos, mensajes de *commit* con vínculo a reporte de *bug*, reportes de *bugs* con vínculo, sobre el total de reportes y sobre el total de reportes corregidos. Algunas de las características definidas (como promedios) son: cantidad de cambios de estado por reporte de bug, cantidad de comentarios por reporte de *bug*, cantidad de reportes de *bug* por usuario, largo de mensajes de *commit*, cantidad de usuarios que reportan bugs por desarrollador, cantidad de *bugs* (reportados y corregidos) por desarrollador, cantidad de *commits* por desarrollador.

También analiza otros *frameworks*, estándares y guías (CMMI, ITIL, COBIT, ISO/IEC, entre otros), pero ninguno de ellos contiene requerimientos específicos para calidad de datos en ingeniería de software.

Bosu y *MacDonell* [25], [26] están actualmente trabajando en la identificación de mecanismos que permitan mejorar la confianza asociada con los datos que son utilizados en Ingeniería de Software Empírica. Se basan en conceptos propuestos en el dominio de *data provenance* con el objetivo de lograr confianza y calidad en los datos utilizados. El foco de su trabajo está en la utilización de modelos de predicción en ingeniería de software empírica. Estos modelos dependen fuertemente de la calidad de los datos que utilizan. Por este motivo, es importante considerar algunos desafíos asociados a esta problemática, tales como el ruido, incompletitud, *outliers* y duplicados.

En particular, se propone una taxonomía de calidad de datos que cubre diferentes aspectos de la calidad de datos [26]. Mediante una revisión de la literatura, se analizan 57 artículos de Ingeniería de Software Empírica y se agrupan en las clases propuestas por la taxonomía. Se identifican un total de 74 problemas de calidad que son considerados en estos trabajos. La taxonomía puede ser utilizada por investigadores y profesionales. Pretende capturar la atención de la comunidad en ingeniería de software empírica, mejorar el entendimiento sobre los problemas de calidad de datos que pueden

existir en los repositorios de datos utilizados en Ingeniería de Software Empírica, y propone técnicas para solucionarlos.

Las categorías que componen la taxonomía propuesta y el porcentaje en que se encuentran presentes en los artículos analizados son los siguientes:

- Exactitud (65%): correctitud de los datos o ausencia de ruido.
- Relevancia (23%): tener y usar los datos apropiados para desarrollar un modelo.
- Origen (12%): limitan la accesibilidad y confianza de los datos, relacionado con la replicación experimental.

Por último, se analiza un conjunto de trabajos que evalúan la calidad de repositorios de datos de la NASA.

En [80] se propone un método de pre-procesamiento de datos con el objetivo de evaluar y mejorar la calidad de los repositorios de datos del programa de métricas, que son utilizados como base para experimentos en predicción de defectos de software. Mientras que los experimentadores asumen que estos repositorios contienen datos de suficiente calidad, existen problemas de calidad que deben ser atacados mediante un pre-procesamiento de los datos. El objetivo es motivar a los investigadores para que consideren seriamente la problemática de la calidad de datos, y cuestionen los resultados obtenidos a partir de estos datos. Luego del proceso de limpieza, cada uno de los 13 repositorios de datos analizados contienen entre 6 y 90% menos de datos.

En [79] se definen diferentes tipos de problemas de calidad de datos que pueden ocurrir en estos repositorios, sin especificar cómo se encuentran y miden. Los problemas podrían asociarse con dimensiones de calidad de datos, pero no hacen referencia a la disciplina.

Como forma de limpieza de estos problemas, se eliminan las instancias con errores: valores "improbables" o "conflictivos" (por ejemplo violación de restricciones de integridad referencial), instancias idénticas o inconsistentes, valores faltantes.

4.2.3 Importancia de la Calidad de Datos en el contexto de la Ingeniería de Software Empírica

En la Ingeniería de software empírica el insumo más importante son los datos, ya que son utilizados para predicciones, descubrimientos y decisiones sobre nuevas estrategias. También son utilizados para conocer si determinadas técnicas o estrategias están siendo efectivas, o qué impacto tienen. Teniendo esto en mente, llama la atención que no se analice la calidad de los datos que son utilizados en esta área [2].

El software construido es cada día más complejo y más grande, y por lo tanto su calidad es cada vez más difícil de evaluar. Acompañando esta evolución, la ingeniería de software también ha cobrado mayor importancia en los últimos tiempos. Una gran cantidad de información valiosa sobre la historia y evolución de los proyectos de software se encuentra disponible en herramientas de ingeniería de software (para gestión de bugs, configuración, control de versiones, entre otras). Muchos investigadores utilizan estos datos como información histórica y fundamental para hacer análisis y obtener conclusiones. A modo de ejemplo, para predecir la cantidad de bugs de un proyecto futuro, o hacer estimaciones de costo o esfuerzo. De la misma manera, la ingeniería de software empírica también utiliza datos recolectados para obtener conclusiones y hacer propuestas [3].

Sin embargo, las técnicas de recolección y procesamiento de datos son inexactas, por lo que los datos primarios podrían contener problemas de calidad. Esto hace que la calidad de los datos almacenados en repositorios de ingeniería de software resulten cuestionables, y por lo tanto impacten en los resultados de las investigaciones basadas en esos datos [3].

En los últimos años la ingeniería de software se ha ido desplazando hacia una posición más empírica, para ser una disciplina basada en la evidencia. Sin embargo, la problemática de la calidad de los datos no ha recibido la misma atención [15]. De esta forma, se pueden tomar decisiones en base a datos cuya calidad es incierta, y por lo cual pueden no ser correctas.

En otras áreas de investigación tales como sistemas de información y minería de datos el impacto de la mala calidad de los datos ha sido reconocido como una problemática que necesita ser atendida tanto por los productores como consumidores de datos. Esta problemática impacta en todos los segmentos de la economía: compañías, gobiernos, universidades, cliente [16], así como en la efectividad de las organizaciones [17].

De la misma manera, la mala calidad de los datos probablemente tendrá también un impacto severo en los datos utilizados para ingeniería de software empírica [2], [48]. Entonces, así como la calidad de datos ha sido considerada como de suma relevancia en otras áreas, ¿no debería ser también considerado un tema central para la ingeniería de software empírica? [3] Si los resultados que se producen son de baja calidad, entonces los profesionales de la ingeniería de software ignorarán los resultados que la comunidad científica les provee, ya que constituyen respuestas sin bases fundamentadas.

La importancia de que los datos en los cuales se basan los estudios empíricos gocen de buena calidad ha comenzado a ser reconocida y estudiada principalmente en los últimos tiempos [1]–[3], [14], [15], [23]–[26], ya que podrían impactar en las decisiones que se toman a partir de los resultados obtenidos. A pesar de que son pocos los trabajos que analizan la calidad de los datos que utilizan (sólo un 1%), su importancia se ha ido incrementando a lo largo de los años [2]. Esto lo demuestra la cantidad de estudios y publicaciones que existen en esta área por cada año.

No es menor destacar que la popularización y crecimiento de los repositorios de datos públicos, es un fenómeno que influye de manera significativa en la problemática de la calidad de datos. Existen grandes cantidades de datos que están disponibles para la comunidad científica y serán utilizados como base para realizar experimentos, sin siquiera llegar a conocer o estimar su calidad (PROMISE, NASA, ISBSG, entre otros).

Liebchen [15] concluye, a partir del análisis de los datos, que los repositorios contienen una gran cantidad de ceros y valores faltantes, así como valores incorrectos y *outliers*, considerando que la calidad de los datos es un aspecto fundamental para la ingeniería de software empírica.

Bachman [3], en el mismo sentido, plantea por qué los investigadores (como consumidores de datos) y profesionales (como productores de datos) en ingeniería de software deberían considerar la calidad de los datos que utilizan y producen respectivamente. La motivación está en que las investigaciones (predicciones de defectos y costo, análisis de procesos, análisis estadísticos) solo serán posibles y válidas si los datos en los cuales se basan gozan de buena calidad.

Por un lado, los investigadores en Ingeniería de Software Empírica utilizan datos generados a partir de procesos de ingeniería de software para sus investigaciones. Si estos datos no son de buena calidad, entonces puede afectar los resultados obtenidos. Es por esto que los investigadores deberían preocuparse por esta problemática, y reportar las posibles amenazas que implica realizar trabajos con esos datos.

Por otra parte, la manera más sencilla y eficaz de lograr una buena calidad en los datos es asegurarla desde su producción (fuente), esto es, por los profesionales (ingenieros, líderes de proyectos y testeo de software, entre otros) y los usuarios de herramientas en ingeniería de software. Corregir los errores a posteriori, es sin duda más costoso que prevenir su ocurrencia. Muchos de los problemas de calidad pueden ser resueltos por quienes producen dichos datos. Los profesionales deberían preocuparse y ser conscientes de la calidad de los datos que generan, invirtiendo más tiempo y esfuerzo

para asegurar la calidad de los datos que producen, ya que son ellos uno de los principales beneficiados en que los resultados obtenidos reflejen la realidad.

Concluye que la mala calidad de datos se debe principalmente a la falta de una correcta integración y a la ausencia de valores en las herramientas en donde se almacenan los datos. Los datos recolectados por estas herramientas pueden no ser de buena calidad, y sin embargo es a partir de estos que se obtienen los resultados. Los datos derivados a partir de procesos de ingeniería de software contienen problemas de calidad, y estos afectan los resultados empíricos que se obtienen a partir de sus datos. De esta manera, existe una gran incertidumbre sobre las asunciones que se realizan a partir de investigaciones empíricas [3].

La importancia de la calidad de datos también se hace presente en el contexto de los datos recolectados por Procesos de Desarrollo de Software. En particular, en el Personal Software Process (PSP) los datos recolectados son utilizados para obtener conclusiones acerca de la efectividad del proceso. Esto se puede considerar también como un caso de aplicación de ingeniería de software empírica. Las publicaciones que existen en este contexto [69]–[71], que surgen hace más de 15 años, demuestran que la preocupación por conocer y mejorar la calidad de los datos que se recolectan está aún presente, debido a su posible impacto en las conclusiones que se obtienen respecto a la efectividad del proceso.

Se resalta la importancia que tiene la calidad de datos para que los análisis que se obtengan a partir de los mismos sean reales y fiables. Se concluye que mientras que no se analice y mejore la calidad de los datos, estos no deberían ser utilizados para evaluar el proceso en sí mismo [70].

4.2.4 Revisiones de la literatura existentes

Se encuentran tres trabajos de gran relevancia en lo que refiere a la revisión de la literatura de Calidad de Datos para Ingeniería de Software Empírica.

Por una parte, *Liebchen* y *Shepperd* [1], [2], [14] realizan un trabajo sumamente interesante al analizar mediante una revisión sistemática de la literatura si las publicaciones en el área de Ingeniería de Software Empírica (al año 2010) hacen mención explícita de la calidad de datos o del "ruido".

Una revisión sistemática consiste en un proceso que requiere la exploración exhaustiva de toda la literatura disponible que cumpla con el criterio especificado [83]. Se considera que los resultados obtenidos a partir de la revisión sistemática proveen una vista adecuada del estado de eventos en la comunidad de ingeniería de software empírica.

El principal objetivo es identificar los estudios relevantes en este sentido, y proveer un contexto sobre el tratamiento que la comunidad en ingeniería de software empírica le ha dado a la temática de calidad de datos.

El criterio utilizado para la selección de artículos es el siguiente: con foco en una investigación empírica en algún aspecto de la ingeniería de software, o que aborde un tema metodológico relevante para dicha investigación; menciona explícitamente datos con "ruido"; es referenciado; está escrito en inglés.

Constituye un interesante punto de partida para conocer qué trabajos existen en el área de la ingeniería de software empírica que traten la problemática de la calidad de los datos. Sin embargo, el foco del trabajo está en investigar específicamente los datos con "ruido" (*noisy data*) en repositorios de ingeniería de software, resultando un alcance menos abarcativo que el propuesto en este trabajo. A pesar de que no existe una definición consensuada sobre el "ruido" (*noise*), lo define como "información incorrecta, falta de información o información no confiable" [2]. Esta definición toma en cuenta sólo las facetas de exactitud y completitud, y sin distinguir entre ambas. Establece que mientras

los datos faltantes pueden ser más fácilmente identificados, los datos con ruido son más difíciles de encontrar [48].

Diferencia también los términos "*noise*" y "*outlier*". Mientras que "*noise*" corresponde a instancias no deseadas que contienen valores incorrectos, los "*outliers*" corresponden a instancias con valores excepcionales en comparación al resto de la muestra.

Shepperd [1] también define la buena calidad de datos como la ausencia del ruido, esto es, cuando el valor registrado para un dato es el mismo que el valor real. La mayor dificultad está en conocer cuál es el valor real.

A continuación se presenta un resumen de los principales resultados obtenidos a partir de la revisión sistemática [1], [2], [14]:

- Solamente un 1% (161 en 17000) de todos los artículos publicados en el dominio de ingeniería de software empírica se consideran relevantes, haciendo mención explícita a la calidad de datos o "ruido", sin necesariamente proponer acciones para atacar esta problemática.
  Esto demuestra que el tema de calidad de datos está siendo desatendido en un dominio de aplicación donde el insumo principal son los datos. Sin embargo, se observa también un incremento en el tiempo que sugiere que la comunidad le está comenzando a prestar mayor atención.
  La calidad inadecuada de los datos amenaza la validez de las conclusiones obtenidos a partir de sus análisis. Estas conclusiones resultan cuestionables si se basan en datos de pobre calidad.
- El 86% considera que la calidad de los datos constituye una amenaza para el análisis de los datos empíricos. Sin embargo, poco trabajo se ha realizado para combatirla.
- Los artículos cubren diferentes tópicos dentro de la ingeniería de software empírica. Los dominios de aplicación más predominantes son predicción de costos y esfuerzo, y calidad de software.
- La mayoría de los artículos (76%) se enfocan en la calidad de datos desde una perspectiva cuantitativa, mientras que un 28% se enfocan en la perspectiva cualitativa, y un 11% se interesa en ambas (cuantitativa y cualitativa).
- El 31% de los artículos seleccionados plantean la mejora en los procedimientos de recolección de datos como la técnica más predominante para tratar la problemática de mala calidad. Esto constituye una actividad de prevención de calidad. Una sugerencia frecuente es la automatización del proceso de recolección de datos y validación de entrada (tales como chequeo de rangos) de forma de evitar errores en la imputación de datos, por ejemplo mediante la incorporación de una herramienta.
  Sin embargo, considerando que los analistas en el dominio de la ingeniería de software empírica tienen que trabajar con datos históricos y en muchos casos no tienen injerencia en el proceso de recolección de los datos, los investigadores deberían aplicar procedimientos de limpieza de datos a posteriori.
- El enfoque práctico más dominante es el chequeo manual de la calidad de datos (22% de los artículos).
- El 11% de los artículos utilizan metadatos para identificar y eliminar las instancias con niveles inadecuados de calidad de los repositorios de datos, y de esta manera mejorar la calidad de los datos. En general estos metadatos son utilizados solamente para sustituir los valores faltantes (completitud).
- El 10% de los artículos emplean técnicas automáticas para la detección y/o limpieza de datos con errores, y solo el 13% llevan a cabo un análisis empírico en la detección del ruido en

repositorios en ingeniería de software. El chequeo automático del "ruido" es llevado a cabo principalmente mediante la utilización de algoritmos de máquinas de aprendizaje (*machine learning algorithm*).

- Solo 21 artículos sobre ingeniería de software empírica satisfacen los criterios de inclusión y referencian explícitamente la calidad de datos, tomando acciones al respecto. Sin embargo, se percibe un incremento en el tiempo, sugiriendo que la comunidad le está dando mayor relevancia y atención a la calidad de datos. El 73% de los artículos consideran que es una temática de suma importancia.

En líneas generales, se encuentran pocos trabajos que analizan la calidad de datos de un repositorio dado. El chequeo y tratamiento automático del "ruido" se encuentra todavía en una fase temprana, y la efectividad de los métodos propuestos para la identificación y manipulación del ruido no ha sido aún probada.

La mala calidad de los datos puede amenazar la validez de las conclusiones de los estudios empíricos. Es por esto que resulta sorprenderte que solo una pequeña porción de los autores reporten de manera explícita problemas en la calidad de los datos. La falta de información sobre la calidad de sus datos puede impactar negativamente en la interpretación de sus resultados, así como en su replicación. Es un hecho conocido que el "ruido" está presente en los datos bajo estudio. Sin embargo, su tratamiento constituye un gran desafío y en general no existen soluciones sencillas de aplicar. El real valor de los datos es realmente desconocido, mientras no se pueda determinar de manera precisa el nivel de ruido que contienen [1].

Las investigaciones son fundadas en datos, y en general se asume que los problemas de calidad que esos datos puedan tener no afectan los resultados. Sin embargo, los resultados de la revisión sistemática muestran que solo una minoría de los investigadores toma en cuenta la calidad de los datos. Se sugiere que esta temática debe ser considerada de gran prioridad entre los investigadores de ingeniería de software empírica [1].

Bachman [3] realiza también una revisión de la literatura buscando dos objetivos. Por un lado, conocer si existen problemas de calidad en los repositorios de datos de ingeniería de software. De las publicaciones que cubren este objetivo, destaca el trabajo realizado por *Aranda y Venolia* [81] por ser el único que intenta verificar la completitud y el grado de validez de repositorios en ingeniería de software, explorando posibles efectos en los resultados de las investigaciones. Esta es la única publicación que responde a la pregunta sobre la existencia de evidencia de "bias" sistemática en la relación entre los reportes de *bugs* y los cambios que los corrigen [28]. Aunque no menciona que se trate de un estudio de calidad de datos, concluye que los datos contenidos en los repositorios de reportes de *bugs* son incorrectos e incompletos, analizando diferentes problemas que encuentran respecto a la calidad de los datos.

El segundo objetivo consiste en conocer si dichos problemas impactan en los resultados de las investigaciones, para lo cual no se encuentran evidencias.

Finalmente, *Bosu* y *MacDonell* [25] realizan una revisión enfocada de la literatura (*targeted literature review*) con el objetivo de identificar evidencia sobre tres elementos potencialmente influyentes en la calidad de los datos de estudios en Ingeniería de Software Empírica. Los estudios se clasifican según: si se reportan los procedimientos de recolección de datos, si los datos fueron pre-procesados, y si se analiza la calidad antes de su utilización para los modelos. Esto permite conocer cómo los investigadores tratan esta problemática y qué mecanismos se utilizan para tratarla.

La revisión cubre estudios de ingeniería software empírica en el período desde Enero 2007 hasta Setiembre 2012, y se restringe la búsqueda a determinadas conferencias. Se consideran los siguientes criterios de inclusión:

- Estudios diseñados para estimar, predecir o modelar algún aspecto de los fenómenos en ingeniería de software, tales como estimación de costo/esfuerzo o predicción de defectos.
- Estudios que introducen mediciones.
- Estudios que analizan o evalúan algún aspecto los repositorios de Ingeniería de Software Empírica o de la calidad de sus datos.

Los resultados obtenidos a partir de la revisión serán de utilidad tanto para que los investigadores conozcan el estado del arte e identifiquen debilidades en sus próximas investigaciones, como para que los profesionales conozcan y prevengan la ocurrencia de problemas de calidad de datos.

Se identifican un total de 221 estudios relevantes que cumplen con el criterio de inclusión definido, y se categorizan en términos de la consideración y tratamiento de la calidad de los datos. Como resultado, se obtiene un entendimiento sobre cómo la comunidad en ingeniería de software empírica considera los tres elementos de calidad de datos propuestos en este estudio.

Se presenta a continuación un resumen de los resultados obtenidos:

- Solo 23 de los 221 (10% de los estudios revisados) reportan en los tres elementos considerados para la calidad de datos. Cuanto más se trata el pre-procesamiento, más problemas de calidad son identificados. Sin embargo, no existe una tendencia identificable entre la recolección de datos y los otros dos elementos analizados.
- El 57% no reportan sobre la calidad de los datos que utilizan, sino que los datos son utilizados tal como fueron obtenidos. El 43% mencionan posibles problemas de calidad de datos, destacando los desafíos que se encuentran en este sentido y en algunos casos describiendo cómo resolverlos. Esto confirma el planteo de *Liebchen* [14], que establece que sólo una minoría considera la calidad de datos en la comunidad de Ingeniería de Software Empírica.
- Los problemas de calidad encontrados corresponden a: 42% incompletitudes, 33% outliers, 13% noise (correctitud), 8% metadatos, 3% inconsistencias, 1% redundancias.
- A pesar de que realiza una clasificación de los problemas de calidad de datos encontrados, no se hace referencia a la disciplina de calidad de datos.
- En el 63% de los casos hay pre-procesamiento de los datos antes de su uso ("limpiezas" o correcciones ad-hoc que se hacen sobre los datos previo a su análisis). Sin embargo, la forma en la que esto se lleva a cabo varía ampliamente, indicando que no existe un proceso a seguir o un enfoque que aporte valor en este sentido.
- El 43% reportan sobre los procedimientos de recolección de datos, aunque el alcance de la descripción es muy variada. En la mayoría de los casos, no se describen los problemas de calidad asociados a esta actividad, lo cual dificulta su identificación.
- Se concluye que los procedimientos de recolección de datos no son reportados de manera consistente en los estudios de ingeniería software empírica, y que la comunidad debería prestar más atención a la calidad de datos. La mejora de la recolección, pre-procesamiento y evaluación de calidad de los datos resultarán en modelos de predicción más confiables.

A pesar de que las tres revisiones se realizan para el mismo contexto (Calidad de Datos para Ingeniería de Software Empírica) y persiguiendo objetivos muy similares, el foco específico de cada revisión no es exactamente el mismo. De todas formas los tres trabajos concluyen que, a pesar de que hay trabajos que reconocen su importancia, son pocos los casos en los que realmente se considera de manera explícita la problemática de calidad de datos, y se plantean acciones en este sentido. Los investigadores en ingeniería de software hacen uso de los datos sin analizar los posibles efectos que podrían tener los problemas de calidad en los resultados obtenidos [2], [3].

Vale destacar que los dos últimos trabajos [3], [25] se apoyan y hacen referencia a la revisión sistemática de la literatura realizada por *Liebchen* [14]. Sin embargo, no se comparan los resultados

obtenidos entre estas revisiones. Mientras que *Liebchen* [14] identifica 21 artículos sobre ingeniería de software empírica que satisfacen los criterios de inclusión y referencian explícitamente la calidad de datos, *Bosu* [25] identifica 23 estudios que reportan sobre alguno de los elementos de calidad de datos considerados en su estudio. No se conoce si estos resultados corresponden o no a las mismas publicaciones (todas o alguna de ellas), y no forma parte del alcance de este trabajo hacer este análisis.

Se concluye que la comunidad de Ingeniería de Software Empírica debería prestarle más atención a la temática de calidad de datos. Los resultados muestran que se ha negado y no se le ha dado la suficiente importancia a analizar y mejorar la calidad de los datos que utilizan. Entonces, ¿cómo es posible que los investigadores confíen en los resultados que obtienen a partir de sus experimentos? [14] ¿Cómo pueden los profesionales beneficiarse de estos resultados, si existe el riesgo de que los datos utilizados no sean confiables, y por lo tanto de que los resultados no reflejen la realidad?

4.2.5 Calidad de Datos aplicada a procesos de desarrollo de software

El principal trabajo en este contexto es un caso de estudio que analiza la calidad de datos recolectados manualmente (sin soporte de herramientas) por el *Personal Software Process* (PSP) [69]–[71]. El objetivo es evaluar la calidad de los datos de PSP, que son utilizados para conocer acerca de la efectividad del proceso.

El *Personal Software Process* (PSP) fue propuesto por Watts Humphrey en 1995. Su objetivo es incrementar la calidad de los productos manufacturados por individuos profesionales, mediante la mejora de sus métodos personales para el desarrollo de software [31]. Pone foco en la reducción de defectos y mejora en la correctitud de estimación, siendo estos los dos objetivos principales que hacen a la mejora del proceso personal.

Los datos son recolectados durante la ejecución de cursos de PSP, en los cuales se enseña progresivamente a los ingenieros algunas prácticas sobre planificación, desarrollo y evaluación del proceso mientras construyen sus programas. Los datos sobre su trabajo son recolectados y analizados por ellos mismos, y utilizan las métricas obtenidas para mejorar su comportamiento y productividad. Sin embargo, existen dudas sobre la calidad de los datos utilizados.

Se analizan los datos generados por 10 estudiantes durante la ejecución de 89 proyectos, y se identifican un total de 1539 errores. Esto representa un 4,8% del total de valores analizados. Para encontrar los errores de calidad, se comparan los resultados de los cálculos manuales realizados por los alumnos con cálculos automatizados, y se analizan las diferencias. Se plantea que existen tres formas mediante la cual se pueden detectan errores: por otro estudiante mediante un revisión técnica, por el instructor durante el proceso de evaluación, o mediante el uso de una herramienta. Sin embargo, esta propuesta no resulta generalizable para otros casos sino que es específico para este caso de estudio.

Se identifican "tipos de errores" de calidad de datos, y se clasifican según en la etapa en la que se introducen.

- Etapa de recolección: errores de omisión (el desarrollador no registra una medida primaria, se asocia con la completitud), errores de adición (el desarrollo incluye un dato que no refleja la realidad, se asocia con la correctitud), y errores de transcripción (equivocaciones durante la transcripción, se asocia con la correctitud).
- Etapa de análisis: errores de omisión (el desarrollador no realiza un análisis de datos primarios que es requerido, se asocia con la completitud), errores de cálculo (análisis incorrecto, se asocia con la correctitud), y errores de transcripción (equivocaciones al mover datos, se asocia con la correctitud).

La corrección parcial de estos errores llevó a encontrar una diferencia significativa en las métricas que evalúan la efectividad de PSP. Esto confirma el impacto de la mala calidad en los datos

analizados, de forma tal de que las conclusiones sobre la mejora del proceso pueden resultar incorrectas. Se establece además que los errores originados durante la etapa de análisis son más sencillos de encontrar y corregir que los originados durante la etapa de recolección.

Como conclusión del trabajo se establece que hasta no ser tratada y resuelta la problemática de calidad de datos para PSP, estos no deberían ser utilizados como base para evaluar y mejorar el propio método. También se establece que el PSP debería ser soportado por alguna herramienta que se integre en la mayor medida posible con el proceso, con el fin de prevenir la ocurrencia de ciertos errores de calidad. Años después se comienza a utilizar una herramienta en *Access* que da soporte al PSP, y sin embargo no se encontraron publicaciones posteriores que analicen la calidad de los datos registrados en herramientas.

A partir de estos trabajos surgen propuestas de herramientas con el objetivo de automatizar en la mayor medida posible el registro, seguimiento, clasificación, reportes, y consolidación de datos de PSP [84]. También se realiza un análisis de las diferentes herramientas que se utilizan para dar soporte al PSP, concluyendo que ninguna de ella cumple con todos los requerimientos especificados [85].

A pesar de que se proponen herramientas y se buscan alternativas para lograr mejorar la calidad de los datos, se debe tener presente que el hecho que exista una herramienta no evita la existencia de datos de mala calidad, sino que solo ayuda a prevenir la ocurrencia de ciertos tipos de errores. Más allá de la automatización de la etapa de análisis, la calidad de datos global de PSP dependerá de la calidad de datos en la etapa de recolección [69].

Finalmente, en [72] se presenta la replicación de un estudio empírico ya realizado en el SEI, con el fin de demostrar que el PSP mejora la "*performance*" individual de los ingenieros. Para ello, toma como referencia los tipos de errores definidos en [69], planteando las acciones que se toman para considerar los problemas de calidad sobre los datos a analizar. De esta manera, mejora la calidad de los datos a partir de los cuales se hacen análisis.

4.3 Posicionamiento de nuestra propuesta

La propuesta presentada en este trabajo se diferencia de los trabajos encontrados y analizados en los siguientes aspectos.

1. **Aplicamos conceptos, técnicas y herramientas propuestas por la disciplina de Calidad de Datos, con el objetivo de evaluar y mejorar la calidad de los datos experimentales en Ingeniería de Software.**

   A partir del análisis de los trabajos propuestos en el área, se observa que la mayoría no consideran o utilizan la calidad de datos como disciplina o área de investigación en sí misma.

   Como se muestra en la Tabla 2, sólo 2 de los trabajos analizados mencionan y/o aplican la Calidad de Datos como disciplina. Aparte de las tesis doctorales antes mencionadas y sus artículos relacionados [2], [3], se analizan otros 3 trabajos cuyo dominio de aplicación no es la Ingeniería de Software Empírica. Estos trabajos fueron analizados de forma de identificar si existe alguna propuesta similar a la de nuestro trabajo (aunque en otro dominio de aplicación) que pueda ser tomada como referencia.
2. **Consideramos la visión multifacética propuesta por la disciplina de Calidad de Datos, teniendo en cuenta las dimensiones y factores de calidad acordados y consensuados por los principales autores del área.**

   En la mayoría de los trabajos, la Calidad de Datos es analizada desde algún aspecto o punto de vista particular, tales como el "ruido" (relacionado con la exactitud) o valores faltantes

(relacionado con la completitud). Esto reduce la probabilidad de encontrar una mayor cantidad de problemas de calidad durante el análisis de los datos, ya que limita la evaluación a determinados aspectos de calidad. También se encuentra, aunque en menor proporción, que algunos trabajos consideran otras facetas de la calidad tales como la consistencia, unicidad y la búsqueda de outliers.

Como se observa en la Tabla 2, mientras que 10 de los trabajos analizados no consideran (explícita ni implícitamente) ninguna dimensión o factor de calidad de datos, 20 consideran 1 o 2 dimensiones, y los restantes 13 tienen en cuenta 3 o más. Notar que esto no significa que se haga mención explícita a las dimensiones como tal, sino que a partir de su análisis se infiere y asocia el trabajo realizado con alguna faceta de calidad conocida. En la mayoría de los casos, por otra parte, se considera las dimensiones de calidad de datos solo desde un aspecto particular (por ejemplo, el "ruido" asociado con la correctitud). En nuestro trabajo, sin embargo, se consideran todos los factores que se agrupan bajo un mismo propósito por cada dimensión de calidad en cuestión.

3. **Definimos métricas de Calidad de Datos como el instrumento que permite identificar los problemas de calidad que están presentes sobre los datos, y así poder tomar las acciones correctivas o preventivas que correspondan.**

   A pesar de que existen algunos ejemplos de definición de métricas y problemas de calidad entre los trabajos encontrados, ninguna se aplica específicamente a datos de experimentos en Ingeniería de Software.

   Sólo en el trabajo de *Bachman* [3] se define un *framework* de calidad de datos que incluye métricas y características específicas para evaluar la calidad de los datos de un conjunto de repositorios de datos usados frecuentemente en ingeniería de software empírica. Las métricas son definidas como ratios y se especifica como calcularlas. A pesar de que estas métricas podrían mapearse con diferentes factores y dimensiones de calidad (exactitud y completitud), no se basa en el marco conceptual provisto por el área de Calidad de Datos para su definición.

   Sin embargo, llama la atención que uno de los trabajos más importantes en esta área (como es el de *Liebchen* [15]) no se definen métricas para evaluar la calidad de los datos. Lo mismo sucede en el trabajo de *Bosu* [26], en el cual se propone una taxonomía de calidad de datos que cubre diferentes aspectos de la calidad de datos, sin definir las métricas posibles.

4. **Formalizamos y aplicamos un enfoque sistemático, disciplinado y estructurado de forma de identificar los problemas de calidad en los datos y corregirlos. Definimos una metodología de trabajo que puede ser replicable sobre datos de experimentos en Ingeniería de Software.**

   En la mayoría de los trabajos analizados no se menciona explícitamente cómo se identifican los datos que contienen errores de calidad o cómo limpiarlos. En general se utiliza un enfoque "ad-hoc" para analizar la calidad de los datos, que no es posible repetir en experiencias similares. No se identifican los objetos relevantes del dominio, de forma de focalizar el trabajo de calidad sobre los datos más importantes para los análisis.

   La aplicación de una metodología y enfoque ordenado como el propuesto en este trabajo, no solo maximiza la cantidad de problemas de calidad que se pueden identificar sobre los datos, sino que también permite que sea repetible en otros estudios de características similares.

   Así como en [57] se establece que no se encuentra en los trabajos analizados la utilización de un criterio común para evaluar la calidad de los estudios, podemos afirmar que tampoco encontramos trabajos que apliquen un enfoque o metodología común para evaluar la calidad de sus datos.

En el trabajo realizado por *Disney* [69]–[71] se encuentran un total de 1538 errores en los datos. Sin embargo, se establece que no se tiene la confianza suficiente como para asegurar que se hayan encontrados todos o la mayor parte de los errores de calidad que pueden presentarse en los datos. Mientras que la definición de una metodología de trabajo y un enfoque ordenado no puede asegurar el cubrimiento del 100% de los problemas de calidad, permite maximizar la probabilidad de que esto ocurra. Por otra parte, el caso presentado en [72], muestra que es posible definir un conjunto de "tipos de errores" (métricas o problemas de calidad) para un contexto dado (como es en este caso el PSP), que resulten replicables en otros casos de estudio.

Liebchen [2] encuentra como resultado de la revisión sistemática que ninguno de los artículos analizados utilizan protocolos de calidad de datos. Esto constituye una fuerte problemática ya que dificulta la replicación de los estudios. No se han encontrado evidencias sobre la aplicación de protocolos de calidad que describan los pasos del proceso que hayan sido seguidos para analizar y mejorar la calidad de los datos.

Liebchen plantea como una oportunidad de investigación a futuro el desarrollo de un protocolo de calidad de datos para la comunidad de ingeniería de software empírica. En particular, se deberían utilizar protocolos que describan el proceso de limpieza de los datos, asegurando la rigurosidad y replicación en futuras investigaciones en ingeniería de software empírica. El desarrollo de un protocolo unificado para la calidad de datos contribuiría en la identificación de las técnicas de limpieza a aplicar, así como de las instancias con "ruido" que podrían ser analizadas nuevamente por otros investigadores.

Este punto es particularmente de interés para nuestro trabajo, ya que uno de los principales objetivos y aportes consiste en la definición de un enfoque sistemático, disciplinado y estructurado para la evaluación de la calidad en datos recolectados en el contexto de la ingeniería de software empírica.

La Tabla 2 resume cómo se clasifican los trabajos analizados según los diferentes aspectos que se consideran respecto a la Calidad de los Datos.

Técnicas y actividades / *Comparación con nuestra propuesta*		**Análisis y/o evaluación**	**Medición**	**Limpieza**
Menciona y/o aplica DQ como disciplina?		[3], [42], [49], [52]		[1], [2], [14]
Dimensiones/ aspectos considerados	Ruido (*noise*), Valores inverosímiles, Exactitud/Correctitud	[3], [23], [24], [26], [56]–[58], [62], [65]–[67], [69]–[74], [76], [77], [81]		[1], [2], [14], [15], [27], [60], [70], [73], [79], [80]
	Valores faltantes, Completitud	[3], [23], [24], [26], [28], [56]–[58], [62], [69]–[72], [74], [76], [77], [81]		[1], [2], [14], [15], [27], [55], [60], [63], [64], [68], [70], [75], [79], [80], [82]
	Consistencia	[26], [53], [56]–[58], [62], [81]		[60], [79], [80]
	Duplicación, Contradicción	[3], [23], [26], [62], [74], [76], [81]		[60], [79], [80]
	Outliers	[3], [26], [57], [58], [62]		[1], [2], [14], [15], [27], [61]
	Bias	[3], [59], [78]		
Cantidad de dimensiones consideradas	No considera	[25], [42], [49], [52], [59], [61], [65]–[67], [78]		
	1 o 2 dimensiones	[1], [2], [14], [15], [24], [27], [28], [53], [55], [63], [64], [68]–[73], [75], [77], [82]		
	3 o más dimensiones	[3], [23], [26], [56]–[58], [60], [62], [74], [76], [79]–[81]		
Definición de problemas	Definición de problemas	[25], [26], [42], [52], [62], [69]–[71], [79]		[25], [79]
Definición de métricas	Definición de métricas	[3], [74]		
Definición de framework, taxonomía, metodología de trabajo	Definición de framework, taxonomía, metodología de trabajo	[3], [26], [42], [49], [52], [74]		[80]

Tabla 2: Clasificación de trabajos según aspectos de Calidad de Datos

Capítulo 5: Metodología de Investigación

En este capítulo se describe brevemente la metodología de investigación que se siguió para la construcción y el desarrollo del modelo de calidad de datos. Finalmente se presenta los roles participantes.

La metodología de investigación seguida está compuesta por dos iteraciones de investigación. La primera iteración consiste en la generación de un modelo genérico inicial de calidad de datos, y una metodología para la aplicación del modelo sobre casos específicos. La segunda iteración consiste en la validación y refinamiento del modelo generado, a partir de su aplicación sobre datos de experimentos en Ingeniería de Software .

5.1 Primera iteración: construcción del modelo de calidad de datos y metodología de aplicación

A continuación se presenta el proceso que se siguió para la construcción de la versión inicial del modelo de calidad de datos y la metodología para su aplicación, mostrando sus entradas, principales actividades y salidas.

Entradas

- Conjunto de dimensiones y factores de calidad propuestos por diferentes trabajos del área de Calidad de Datos.
- Necesidades y requerimientos de los experimentadores en ingeniería de software.
- Limitaciones y características particulares de los datos del dominio bajo estudio (experimentos en ingeniería de software ejecutados por sujetos humanos).

Actividades principales

1. Se analizan diferentes propuestas que existen en el área de calidad de datos sobre dimensiones y factores de calidad, descriptas en el Capítulo 2.
   Las dimensiones y factores de calidad considerados así como sus definiciones están basadas en los conceptos conocidos y más referenciados, que son propuestos y compartidos por los principales autores del área [10]–[12], [16], [19]. De la propuesta de dimensiones consensuadas, se consideran inicialmente las dimensiones *Exactitud, Completitud, Consistencia y Unicidad*, y sus respectivos factores de calidad. A esto llamamos dimensiones y factores base. No se consideran las dimensiones relacionadas con el tiempo y la vigencia de los datos, ya que no tiene aplicabilidad en este dominio de estudio. Esto se debe a que los datos de un experimento son el resultado de una ejecución particular en un momento dado, no requieren ser actualizados ni perderán vigencia.
2. Se analiza si las dimensiones seleccionadas son necesarias y suficientes para evaluar los datos en el dominio bajo estudio. Para el dominio de aplicación bajo estudio, se destacan las siguientes características a tener en cuenta:
   - Los datos son ingresados por parte de sujetos o experimentadores (humanos). Esto puede influir en la cantidad y tipos de errores que se cometan.
   - Se identifican datos que son significativos para los análisis estadísticos. Esto hace que se seleccionen o prioricen los datos a ser analizados así como las correcciones a ejecutar una vez identificados los problemas de calidad.
   - Los experimentos se ejecutan en un contexto académico. Los sujetos son en general estudiantes que están realizando algún curso, por lo cual la motivación o interés que tienen en registrar los datos puede variar.

   A su vez, los experimentadores también plantean sus necesidades y requerimientos con respecto a la calidad de los datos que recolectan y utilizan, y que son utilizados como input para la construcción del modelo.

3. Considerando todos estos elementos, se procede a identificar cuáles son los posibles problemas de calidad que podrían presentarse en los datos bajo estudio. Estos problemas se mapean con las dimensiones y factores base. Si no existen dimensiones o factores correspondientes se definen nuevos basados en la literatura base de referencia. Luego, se definen métricas de calidad para medir los factores. De esta manera se obtiene una versión inicial del modelo.
4. Una vez definido el modelo de calidad, y de forma de que el mismo pueda ser aplicado de manera sistemática sobre los datos de diferentes experimentos en ingeniería de software, se definen las fases que conforman la metodología de aplicación del modelo de calidad sobre casos (experimentos) particulares.

Salidas

La salida generada corresponde a los productos de investigación resultantes. Los mismos son descritos en detalle en el Capítulo 6.

- Modelo de calidad de datos inicial. El modelo de calidad definido contiene el el conjunto de dimensiones, factores y métricas de calidad de datos, así como la relación que existe entre estos, para ser aplicadas sobre el dominio específico de experimentos en ingeniería de software.
- Metodología para la aplicación del modelo. Esta metodología define cómo se aplica el modelo de calidad sobre los datos de experimentos en ingeniería de software.

5.2 Segunda iteración: validación y refinamiento del modelo de calidad de datos

Esta segunda iteración consiste en la validación, refinamiento y ajustes del modelo de calidad de datos, que surgen a partir de su aplicación sobre experimentos en ingeniería de software. La validación consiste en evaluar si el modelo de calidad definido resulta aplicable al dominio bajo estudio, y si los resultados obtenidos a partir de su aplicación son de valor para el experimentador y la comunidad. El refinamiento y ajustes son mejoras que pueden introducirse al modelo de calidad, de forma que pueda ser aplicado también sobre otros casos de forma beneficiosa.

La estrategia de investigación seguida para esta segunda iteración es la experimentación: se aplica el modelo de calidad de datos a casos específicos para validarlo y ajustarlo según corresponda. Para llevar a cabo esta segunda iteración se utilizó como herramienta el círculo de Deming (Ciclo PDCA: *Plan-Do-Check-Act*) [29] que define una estrategia para la mejora continua de la calidad en cuatro pasos como se muestra en la Ilustración 4. La metodología de investigación para el desarrollo y refinamiento del modelo de calidad de datos se describe a continuación.

Plan **(Planificar)**

En esta etapa se planifica la aplicación de la última versión generada del modelo de calidad de datos base, a los datos de un experimento en particular. Esto incluye las siguientes actividades: identificar el experimento, contactar con los responsables del experimento, agendar las reuniones de trabajo, establecer los objetivos y alcance del trabajo.

Do **(Hacer)**

En esta etapa se aplica la última versión generada del modelo de calidad de datos sobre el experimento seleccionado en la etapa anterior. La aplicación del modelo de calidad sobre los datos recolectados por el experimento se realiza utilizando la metodología definida (que será presentada en el Capítulo 6). De esta forma se obtiene un modelo de calidad instanciado para un experimento concreto, que permite evaluar y mejorar la calidad de sus datos.

Check **(Verificar)**

En esta etapa se verifica si se alcanzaron los objetivos previstos, esto es, si el modelo de calidad utilizado para medir la calidad de los datos de un caso (experimento) particular, fue adecuado y suficiente. Se identifica si existen nuevas métricas que es necesario considerar pero no fueron incluidas aún en el modelo, y a qué dimensiones y factores de calidad corresponden.

Act **(Actuar)**

En esta etapa se analizan y corrigen las desviaciones detectadas, y se aplican las mejoras identificadas al modelo. Es decir, se actúa mejorando el modelo de calidad para ser utilizado en próximos experimentos.

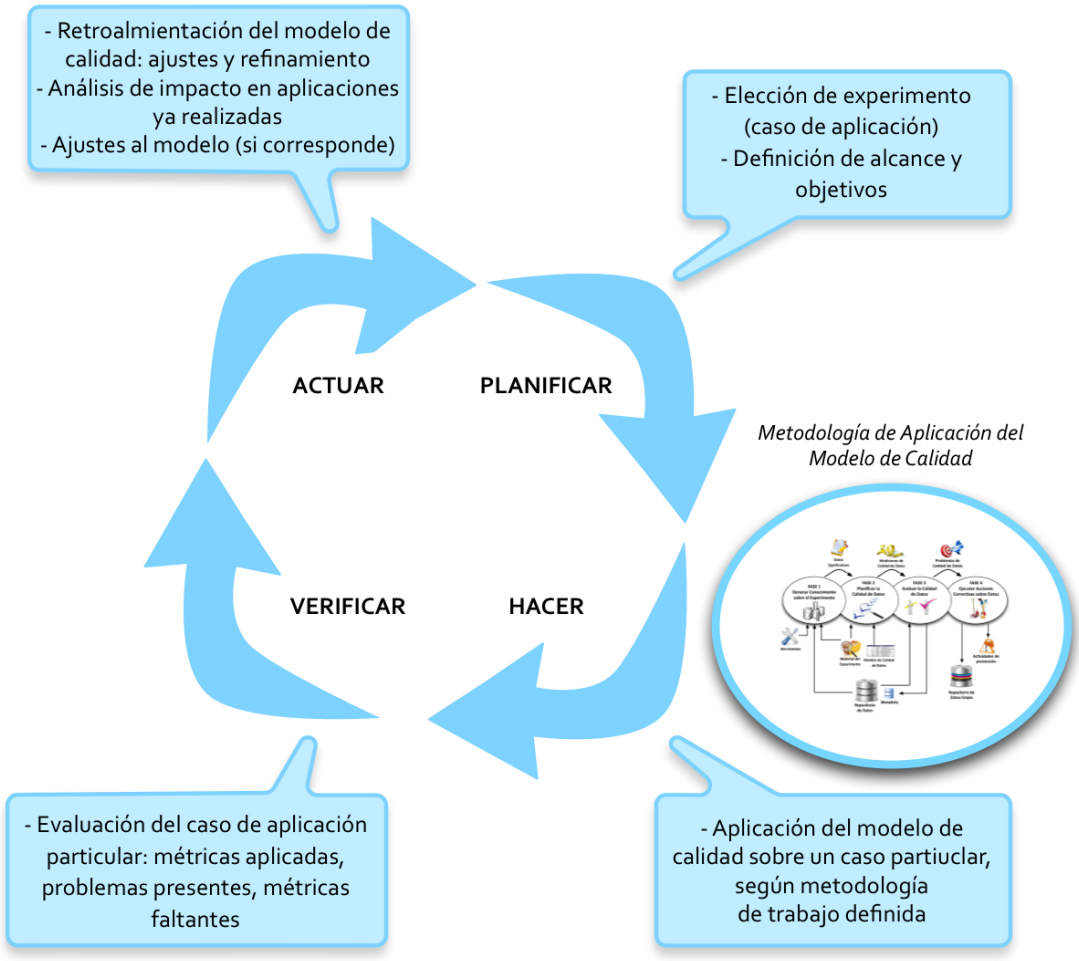

Ilustración 4: Estrategia para la mejora continua del modelo de calidad de datos

Inicialmente se construye una primera versión del modelo de calidad (primera iteración). Luego, el modelo queda inmerso en un ciclo de mejora continua (segunda iteración). Este ciclo consiste en el refinamiento progresivo y retroalimentación de las "lecciones aprendidas" a partir de la aplicación del modelo de calidad base (siempre la última versión generada) sobre datos de sucesivos casos (experimentos) particulares.

Parte de las lecciones aprendidas provienen de las respuestas a los cuestionarios de satisfacción realizados a los experimentadores (ver Capítulo 9), con el objetivo de conocer su opinión y experiencia respecto al uso y aplicación del modelo y la metodología de calidad. A partir de los resultados obtenidos para las diferentes variables de satisfacción y de los aspectos positivos y negativos que se planteen, se identifican posibles acciones de mejora a aplicar.

El modelo se refina y ajusta de acuerdo a las características de la realidad bajo estudio. En este sentido, la metodología de investigación seguida es *iterativa e incremental*: el modelo se va complementado luego de cada caso de aplicación particular, ya sea por los aportes y experiencia de los experimentadores o del propio analista de calidad de datos.

El refinamiento del modelo puede implicar la inclusión, eliminación o modificación de las métricas de calidad que conforman el modelo, así como agregar alguna nueva dimensión o factor de calidad que se identifique necesaria. A modo de ejemplo, la inclusión de las dimensiones de *Interpretabilidad* y *Representación* se realiza a raíz de la aplicación del modelo de calidad a experimentos que utilizan planillas de cálculo como repositorio de sus datos.

En caso de que el modelo de calidad de datos haya sufrido alguna variante luego de su aplicación sobre algún caso particular, entonces se analiza el impacto del cambio sobre los casos de aplicación ya ejecutados. De esta forma, se evalúa si el cambio es generalizable para todos los casos. Si es así, se aplica el cambio. El responsable del modelo de calidad es quien determinará si se realizan los ajustes propuestos sobre el modelo base. A modo de ejemplo, en el caso presentado en el párrafo anterior se verifica si estas mismas dimensiones (*Interpretabilidad* y *Representación*) resultan también aplicables sobre experimentos que utilizan otros tipos de repositorios (bases de datos relacionales).

En cada refinamiento del modelo no se debe perder de vista cuál es nuestro objetivo final: que el modelo de calidad pueda ser aplicado sobre datos de experimentos con las características antes mencionadas. Es por esto que cada vez que se ajusta el mismo, se debe tener presente la generalidad que se desea alcanzar.

5.2.1 Roles

En la Tabla 3 se presentan los roles que participan por cada iteración de investigación, y por cada etapa del ciclo PDCA. En este caso participa el rol de Responsable del Modelo de Calidad de Datos. Debido a que su principal responsabilidad es la generación y mantenimiento del modelo de calidad, su participación es en la primera iteración para la construcción de la versión inicial del modelo, y en la segunda en caso que surja una mejora o cambio sobre el mismo. Es quien analizará el impacto de los cambio propuestos al modelo y aprobará en caso de llevarlo a cabo.

El modelo de calidad es definido para ser aplicado sobre los datos generados durante la ejecución de experimentos controlados en ingeniería de software que involucran sujetos humanos. Por este motivo, no podemos afirmar si el modelo que definimos en este trabajo podrá ser aplicado sobre datos generados en otras áreas de la ingeniería de software, o incluso de la ingeniería de software empírica. Además, cada contexto tiene particularidades que es necesario considerar ya que estas podrán limitar el dominio de aplicación del modelo presentado.

Iteración de Investigación	**Etapa Ciclo PDCA**	**Roles Participantes**
		Responsable del Modelo de Calidad
Primera iteración	N/A	Sí
Segunda iteración	*Plan* (Planificar)	Sí
	Do (Hacer)	No
	Check (Verificar)	Sí
	Act (Actuar)	Sí

Tabla 3: Roles participantes en la metodología de investigación y aplicación de modelo de calidad de datos

En nuestro trabajo, la metodología de trabajo y el modelo de calidad propuestos fueron aplicados sobre los datos de cuatro experimentos: UdelaR, Base-UPV, Replic-UPV y UPM. Esto se muestra en la Ilustración 5. Partiendo desde el modelo de calidad construido inicialmente y la metodología de aplicación definida, el ciclo de mejora continua y la metodología de aplicación del modelo de calidad fueron ejecutados en cuatro oportunidades. Como se muestra en la Ilustración 5, surgieron mejoras y ajustes al modelo luego de la segunda aplicación del modelo de calidad (Experimento Base MDD-UPV). Una vez introducidas las mejoras (se identifican nuevas dimensiones, factores y métricas de calidad que son importantes considerar), se vuelve a aplicar el modelo de calidad sobre los dos casos que habían sido previamente ejecutados.

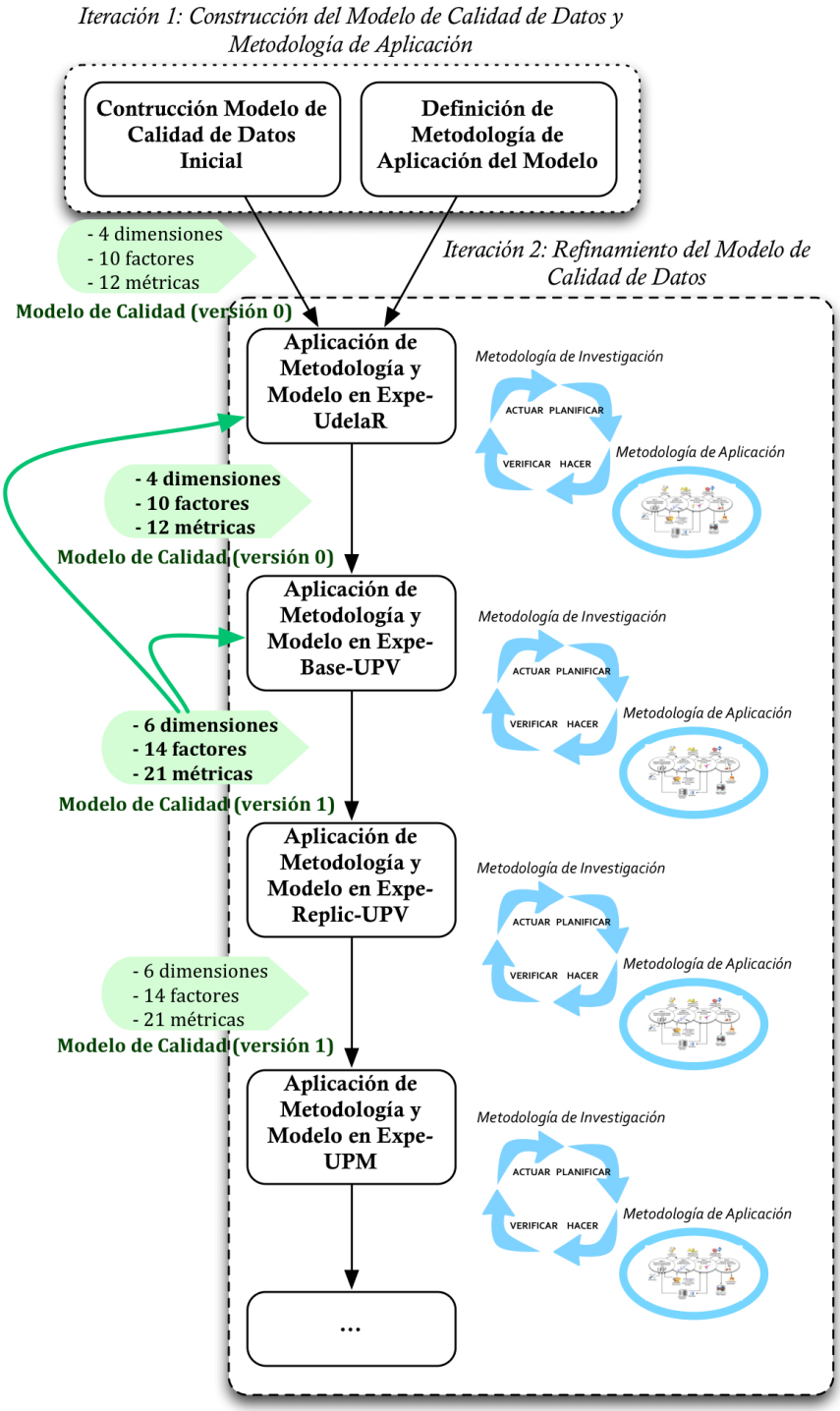

Ilustración 5: Aplicaciones, ajustes y mejoras sucesivas del modelo de calidad de datos

Capítulo 6: Modelo de Calidad de Datos y Metodología de Aplicación para Experimentos en Ingeniería de Software

En este capítulo se presenta el modelo de calidad de datos propuesto para evaluar y mejorar la calidad de los datos en experimentos de ingeniería de software, y se definen las métricas de calidad que lo conforman. Luego se presenta la metodología de trabajo propuesta para aplicar el modelo de calidad sobre los datos de experimentos concretos.

6.1 Modelo de Calidad de Datos para experimentos en Ingeniería de Software

Es posible definir un modelo de calidad de datos que resulte aplicable a los casos particulares de un determinado contexto. En particular, en este trabajo definimos un modelo de calidad que puede ser aplicado sobre los datos resultantes de la ejecución de Experimentos en Ingeniería de Software que involucran sujetos humanos. Este modelo proporciona la base para evaluar la calidad de los datos que se recolectan para la realidad bajo estudio.

El modelo propuesto surge como resultado de refinamientos sucesivos a partir de su aplicación sobre los datos de distintos experimentos de ingeniería de software [32], [33], como fue presentado en el Capítulo 5. De esta forma, las métricas de calidad que componen el modelo fueron definidas por inducción, ya que fueron aplicadas sobre experimentos específicos (de lo particular), y ajustadas de forma tal que puedan llegar a ser aplicadas a datos de otros experimentos en ingeniería de software (a lo general).

A continuación se definen más específicamente los conceptos referentes a la medición de la calidad de datos que serán utilizados y aplicados a lo largo de este trabajo. Estas definiciones se basan en conceptos de la teoría de la medida [86]–[88] y del área de calidad de datos [19].

Se define métrica como el criterio a partir del cual se genera un mapeo entre un atributo de una entidad y un valor de medición, generalmente numérico. Para este trabajo, una métrica genera un mapeo de factores de calidad de los datos de experimentos en ingeniería de software a un valor de medida (en general entre cero y uno).

Se define medición como el acto de medir. Corresponde al proceso mediante el cual se asignan números o símbolos a atributos de una entidad del mundo real. Para este trabajo, la medición es la aplicación de una métrica sobre atributos de entidades particulares del dominio para obtener la medida de un atributo.

Se define medida como el número o símbolo asignado a un atributo de una entidad del mundo real, obtenido mediante una medición y que corresponde al mapeo de dicho atributo con el valor de medición definido por la métrica. La medida corresponde al valor de calidad obtenido como resultado de la medición.

En la Tabla 4 se encuentra definido el modelo de calidad de datos propuesto en el marco del presente trabajo, construido mediante la metodología de investigación descrita en el próximo capítulo. Se define además cada una de las métricas de calidad que conforman el Modelo de Calidad de Datos para Experimentos en Ingeniería de Software, y cómo se mide cada una de ellas.

Cada métrica del modelo de calidad queda definida por las siguientes características:

- *Semántica.* Corresponde a la descripción de cada métrica.
- *Definición.* Indica de qué forma se debe aplicar la métrica sobre los objetos del dominio.
- *Unidad del resultado de la medición.* Puede ser un booleano (0 si contiene un problema de calidad, 1 en caso contrario), un grado (valor entre 0 y 1), o un enumerado (bueno, regular, malo).
- *Granularidad.* Puede ser a nivel de celda, columna, tupla, tabla o el conjunto de datos entero.

Se definen además las siguientes características asociadas a cada métrica. Estas características pueden variar para cada métrica en cada aplicación.

- *Clasificación.* Permite clasificar la naturaleza de los problemas de calidad. Existe un problema de calidad en los datos si el valor de calidad obtenido es menor que 1,00 o "Regular" (considerando como escala la unidad del resultado de la medición definido anteriormente). Los problemas de calidad se clasifican como sigue.
  - *Error en los datos*: se tiene la certeza de que el problema identificado corresponde a un dato erróneo y por lo tanto, siempre que sea posible, debe ser corregido.
  - *Valor sospechoso:* no se puede asegurar si realmente existe un error en los datos, ya que el problema de calidad podría deberse a la existencia de un outlier (suceso particular o fuera de lo común que puede corresponder o no a un evento de la realidad). El hecho de que existan valores anómalos pero correctos es parte de toda experiencia empírica. La forma de asegurar la existencia de un dato erróneo es la comparación del valor registrado con el valor real. Si no es posible conocer el valor real, entonces no se podrán realizar correcciones sobre los datos.
  - *Oportunidad de mejora*: se identifican aspectos que pueden ser mejorados, y que de no hacerlo podría llegar a provocar un potencial error en los datos a futuro. Estos casos, a pesar de que también corresponden a problemas de calidad, no son errores sobre datos particulares, sino que corresponden a sugerencias de mejora sobre el conjunto de datos completo.

  Mientras que las dos primeras clasificaciones están enfocadas en la corrección, las oportunidades de mejora se enfocan en la prevención de errores.
- *Nivel de riesgo.* El nivel de riesgo es obtenido a partir de la experiencia (aplicación del modelo de calidad a datos de diferentes experimentos). Establece tres niveles basados en la proporción de veces en las que se detecta la presencia de un problema de calidad en los datos. En este caso, se establece el nivel de riesgo en base a las aplicaciones del modelo de calidad a datos de experimentos realizados por esta tesis en particular. Cada grupo de investigación que aplique el modelo de calidad de datos podría modificarlo de acuerdo a sus propias aplicaciones.
  - *Alto*: para todos los casos de aplicación se encuentra la presencia del problema de calidad.
  - *Medio*: para más de la mitad de los casos de aplicación se encuentra la presencia del problema de calidad.
  - *Bajo*: para menos de la mitad de los casos de aplicación se encuentra la presencia del problema de calidad.

La Ilustración 6 presenta de forma gráfica cómo se define el modelo de calidad propuesto y cómo se aplica a los datos de un experimento en particular. Este modelo está basado en los conceptos del área de Calidad de Datos presentados en el Capítulo 2, e instanciado al dominio específico de Experimentos en Ingeniería de Software. Llamamos métrica de calidad instanciada a la aplicación de una métrica de calidad sobre un objeto particular del dominio, y modelo de calidad de datos instanciado al conjunto de métricas instanciadas para un experimento particular.

La Tabla 4 presenta el modelo de calidad de datos para su instanciación sobre objetos particulares del dominio. Las métricas de calidad se aplican a los niveles de granularidad definidos, de forma de conocer cuáles son los datos particulares que presentan algún problema de calidad y así tomar las acciones que correspondan. Para disminuir el nivel de granularidad se calculan agregaciones de los valores de calidad obtenidos mediante la medición sobre los datos correspondientes. Para los casos en

que el resultado es un booleano o grado, la agregación consiste calcular el ratio como la sumatoria de valores obtenidos (entre 0 y 1) sobre la cantidad de elementos medidos. A modo de ejemplo, si una métrica se define con nivel de granularidad "celda" y la unidad de medida es "booleano", entonces se calcula la agregación al nivel de granularidad "columna" como el ratio de los valores de calidad obtenidos para cada celda. De la misma forma se calcula la agregación para pasar del nivel de granularidad "tupla" a "tabla". Para el nivel de granularidad "conjunto de datos" no hay agregaciones posibles.

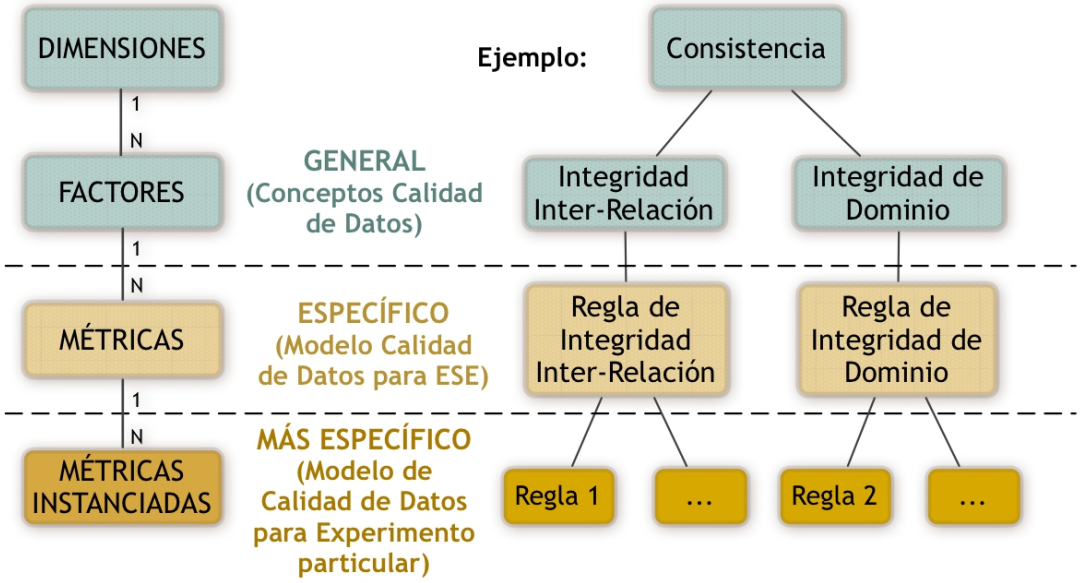

Ilustración 6: Modelo de Calidad de Datos para Experimentos en Ingeniería de Software

Se define también una agregación para cada métrica de calidad, para pasar al nivel de granularidad "conjunto de datos". La medida agregada para una métrica se calcula como el promedio de los valores de calidad obtenidos para todos los casos (ya sea columna o tabla) sobre los cuales fue aplicada la métrica. De esta forma, un valor de calidad menor a 1,0 estará indicando la presencia de un problema de calidad en los datos. Esta agregación permite conocer cuáles son los problemas de calidad que están presentes en un experimento dado, independientemente de los datos particulares que lo contengan.

Dimensión de Calidad	Factor de Calidad	Métrica de Calidad	Semántica	Definición	Unidad de la medida	Granularidad
Exactitud	Exactitud Sintáctica	Valor fuera de rango	Valores situados fuera de rango definido como válido	Valor entre 0 y 1 tal que un valor más cercano a 0 indica que la distancia entre el valor medido y el rango definido es mayor, mientras que un valor más cercano a 1 indica que la distancia al rango es menor.	Booleano o grado	Celda
		Falta de estandarización	Valores registrados en diferentes formatos o en un formato diferente al definido como estándar	1 si tiene el formato adecuado 0 en caso contrario	Booleano	Celda
		Valor embebido	Valores embebidos dentro de otros (ej. en un texto y debe procesarse por separado)	1 si tiene algún valor embebido 0 en caso contrario	Booleano	Celda
	Exactitud Semántica	Registro inexistente	Registros que no corresponden a ningún objeto de la realidad	1 si el registro no existe en la realidad 0 en caso contrario	Booleano	Tupla
		Registro con errores	Valores con desviaciones respecto a lo que sucedió en la realidad	Valor entre 0 y 1 que indica la desviación del valor real respecto del valor registrado	Grado	Celda
		Valor fuera de referencial	Valores situados fuera de referencial definido como válido	1 si el valor es parte del referencial 0 en caso contrario	Booleano	Celda
	Precisión	Falta de precisión	Valores que no contienen el nivel de detalle o precisión requerida	1 si tiene la precisión adecuada 0 en caso contrario	Booleano	Celda
Completitud	Densidad	Valor nulo	Valores nulos que deberían ser no vacíos	1 si contiene algún valor no vacío 0 en caso contrario	Booleano	Celda
		Información omitida	Registros para los cuales se omitió el ingreso de cierta información	1 si contiene toda la información 0 en caso contrario	Booleano	Tupla
	Cobertura	Registro faltante	Registros que existen en la realidad pero se omitió su ingreso	1 si el registro fue ingresado 0 en caso contrario	Booleano	Tupla

Consistencia	Integridad de dominio	Regla de integridad de dominio	Valores que no cumplen con la regla de integridad sobre el dominio de un atributo	1 si cumple la regla de integridad 0 en caso contrario	Booleano	Celda
	Integridad intra-relación	Regla de integridad intra-relación	Valores que no cumplen con la regla de integridad sobre atributos de una relación	1 si cumple la regla de integridad 0 en caso contrario	Booleano	Celda
		Valor único	Valores repetidos que deben ser únicos	1 si si el valor es único 0 en caso contrario	Booleano	Celda
	Integridad inter-relación	Regla de integridad inter-relación	Valores que no cumplen con la regla de integridad sobre atributos de diferentes relaciones	1 si cumple la regla de integridad 0 en caso contrario	Booleano	Celda
		Referencia inválida	Registros con referencias a otros registros que no existen (son inválidas)	1 si contiene alguna referencia inválida 0 en caso contrario	Booleano	Tupla
Unicidad	Duplicación	Registro duplicado	Registros ingresados por duplicado (repetidos de manera exacta)	1 si el registro está duplicado 0 en caso contrario	Booleano	Tupla
	Contradicción	Registro contradictorio	Registros ingresados de forma contradictoria (repetidos con contradicciones)	1 si el registro es contradictorio 0 en caso contrario	Booleano	Tupla

Representación	Estructura de datos	Estructura de datos	Conjunto de datos representados en una estructura adecuada para la realidad dada	Bueno: restricciones entre datos definidas y documentadas; representación de datos concisa y consistente con la realidad Regular: restricciones entre datos definidas de forma implícita; representación de datos adecuada para la realidad Malo: restricciones entre datos no definidas; representación de datos no adecuada para la realidad	Enumerado	Conjunto de datos
	Formato de datos	Formato de datos	conjunto de datos representados en un formato consistente	Bueno: utilización del mismo formato para representar los mismos datos Regular: utilización de formato similar para representar los mismos datos Malo: utilización de diferente formato para representar los mismos datos	Enumerado	Conjunto de datos
Interpretabilidad	Facilidad de entendimiento	Facilidad de entendimiento	Conjunto de datos entendibles por quien va a hacer uso de ellos	Bueno: datos entendibles por alguien ajeno al experimento que hará uso de ellos Regular: es necesario hacer consultas particulares para lograr su entendimiento Malo: es necesario una descripción detallada para lograr su entendimiento	Enumerado	Conjunto de datos
	Metadata	Metadata	Utilización de metadata para describir el conjunto de datos	Bueno: existe esquema conceptual; se define y registra metadata e información de trazabilidad Regular: los conceptos de la realidad están claros; se define metadata e información de trazabilidad de forma implícita Malo: no existe esquema conceptual; no se define ni registra metadata o información de trazabilidad	Enumerado	Conjunto de datos

Tabla 4: Modelo de Calidad de Datos para Experimentos en Ingeniería de Software

A modo de ejemplo, consideramos la métrica "Valor nulo" aplicada sobre los atributos "tiempo" y "línea". Supongamos que la aplicación de la métrica al atributo "tiempo" da como resultado que existen 4 nulos entre los 10 datos medidos (valor de calidad agregado por columna = 0,6), mientras que para el atributo "línea" no existe ningún nulo (valor de calidad agregado por columna = 1,0). Entonces, el valor de calidad agregado para la métrica "Valor nulo" es 0,8 (promedio entre 0,6 y 1,0). Además, como el valor de calidad para la métrica es 0,8 (menor que 1,0), sabemos que el problema de calidad "Valor nulo" está presente en los datos del experimento analizado.

6.1.1 Metadatos de calidad

Para registrar los resultados obtenidos a partir de las mediciones es necesario definir los metadatos de calidad que se utilizarán. Los metadatos indican cuáles son los datos que contienen algún problema de calidad, y se asocia el resultado a cada medición particular. Además, a partir de estos metadatos es posible conocer los valores de calidad asociados a cada medición y calcular las agregaciones correspondientes.

En cada aplicación particular se define la estructura de los metadatos a registrar. Esto dependerá de la realidad que se está analizado, la forma en que son almacenados los datos y qué repositorio se utiliza.

En todos los casos, se registran al menos los siguientes metadatos:

- Qué métrica fue aplicada.
- Sobre qué datos se aplicó.
- Si el dato presenta o no el problema de calidad medido.

La Ilustración 7 presenta mediante un modelo conceptual cuáles son las principales entidades y atributos que representan el registro de metadatos de calidad que utilizamos en este trabajo. Cada métrica de calidad es definida en base a un factor y dimensión de calidad. Por otra parte, una métrica es instanciada sobre objetos del dominio, que se traducen a tablas y atributos en el repositorio de datos medido. Finalmente, es necesario identificar cuáles son las tuplas que contienen el problema de calidad medido.

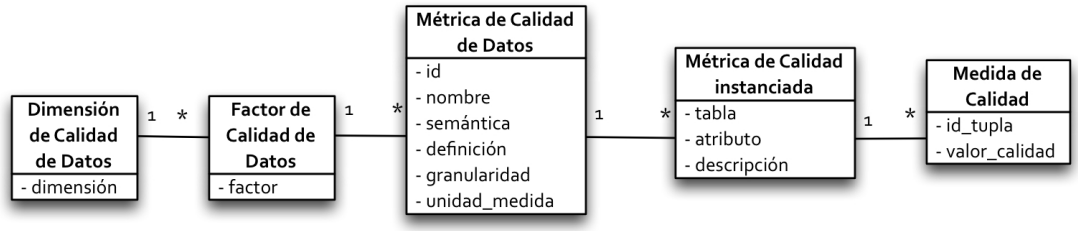

Ilustración 7: Metadatos de calidad

En la Ilustración 8 se muestra a modo de ejemplo cómo se instancia el modelo de metadatos de calidad definido al caso particular de "Valor Nulo" sobre los atributos "tiempo" y "línea".

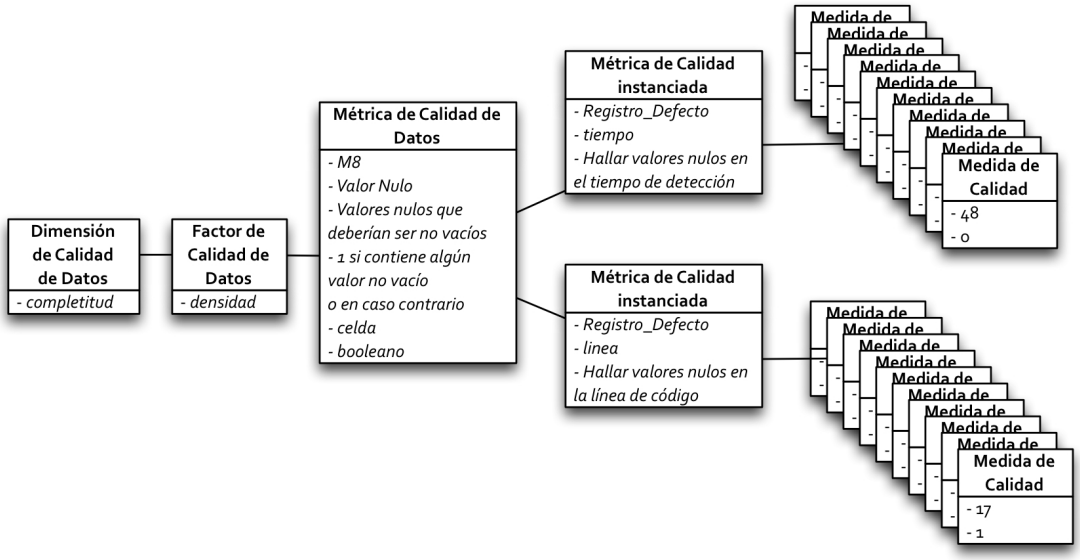

Ilustración 8: Ejemplo de aplicación de metadatos de calidad

6.1.2 Métricas de Calidad de Datos

En esta sección se presenta una descripción detallada de las métricas incluidas en el Modelo de Calidad de Datos propuesto.

Valor fuera de rango

Definición. Resulta de interés que los valores se encuentren dentro de un rango determinado, el cual debe ser previamente definido de acuerdo a la realidad y el contexto bajo estudio. El rango puede definirse en base a la experiencia del experimentador (juicio de experto), consultando datos históricos, o ser determinado de forma estadística.

Si un valor no pertenece al rango determinado como válido no significa necesariamente que dicho valor sea erróneo. El resultado de esta métrica estará alertando sobre la existencia de valores sospechosos, una advertencia de que algo "fuera de lo común" puede haber sucedido. No será posible asegurar la existencia de un error real sobre los datos salvo que se pueda comparar el valor registrado con el valor de la realidad (por ejemplo, comparando con otra fuente de datos, o consultando a quien registró los datos).

Medición. La medición consiste en establecer, para todos los valores involucrados, si se encuentran fuera del rango determinado como válido.

Para cada uno de los casos identificados se establecen los criterios para determinar apropiadamente el rango que se va considerar para evaluar los valores, y se identifican los outliers.

Se considerará libre de errores a aquellos valores dentro del rango definido, mientras que los datos fuera de dicho intervalo se analizarán de manera aislada. Se debe comprobar para cada uno de estos casos si corresponden a errores en los datos o a sucesos de la realidad "fuera de lo común".

Falta de estandarización

Definición. Interesa que los datos se encuentren registrados de forma estándar, de forma tal de que las operaciones o manipulaciones que se realicen con estos valores tengan sentido. Esto puede incluir el formato de los datos, la unidad de medida o el tipo de datos, entre otros. A modo de ejem-

plo, no se podría realizar una operación aritmética entre dos valores de tiempos registrados en horas y minutos respectivamente.

Medición. La medición consiste en verificar si los valores no se encuentran registrados con el formato adecuado. Para cada uno de los casos identificados se debe definir cuál es el formato en el que deberían estar representados los datos, e identificar aquellos que no lo cumplen.

Valor embebido

Definición. Hay datos que pueden estar embebidos en un texto libre, y que no podrían ser procesados o analizados si no se obtienen individualmente. Como un ejemplo sencillo se puede pensar en un dato de "domicilio" que tiene embebido el nombre de la calle, el número de puerta, el código postal, el barrio y la ciudad. Estos datos deberían registrarse por separado para poder ser manipulados individualmente.

Medición. La medición consiste en verificar si existen datos que contengan valores embebidos en algún texto libre, y que será necesario analizar individualmente.

Registro inexistente

Definición. Se identifican los registros que se encuentran almacenados en el repositorio de datos pero no se asocian a ningún objeto de la realidad. Los registros inexistentes no deberían formar parte del repositorio de datos ya que no reflejan la realidad que se está queriendo representar.

Medición. La medición consiste en verificar si existen registros en el repositorio de datos que no corresponden a objetos de la realidad.

Registro con errores

Definición. Hay registros que existen en la realidad pero se asocian de manera incorrecta a un objeto real. Esto significa que los datos que se registran no son válidos para ese objeto de la realidad (a pesar de que pueden ser valores sintácticamente correctos). Un registro con errores no permitirá, en general, lograr identificar a partir de sus datos a qué objeto de la realidad corresponde.

Medición. La medición consiste en verificar si existen registros en el repositorio de datos que contienen datos incorrectos o inválidos para un objeto dado.

Valor fuera de referencial

Definición. Los valores deben pertenecer a los referenciales definidos. Para aquellos campos que contienen valores predefinidos, se puede verificar que sólo existan los considerados válidos. Si se incorporan datos externos o se ingresan datos directamente en el repositorio, podrían existir valores que no corresponden a ninguno de los permitidos por un referencial dado.

Medición. La medición consiste en identificar cuáles son los datos cuyos valores deben pertenecer a un referencial dado y compararlos contra los considerados válidos.

Falta de precisión

Definición. Resulta de interés conocer si el nivel de precisión o detalle con el que se registran los valores es el esperado para realizar los cálculos correspondientes. La falta de precisión (por ejemplo, un redondeo a número entero en un campo que requiere decimales) puede impactar en la precisión del resultado obtenido.

Medición. La medición consiste en verificar si existen valores que no se encuentran registrados con el nivel de precisión adecuado.

Valor nulo

Definición. En todo repositorio de datos pueden existir valores nulos. Sin embargo, interesa conocer cuáles son los datos que no fueron registrados pero que deberían contener algún valor diferente

de vacío. Siempre que sea posible, se desea conocer la causa de la omisión y determinar el valor que debería tomar en lugar de nulo. Un nulo puede tener diferentes interpretaciones y por lo tanto debe quedar claramente establecido cuál es su significado (por ejemplo, un nulo en un tiempo puede significar que el tiempo no se tomó, que se olvidó registrarlo o que su valor es 0).

La existencia de valores nulos influye en el análisis de los datos que se lleve a cabo, ya que al obtener estadísticas de los mismos se hace necesario dejar de lado aquellos valores vacíos. Por ejemplo, un nulo en un dato que no sea considerado como tal al momento de calcular un promedio, afectará el resultado obtenido.

Medición. La medición consiste en verificar si existen valores nulos en los datos que deberían contener algún valor no vacío. Se deben identificar cuáles son los datos que admiten nulos y cuáles no, según las restricciones definidas en el repositorio de datos. Luego se identifican aquellos que admiten nulos, pero deberían contener algún valor distinto de vacío. Este último caso es el que interesa medir.

Información omitida

Definición. Interesa conocer qué información fue omitida para determinado objeto de la realidad. La diferencia con la métrica "Valor nulo", es que mientras esta considera valores individuales, la presente métrica considera un conjunto de datos o información que debe ser registrada.

Medición. La medición consiste en verificar si existe información omitida (conjunto de valores nulos) para los datos en los que se requiere contener algún no vacío.

Registro faltante

Definición. Interesa conocer si existen registros de la realidad que no fueron ingresados en el repositorio de datos. Si esto sucediera, se estaría considerando sólo una porción de los datos de la realidad. En este caso no se verifica la existencia de cada dato en particular, sino del registro mismo.

Medición. La medición consiste en verificar si existen registros de la realidad que no fueron ingresados en el repositorio bajo estudio. Es posible tomar algún referencial o repositorio de datos externo que represente la realidad.

Regla de integridad de dominio

Definición. Se definen reglas que deben cumplirse sobre el dominio válido al cual deben pertenecen los valores. A diferencia de la métrica "Valor fuera de rango", en este caso se conoce el dominio exacto de valores. Mientras que en el primero el rango definido puede ser arbitrario y no asegurar la existencia de un dato erróneo, en este caso el dominio es conocido y por lo tanto, se estará detectando un error en el dato.

Medición. La medición consiste en establecer, para todos los valores involucrados, si se encuentran fuera del dominio determinado como válido. Se identifica para qué datos se deben definir reglas, y cuál es el dominio válido para cada caso.

Regla de integridad intra-relación

Definición. Se definen un conjunto de reglas sobre los datos de una misma relación que deben ser satisfechas sobre el repositorio bajo estudio. El hecho de que alguna de estas reglas sea violada, afecta la consistencia de los datos y por lo tanto cualquier análisis que se lleve a cabo a partir de estos. Si estas reglas ya fueron definidas sobre el repositorio de datos y se controla su cumplimiento, entonces no será necesario considerar esta métrica.

Medición. La medición consiste en verificar si se viola alguna de las reglas de integridad intra-relación definidas para los datos de la realidad bajo estudio.

Valor único

Definición. Interesa conocer si existen registros que contengan el mismo valor en datos (distintos al identificador que ya lo cumple por definición) que deberían ser valores únicos.

Medición. La medición consiste en verificar cuáles son los valores que no cumplen con las restricciones de unicidad definidas. Se identifican cuáles son los datos sobre los cuales no se establece la restricción de valor único pero deberían tener algún valor diferente a todos los demás.

Reglas de integridad inter-relación

Definición. Es necesario considerar la satisfacción de reglas sobre datos de distintas relaciones que deben cumplirse sobre el repositorio bajo estudio. El hecho de que alguna de estas reglas sea violada, afecta la consistencia de los datos. Al igual que para las reglas de integridad intra-relación, para los casos en que existan reglas ya definidas que controlen su cumplimiento, entonces no será necesario considerar esta métrica.

Medición. La medición consiste en verificar si se viola alguna de las reglas de integridad inter-relación definidas para la realidad bajo estudio.

Referencia inválida

Definición. Interesa identificar si hay referencias a registros que no existen en el repositorio de datos, y por lo tanto resultan ser referencias inválidas.

Medición. La medición consiste en verificar si existen registros que contengan referencias inválidas. Se analizan si existen restricciones de integridad referencial, y se identifican los datos para los cuales se omitió su definición.

Registro duplicado

Definición. Resulta de interés identificar si existen dos o más registros que fueron ingresados de forma repetida. Pueden suceder dos situaciones: cuando contienen el mismo valor en su identificador y demás datos (o en su defecto valores nulos); a pesar de contener distinto identificador, hacen referencia al mismo objeto de la realidad y contienen los mismos valores en los datos que se definan (según el criterio de duplicación considerado).

Medición. La medición consiste en definir los criterios de duplicación para los casos que correspondan, y luego realizar los chequeos necesarios para verificar la existencia de registros duplicados.

Registro contradictorio

Definición. Resulta de interés identificar si existen dos o más registros que aparecen repetidos de manera contradictoria. Esto significa que contienen distinto valor en su identificador y/o demás datos (o en su defecto valores nulos), a pesar de que hacen referencia al mismo objeto de la realidad. De no considerar estos casos, se podría estar contando dos veces el mismo registro.

Medición. La medición consiste en definir los criterios de contradicción para los casos que correspondan, y luego realizar los chequeos necesarios para verificar la existencia de registros contradictorios.

Las métricas que se plantean a continuación se miden sobre el conjunto de datos. La medida será alguno de los valores: bueno, regular o malo, dependiendo del grado de cumplimiento de los criterios definidos para cada métrica, mediante una evaluación subjetiva.

Estructura de datos

Definición. Es de interés que los datos estén representados en una estructura adecuada para la realidad dada.

Medición. La medición consiste en verificar si se encuentran definidas en el repositorio de datos las reglas y restricciones de integridad que existan sobre los datos. También es importante evaluar si la representación de los datos es concisa y consistente, y se adecua a la realidad dada.

Formato de datos

Definición. Es de interés que los datos estén representados en un formato consistente, de forma tal que los mismos datos sean siempre representados de igual forma.

Medición. La medición consiste en verificar si se utiliza el mismo formato para representar los mismos datos.

Facilidad de entendimiento

Definición. Es necesario que los datos sean entendibles por quienes van a hacer uso de ellos. Además del experimentador que tendrá un profundo conocimiento del repositorio y sus datos, los datos deben ser entendidos por cualquier persona que requiera utilizarlos (por ejemplo, el mismo analista en calidad de datos al analizar y evaluar su calidad).

Medición. La medición consiste en evaluar la claridad y no ambigüedad en el significado y uso de los datos. Se mide el grado en que la información es capaz de ser entendida e interpretada, el alcance en que los datos son claros, sin ambigüedades y fácilmente comprensibles.

Metadata

Definición. Es de interés que se utilice documentación y metadata para describir los datos del repositorio, de forma de poder interpretar correctamente el significado y propiedades de las fuentes de datos.

Medición. La medición consiste en verificar la existencia del esquema conceptual correspondiente, de la definición de metadata, y de la información histórica y de trazabilidad, de forma de poder conocer el origen de los datos.

6.2 Metodología para la aplicación del modelo de Calidad de Datos

En esta sección se describe la metodología de trabajo que proponemos para utilizar el modelo que desarrollamos. Esta metodología fue aplicada en todos los experimentos que analizamos y, al igual que el modelo, fue construida mediante refinamientos sucesivos.

Esta metodología define los pasos y guías para aplicar el Modelo de Calidad de Datos a los datos resultantes de la ejecución de Experimentos en Ingeniería de Software que involucran sujetos humanos. La metodología es genérica, por lo que podría ser aplicada sobre cualquier experimento con estas características. En la Ilustración 9 se muestra cuáles son las fases de la metodología propuesta. Para cada fase describiremos los artefactos de entrada a la misma, las actividades principales que se utilizan y realizan en la fase, así como los artefactos de salida.

6.2.1 Fase 1: Generar conocimiento del experimento

La primera fase consiste en conocer el experimento y la realidad bajo estudio. En particular, se requiere conocer el diseño experimental incluyendo todos los aspectos relevantes tales como: objetivos, sujetos, objetos, factores, alternativas y variables de respuesta. El foco del relevamiento estará en los datos que se registran y utilizan, quiénes son los responsables y participantes, cuál es el método de recolección de datos y dónde son almacenados. En caso de que se utilicen herramientas para la recolección y/o procesamiento de los datos, también será necesario conocer las herramientas y su forma de uso, así como el repositorio donde se almacenan los datos recolectados.

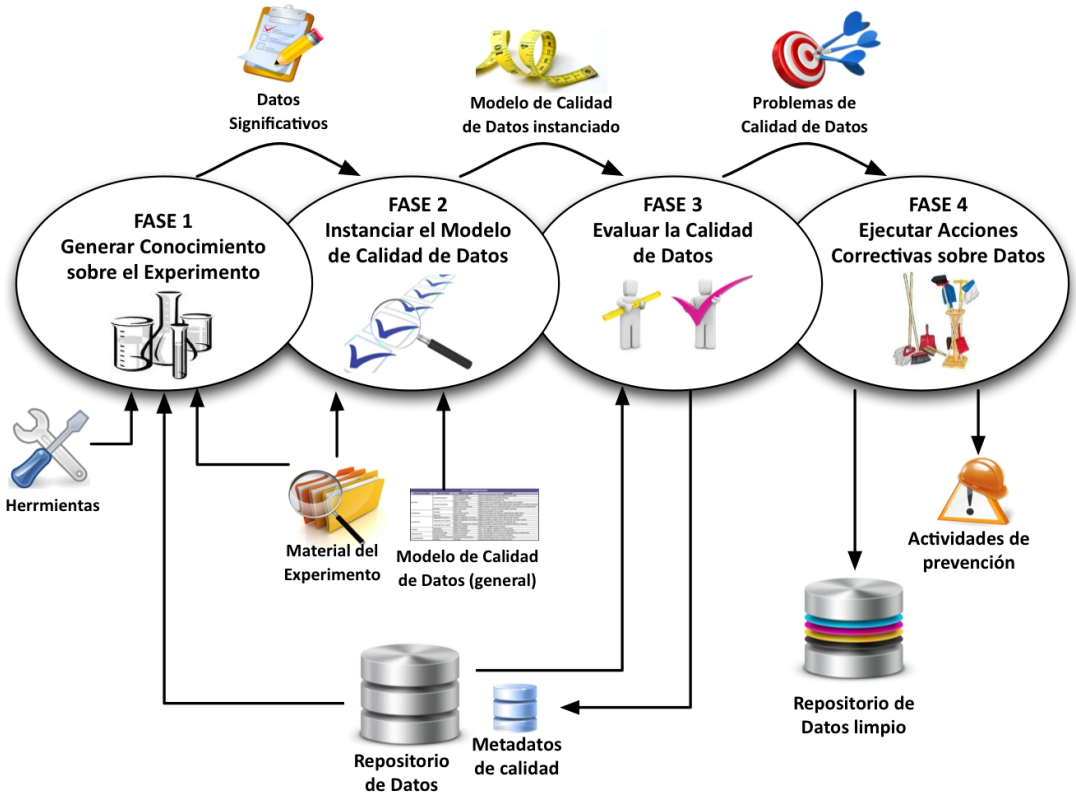

Ilustración 9: Metodología de aplicación del Modelo de Calidad de Datos

Durante esta fase participa activamente el responsable del experimento ya que es quien tiene el conocimiento más profundo y específico que se requiera relevar. A lo largo del trabajo pueden surgir consultas puntuales sobre los datos y la realidad bajo estudio, que deberán ser evacuadas por el responsable del experimento para facilitar el trabajo del analista de calidad (los roles se definen al final de la sección).

Una vez conocido el contexto bajo estudio se identifican cuáles son los datos significativos, ya que son los más importantes en cuanto al análisis de calidad se refiere. Los datos significativos son aquellos que serán utilizados para los análisis estadísticos del experimento. Esto incluye tanto datos crudos como calculados.

En caso de que exista el diagrama de clases y/o diseño de base de los datos a ser analizados se recomienda su uso ya que facilita el entendimiento del experimento y la naturaleza de los datos. En caso de no existir se podría elaborar un modelo de datos (diagrama de clases, modelo ER, etc.) como parte del trabajo de calidad de datos.

Entradas:

- Material sobre el experimento (documentación en general, esquemas/modelos conceptuales, contexto experimental).
- Repositorio de datos.
- Herramienta utilizada para el registro de los datos (si existe).

Actividades principales:

- Reuniones de trabajo entre Analista de Calidad de Datos y Responsables del Experimento.
- Relevamiento de información relativa al experimento y los datos que se recolectan.

- Lectura y análisis de material relevante (artículos, diagramas u otros documentos vinculados al experimento).
- Conocimiento, exploración y análisis de las herramientas utilizadas y del repositorios de datos.

Salidas:

- Contexto experimental documentado, incluyendo la descripción del experimento, su diseño, datos, repositorios, herramientas, entre otros.
- Datos significativos identificados.

6.2.2 Fase 2: Instanciar el modelo de calidad de datos

En el Modelo de Calidad de Datos para Experimentos en Ingeniería de Software se encuentran definidas, por cada dimensión y factor de calidad, cuáles son las diferentes métricas de calidad que lo conforman y que podrían ser aplicadas sobre los datos de un experimento dado. Las métricas definidas son de carácter genérico y podrían ser aplicadas a cualquier caso.

En esta fase se identifican cuáles de esas métricas se aplicarán y sobre qué datos del experimento bajo estudio. De esta forma, queda definido el modelo de calidad instanciado sobre los datos de un experimento en particular. Como se detalla en la sección de "Metodología de investigación", el modelo de calidad se encuentra inmerso en un ciclo de refinamiento y mejora continua. Por este motivo, en caso de identificarse nuevas métricas que no forman parte del modelo de calidad base, se analizarán su inclusión como parte de dicho ciclo, quedando fuera del alcance de la metodología de trabajo aquí descripta.

Se definen distintas estrategias de identificación de métricas que se recomienda seguir, de forma de lograr, en la mayor medida posible, la inclusión de todas las métricas de calidad relevantes.

Estrategia 1: Bottom-up, a partir de los datos recolectados

La primera estrategia consiste en, partiendo de los datos que resultan significativos para los análisis estadísticos (resultantes de la etapa anterior), identificar cuáles son los posibles problemas o errores que podrían presentarse sobre los mismos. A modo de ejemplo, si se registran datos de tiempos se podría pensar que si hay valores menores a 0 entonces estaríamos ante la presencia de un problema de calidad. Este problema es posible medirlo mediante la definición de una regla de integridad de dominio que establezca que los valores de tiempos deben ser siempre mayores que 0. A continuación, elaborar un listado de todos los posibles errores de calidad que podrían darse sobre los datos, mapeando cada uno de ellos con una métrica del modelo de calidad.

Estrategia 2: Top-down, a partir de las métricas de calidad definidas

La segunda estrategia parte de las métricas de calidad. Consiste en analizar cada métrica definida en el modelo de calidad, y a partir de estas identificar los posibles problemas o errores que podrían suceder sobre los datos significativos del experimento. A modo de ejemplo, tomando la métrica de calidad "Registro duplicado", se podría pensar en objetos del dominio que podrían haberse registrado de forma repetida en el repositorio de datos. Incluir los posibles errores identificados en la lista elaborada anteriormente, de forma de consolidar una única lista de posibles errores sobre los datos, todos ellos mapeados con las métricas de calidad que conforman el modelo.

Estrategia 3: A partir de las herramientas utilizadas

Una tercer estrategia considera la herramienta utilizada para el registro de datos (si existe). En este caso, se tiene en cuenta la forma en que se ingresan los datos mediante el uso de la herramienta, los controles que hayan sido implementados sobre la misma, y la forma en que se almacenan

los datos en el repositorio correspondiente. Así se identificarán otros posibles problemas de calidad que puedan suceder sobre los datos, y con qué métricas podrá verificarse su presencia.

Luego de aplicadas las estrategias propuestas, por cada métrica identificada se verifica la factibilidad de su aplicación. Por ejemplo, si los datos se ingresan mediante una herramienta que incluye el control de datos negativos, entonces no existirán tiempos menores a 0 y por lo tanto este control no será considerado. También se deben conocer las restricciones y reglas que se hayan definido sobre los datos, y cuáles de ellas se encuentran implementadas en el repositorio de datos.

A partir de las métricas de calidad que resultan factibles, se definen si existen otros objetos de la realidad (además de los ya identificados) sobre lo que se aplicarán cada una de estas. Por cada métrica, es posible definir varias métricas instanciadas, una por cada objeto sobre el cual se aplica.

El responsable del experimento participa en esta etapa validando las métricas y métricas instanciadas que fueron identificadas. Se debe verificar si los problemas identificados podrían realmente presentarse considerando la realidad y los datos bajo estudio, descartando aquellas que no sean aplicables, e incluyendo nuevas que pueda identificar a partir de su conocimiento de la realidad. A partir de la validación, se realizarán los ajustes que resulten necesarios sobre el Modelo de Calidad instanciado.

Entradas:

- Modelo de Calidad de Datos para Experimentos en Ingeniería de Software, incluyendo la definición de cada métrica de calidad
- Esquema conceptual de los datos
- Reglas/Restricciones definidas sobre los datos
- Datos significativos para los análisis estadísticos

Actividades principales:

- Identificación de métricas de calidad aplicables a los datos del experimento (a partir de las métricas, y a partir de los datos)
- Reuniones de trabajo para validación de métricas de calidad
- Definición de métricas instanciadas

Salidas:

- Modelo de Calidad de Datos instanciado
- Métricas de calidad instanciadas, definidas y validadas

6.2.3 Fase 3: Evaluar la calidad de los datos

Para cada una de las métricas instanciadas que fueron definidas en la fase anterior, se implementan los métodos de medición que correspondan en el repositorio de datos bajo estudio. La forma de implementación puede ser manual o automática, y dependerá de cada medición así como de la forma en que se encuentran almacenados los datos. Por ejemplo, si los datos están almacenados en un base de datos relacional, las mediciones se podrán implementar mediante consultas SQL; mientras que en planillas de cálculo podrían utilizarse fórmulas.

Durante esta fase también se identifica si existen mediciones que no podrán ser ejecutadas, ya sea por su alto costo de implementación asociado, o por no ser factible su ejecución.

Una vez ejecutadas las mediciones, es posible conocer cuáles son los datos que presentan problemas de calidad. Es necesario hacer un análisis caso a caso y diferenciar los que corresponden a errores reales sobre los datos, de los valores sospechosos u oportunidades de mejora, ya que las acciones que se toman en cada caso son diferentes.

Para los casos de errores en los datos, se debe validar con el responsable del experimento si realmente existe un dato erróneo. Para los valores sospechosos se debe analizar si corresponden a su-

cesos inusuales pero reales, o si también son errores en los datos. En ambos casos, siempre que sea posible se aplican las acciones correctivas que correspondan.

Se registran los resultados de todas las mediciones utilizando los metadatos de calidad que hayan sido definidos. El registro de los resultados nos permite no sólo conocer los datos particulares que presentan un problema de calidad, sino también poder calcular las medidas agregadas para cada métrica.

Cada medición ejecutada devuelve la medida (valor de calidad.) También se obtienen las medidas agregadas por métrica.

Entradas:

- Modelo de calidad de datos instanciado
- Repositorio de datos

Actividades principales:

- Ejecución de métodos de medición (consultas SQL, fórmulas de cálculo, algoritmos programados)
- Registro de metadatos de calidad
- Reuniones de trabajo y validación
- Cálculo de medidas

Salidas:

- Problemas de calidad en los datos identificados y clasificados
- Metadatos registrados
- Medidas calculadas para cada medición y métrica de calidad

6.2.4 Fase 4: Ejecutar acciones correctivas sobre los datos

Para los casos en los cuales existe un error en los datos que debe ser corregido, se identifican las formas de limpieza que podrían ser aplicadas. Esto se realiza en conjunto con el responsable del experimento. Para algunos casos, podría suceder que la limpieza no se considere necesaria o no sea factible de implementar (dada su relación costo-beneficio).

Es importante considerar que las mismas acciones correctivas pueden introducir nuevos errores en los datos. Por este motivo, resulta fundamental ser cuidadoso en la selección y aplicación de las acciones correctivas.

Se pueden utilizar diferentes técnicas (presentadas en el Capítulo 2) dependiendo de las características de la fuente de datos y de la naturaleza del error a corregir.

En esta fase se implementan y ejecutan las limpiezas identificadas ya sea de forma automática, semiautomática o manual.

Si se identifican errores de calidad en el diseño del esquema de la base, se puede definir un nuevo esquema que los corrija y realizar la migración de datos correspondiente. La ejecución de las limpiezas se puede realizar sobre la misma base de datos origen, o acompañando la migración a un nuevo repositorio de datos.

Por último se identifican las actividades de prevención y oportunidades de mejora podrían llevarse a cabo de forma tal de evitar la ocurrencia de ciertos problemas de calidad a futuro. Estas actividades podrían incluirse en próximas replicaciones del experimento o en otros experimentos de características similares.

Entradas:

- Problemas de calidad en los datos
- Técnicas de limpieza
- Repositorio de datos origen con metadatos de calidad registrados

Actividades principales:

- Selección de técnicas de limpieza que puedan ser aplicadas para los errores identificados
- Ejecución de limpiezas sobre el repositorio de datos
- Implementación de repositorio de datos destino (si corresponde)
- Ejecución de migración de datos (si corresponde)

Salidas:

- Repositorio de datos "limpio" (con errores corregidos)
- Actividades de prevención identificadas

6.2.5 Roles participantes

En la Tabla 5 se presentan los distintos roles por cada fase de la metodología de aplicación, y por cada etapa del ciclo PDCA e iteración de la metodología de investigación.

Iteración de Investigación	**Etapa Ciclo PDCA**	**Fase de la Metodología de Aplicación del Modelo de Calidad**		**Roles Participantes**		
				Analista de Calidad de Datos	**Responsable del Experimento**	**Responsable del Modelo de Calidad**
Primera iteración				No	No	Sí
Segunda iteración	Plan (Planificar)			No	No	Sí
	Do (Hacer)	1	Generar conocimiento del experimento	Participa en todas las fases	Sí	No
		2	Planificar la evaluación de la calidad de los datos		Validación	No
		3	Evaluar la calidad de los datos		No	No
		4	Ejecutar acciones correctivas sobre los datos		Validación	No
	Check (Verificar)			No	No	Sí
	Act (Actuar)			No	No	Sí

Tabla 5: Roles participantes por Fase de la metodología de aplicación

Los roles que participan durante la metodología de aplicación son los siguientes:

- *Analista en Calidad de Datos*. Debe participar durante todo el proceso ya que es el principal responsable de ejecutar las actividades referentes a la calidad de datos. No es necesario contar con un experto en calidad de datos para poder llevar a cabo este trabajo. A partir del modelo de calidad de datos, que contiene la descripción de cada una de las métricas de calidad, alguien ajeno a esta temática (inclusive el mismo experimentador) podría también llevar adelante estas actividades.

- *Responsable del Experimento*. El involucramiento del responsable del experimento en ciertas etapas del proceso es fundamental, ya que es quien tiene el conocimiento necesario. Durante las validaciones debe asegurar que las métricas y mediciones que se definan sean consistentes con la realidad, e identificar si los problemas encontrados corresponden a errores reales en los datos. También identifica otras métricas o reglas que el analista de calidad puede haber omitido por falta de conocimiento de la realidad. Finalmente, es quien deberá colaborar con el analista de calidad en identificar las posibles correcciones o limpiezas que se requiera realizar sobre los datos erróneos.

6.2.6 ¿Cuándo se aplica la metodología?

Como se muestra en la Ilustración 10, la aplicación tanto de la metodología de trabajo como del modelo de calidad de datos se propone que sea llevado a cabo luego de la ejecución del experimento, y de forma previa al análisis de resultados. Esto se debe principalmente a que el principal objetivo del análisis y evaluación de la calidad de los datos es que se puedan obtener los resultados del experimento basado en datos "limpios" o sin errores, de forma de reflejar de manera más fiel la realidad bajo estudio.

Sin embargo, alguna de las fases de la metodología podrían comenzar a ejecutarse antes de la operación del experimento. Tanto las fases 1 y 2 pueden ser llevadas a cabo luego de la planificación del experimento, en cuanto se disponga de las entradas requeridas por cada una de las fases.

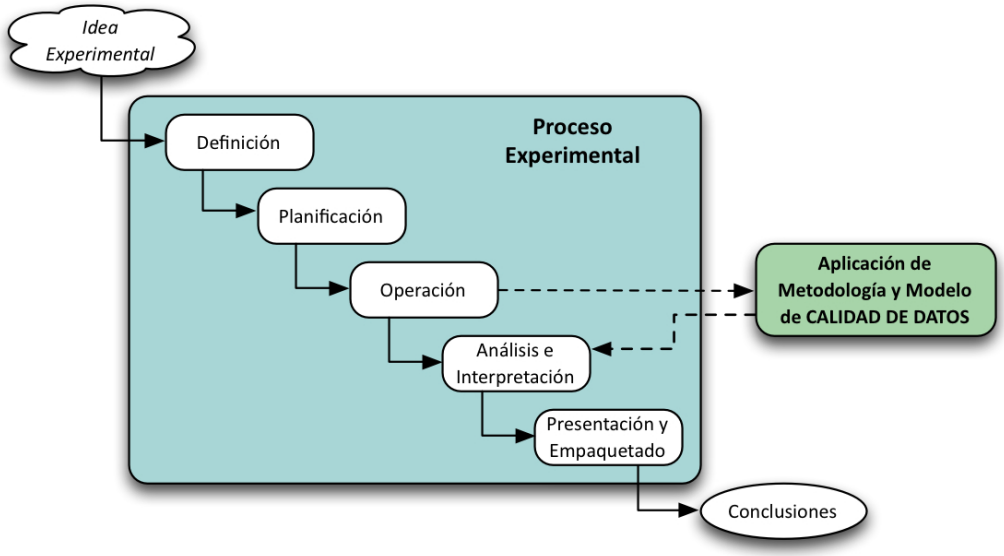

Ilustración 10: Aplicación de la Metodología en el Proceso Experimental

Capítulo 7: Aplicación de la Metodología y Modelo de Calidad sobre los Datos de Experimentos de Métodos de Desarrollo

En este capítulo se presenta cómo se aplicó la metodología de trabajo y el modelo de calidad propuesto sobre los datos de dos experimentos ejecutados por el Centro de Investigación en Métodos de Producción de Software (PROS) [89] de la Universitat Politècnica de València (UPV). Se describe cómo se ejecutó cada fase de la metodología propuesta y los resultados obtenidos.

Los experimentos seleccionados tienen el mismo diseño experimental, con algunas variaciones que se detallarán en este capítulo. Decimos experimento "base" para referirnos a la primera experiencia, y "replicación" para referirnos a la segunda.

7.1 Fase 1: Generar conocimiento del experimento

De forma de generar el conocimiento necesario del experimento base, se llevaron a cabo dos reuniones de trabajo entre el analista de calidad de datos y el responsable del experimento. También se tuvo en cuenta el material enviado por el experimentador (artículos y diagramas) [90], [91], y se realizaron algunas consultas puntuales vía mail. Para el caso de la replicación, solo fue necesario realizar una reunión de trabajo para comprender las diferencias que existen entre ambas ejecuciones, sobre todo respecto a los nuevos datos que se registran.

Al relevar la información se obtiene conocimiento en el diseño del experimento sobre el cual se analizará la calidad de sus datos. A continuación se presenta la información recolectada de forma resumida.

7.1.1 Diseño experimental

El objetivo de este experimento consistió en comparar el paradigma *Model Driven Development* (MDD) [90] con métodos de desarrollo de software tradicional. Su propósito estaba en cubrir la carencia de evidencias empíricas existente sobre las ventajas y desventajas de MDD. Se buscaba identificar las diferencias que pueden apreciar los desarrolladores cuando construyen un sistema desde el inicio, y conocer de qué forma MDD mejora con respecto a los métodos de desarrollo de software tradicional.

La Ilustración 11 muestra gráficamente cómo se aplican los principales conceptos del diseño de experimentos (presentados en el Capítulo 3) en este caso particular.

El experimento base se desarrolló en el año 2012 en el marco de un curso de MDD. Los participantes involucrados en el estudio fueron 26 estudiantes de maestría de la Universidad Politécnica de Valencia (UPV), que habían participado en dos cursos de Ingeniería de Software previamente. Todos tenían conocimientos previos sobre el paradigma de orientación a objetos, pero solo 3 de ellos conocían MDD. La replicación se desarrolló en el año 2013 bajo el mismo contexto. Participaron 19 estudiantes, conformando un total de 10 parejas.

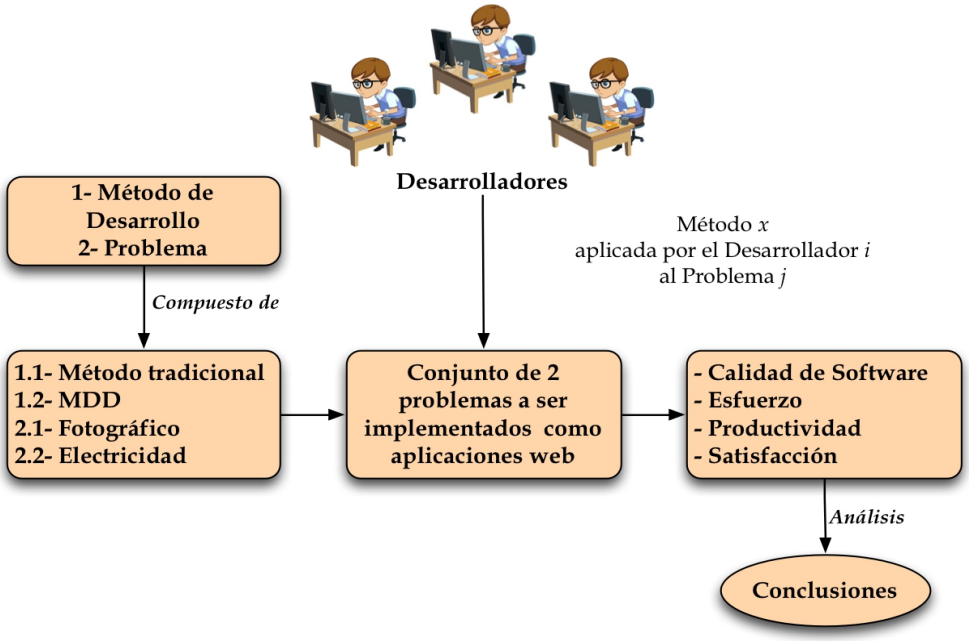

Ilustración 11: Diseño experimental MDD

Las unidades experimentales correspondieron a dos problemas que debían ser implementados como aplicaciones web, creadas específicamente para este experimento. Los sujetos disponían de 4 horas (divididas en 2 sesiones de 2 horas cada una) para desarrollar cada caso.

Se definieron dos factores para el experimento: el método de desarrollo y el problema. El primero se utilizó para verificar si el método de desarrollo tiene algún impacto sobre el producto y el proceso. Se definieron dos alternativas posibles para este factor.

- *Método tradicional*: implementación manual del código, producido por los desarrolladores (utilizando o no modelos conceptuales). No se genera código de forma automática.
- *MDD*: construcción de modelo conceptual transformado automáticamente en código. El método seleccionado en este caso es *OO-Method* [91], el cual es soportado por la herramienta INTEGRANOVA [92]. El desarrollador enfoca su esfuerzo en la construcción de modelos conceptuales y no en la construcción del código.

Para el factor "problema" existían también dos alternativas posibles. El Problema 1 (Fotográfico) es un sistema para gestionar una agencia de fotografía, y el Problema 2 (Electricidad) es un sistema de gestión de solicitudes de reparaciones en un compañía eléctrica. Ambos problemas son de complejidad similar. Este factor se definió para verificar si el problema afectaba las variables de respuesta.

En la replicación, sin embargo, se decidió extender el alcance de los problemas planteados. Esto se debe a que durante el experimento base se observó que algunos estudiantes finalizaban la tarea asignada antes de tiempo. Cada problema contenía entonces tres ejercicios, de forma tal que el ejercicio 1 era equivalente al problema planteado en el caso base. Se incluyeron dos nuevos ejercicios que se implementaban si el tiempo era suficiente. Los datos de estos dos ejercicios no fueron considerados para los análisis estadísticos.

Las variables de respuesta del experimento son las siguientes:

- *Calidad de software*: se calcula como el porcentaje de casos de prueba que han sido exitosos sobre las aplicaciones web construidas por los sujetos.
  Cada caso de prueba era definido por los experimentadores como una secuencia de pasos (también denominados "ítems"). Un caso de prueba se consideraba exitoso si todos sus pasos lo eran. El resultado del caso de prueba así como de sus pasos era siempre booleano (1 si es exitoso, 0 si falla).
- *Esfuerzo*: es el tiempo dedicado para desarrollar la aplicación web desde el inicio.
- *Productividad*: se calcula como el ratio entre calidad de software y esfuerzo (calidad/esfuerzo).
- *Satisfacción*: para medir la satisfacción (experiencia del usuario) se utiliza como instrumento un cuestionario de satisfacción aplicando el framework propuesto por Moody [93], [94].
  La satisfacción era medida en términos de tres variables: usabilidad percibida (PU), facilidad de uso percibida (PEOU), e intención de uso (ITU). Se diseñó un cuestionario que es adaptado para cada tipo de problema (MDD y tradicional), en el que se definen ocho preguntas para medir PU, 6 para PEOU y 2 para ITU.

La Ilustración 12 muestra los principales pasos que conformaron la ejecución de los experimentos. Cada uno se ejecutó a lo largo de 16 sesiones de 2 horas de duración (1 sesión por semana). Al comienzo los estudiantes completaron un cuestionario demográfico de forma de que los responsables conocieran su experiencia previa, contexto y antecedentes.

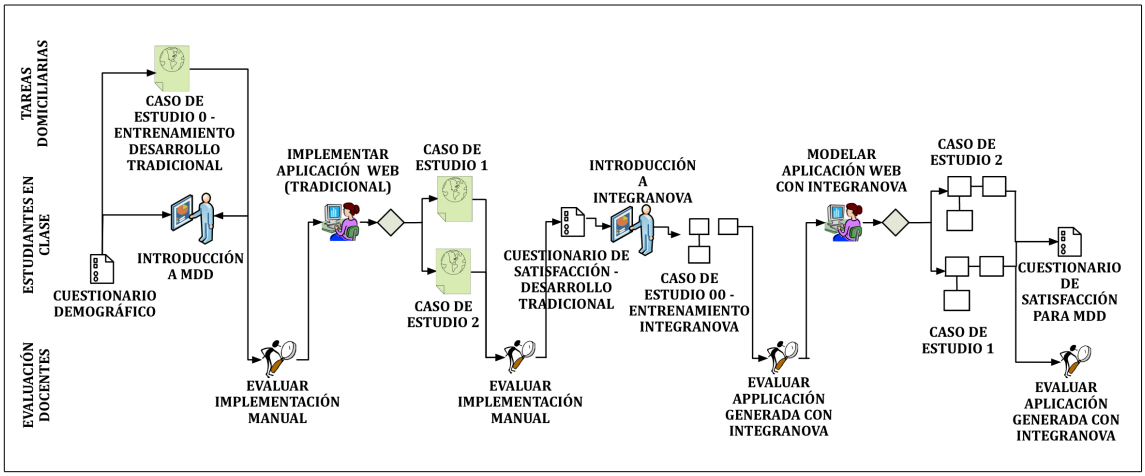

Ilustración 12: Ejecución del Experimento Base MDD – UPV

Luego se llevan a cabo las actividades experimentales para el método de desarrollo tradicional. Se realizó un primer ejercicio de entrenamiento con el objetivo de que los estudiantes adquieran práctica en un lenguaje de programación conocido. Los sujetos trabajaban en pares (equipos de dos personas). Este ejercicio fue corregido por los profesores de forma de asegurar que los estudiantes estén preparados para realizar la próxima actividad experimental.

A continuación las parejas se dividieron en dos grupos. A los pares del mismo grupo se les asignó un único problema que implementan con un método de desarrollo tradicional, y utilizando el mismo lenguaje de programación que para el ejercicio de entrenamiento. Esto se realizó en las siguientes 2 sesiones. Al finalizar el ejercicio, se evaluó la aplicación web desarrollada mediante la ejecución de casos de prueba. Finalmente los estudiantes completaron el cuestionario de satisfacción, que refiere a su experiencia con el método de implementación manual utilizado.

Luego se repitió el mismo proceso utilizando MDD como método de desarrollo y la herramienta INTEGRANOVA. Los grupos fueron intercambiados, de forma que cada pareja tenía asignado un problema para ser implementado con MDD diferente al utilizado para el método de desarrollo tradicional.

Durante la replicación se incluyó además un nuevo cuestionario al finalizar el curso. El objetivo era conocer las principales dificultades con los que se enfrentan los estudiantes al utilizar la herramienta INTEGRANOVA, independientemente del paradigma MDD.

7.1.2 Datos recolectados y almacenados

Los datos recolectados durante la ejecución de los experimentos se registraron en planillas de cálculo. Las planillas conservaban el mismo formato en ambas ejecuciones, con la diferencia que durante la replicación se incluyeron nuevas columnas y filas para registrar los datos que fueron agregados. A continuación se describe el contenido de cada una de las planillas, así como los datos recolectados y calculados que resultaron significativos para los análisis estadísticos, ya que sobre estos se analizará su calidad. La identificación de estos datos se realizó en una reunión de trabajo en conjunto con el experimentador.

- *Evaluación 2012.* Esta planilla contiene datos registrados por los responsables del experimento que permiten calcular las variables de respuesta calidad de software, esfuerzo y productividad. Por cada problema, cada método de desarrollo y cada pareja de sujetos, se registraron los siguientes valores de los datos:
  - Para la variable *esfuerzo*: hora de inicio (comienzo de clase) y fin (hora en que los estudiantes suben la aplicación web al repositorio) de cada una de las sesiones (1 y 2).
    Se calculó el tiempo de cada una de las sesiones (1 y 2) como la diferencia entre el tiempo de fin e inicio de cada sesión; y el tiempo total como la suma de los tiempos de ambas sesiones.
    Durante la replicación se solicitó a los estudiantes que registren además el tiempo dedicado a las tareas de instalación y compilación de sus programas. En el experimento base se observó que el esfuerzo que los sujetos dedicaban a las mismas podría ser significativo en relación al tiempo total. Para la implementación manual, el tiempo registrado correspondía al esfuerzo en instalaciones y configuraciones iniciales que deben realizarse antes de comenzar a programar (se registraba por única vez). Por otro lado, para la implementación MDD el tiempo correspondía al esfuerzo dedicado en las sucesivas instalaciones (compilaciones) de la herramienta INTEGRANOVA (se registraba por cada sesión).
  - Para la variable *calidad de software*: cada hoja contiene los datos correspondientes a cada problema y cada implementación (Fotográfico-Tradicional, Electricidad-Tradicional, Fotográfico-MDD, Electricidad-MDD). Se registró la descripción de los casos de pruebas que se ejecutan sobre las aplicaciones construidas así como la secuencia de pasos que los conforman. En la replicación se incluyen nuevos casos de prueba para probar las funcionalidades agregadas (ejercicios 2 y 3).
    Los responsables registraron el resultado obtenido a partir de la ejecución de los casos de prueba y de cada uno de sus pasos. Se registró un 0 si falló y un 1 si pasó.
    Se calculó el porcentaje de casos de prueba que son exitosos por pareja.

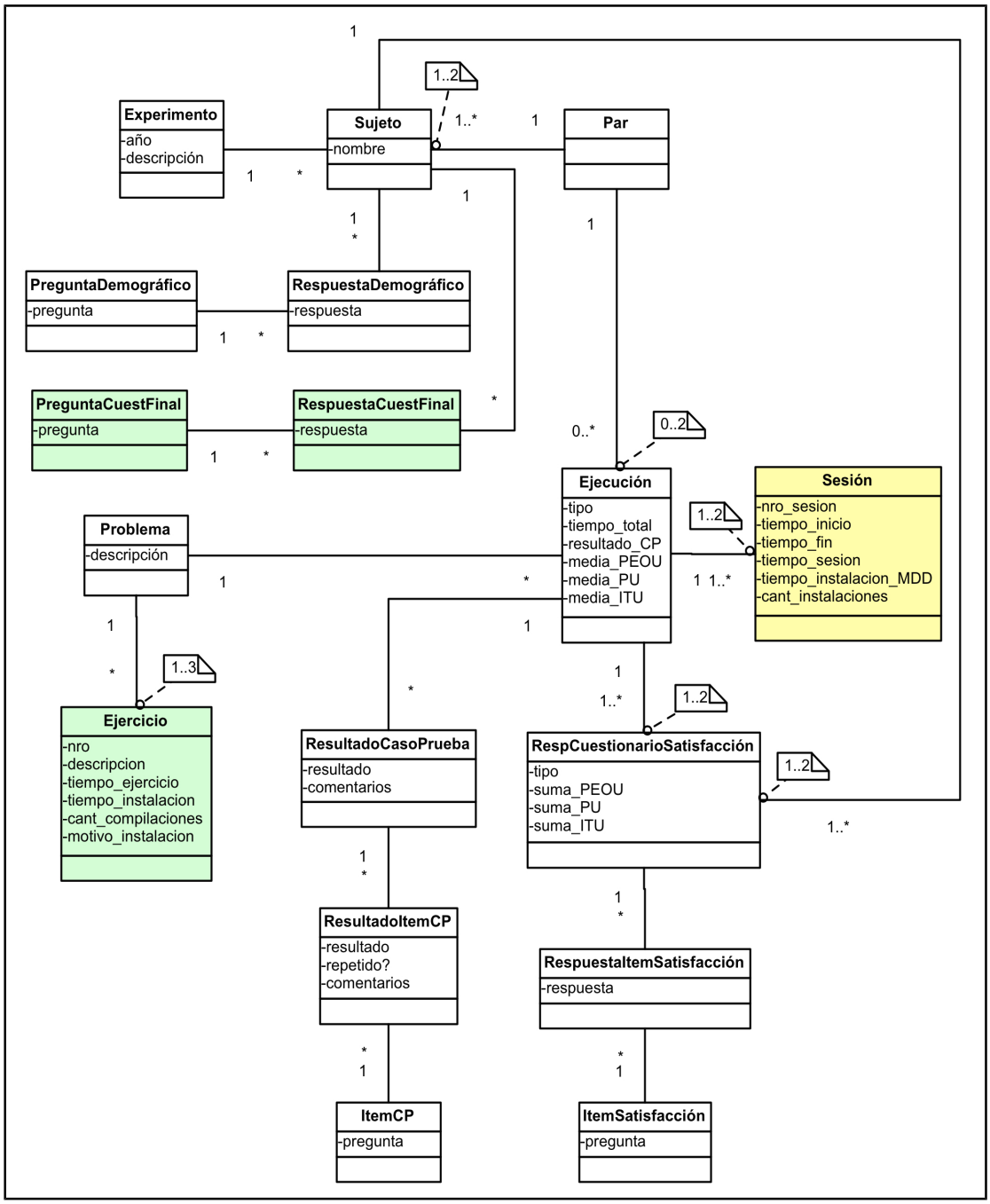

Ilustración 13: Diagrama de Clases para los Experimentos de MDD

Para cada cuestionario, los sujetos del experimento base registraron sus respuestas y sus datos identificatorios en un formulario web diseñados para tal fin. Estos datos fueron luego exportados en planillas de cálculo. Sin embargo, en la replicación los sujetos ingresaban las respuestas a los cuestionarios de forma manual en hojas de papel, las cuales fueron transcriptas luego por el responsable del experimento en las planillas correspondientes.

- *Cuestionario_ Demográfico.* Contiene las 11 preguntas que conforman el cuestionario demográfico y las respuestas (en formato texto libre y múltiple opción) ingresadas por los sujetos.
- *Cuestionario_ MAM_ tradicional.* Contiene las 16 preguntas del cuestionario de satisfacción y las respuestas (en formato múltiple opción) ingresadas por los sujetos, que muestran su experiencia utilizando un método de desarrollo tradicional.
- *Cuestionario_ MAM_ MDD.* Contiene las 16 preguntas del cuestionario de satisfacción y las respuestas (en formato múltiple opción) ingresadas por los sujetos, que muestran su experiencia utilizando MDD.
- *Cuestionario_ Final.* Contiene las 14 preguntas que conformaban el cuestionario final y las respuestas (en formato múltiple opción y texto libre) ingresadas por los sujetos. Aplica solamente para el caso de replicación.

En las dos planillas "MAM" se encuentran los datos a partir de los cuales se calcularon las tres variables de respuesta correspondientes a la satisfacción por estudiante y por pareja (Media PEOU, Media PU, Media ITU). El valor de cada media por estudiante se calculó como la suma de las respuestas al cuestionario de satisfacción, para las preguntas que corresponden a la variable en cuestión, mientras que el valor de cada media por pareja era igual a la suma de las medias de los estudiantes que la conforman.

En este experimento contamos tanto con datos cuantitativos objetivos (como ser los datos de esfuerzo y calidad de software), como cuantitativos subjetivos (respuestas a los cuestionarios de satisfacción), y también datos cualitativos subjetivos (respuestas al cuestionario demográfico).

Con el fin de lograr un mejor entendimiento de la realidad del experimento así como de los datos utilizados y registrados, se generó el diagrama de clases correspondiente que se muestran en la Ilustración 13. Los objetos en verde son aquellos que aplican para el caso de la replicación pero no para el base, y los objetos en amarillo son los que contienen datos particulares utilizados solamente en la replicación.

7.2 Fase 2: Instanciar el modelo de calidad de datos

Durante esta fase en el experimento base, se llevaron a cabo dos de las tres estrategias propuestas en la metodología: *bottom-up* y *top-down*. A partir de los datos que resultaron significativos para los análisis, se elaboró una lista primaria de los posibles errores que pueden tener los mismos. En este caso, los datos significativos son aquellos utilizados para calcular las variables de respuesta del experimento: tiempo invertido por los sujetos, resultados de la ejecución de casos de prueba, y respuestas a los cuestionarios de satisfacción. Luego se mapearon los posibles errores con las métricas de calidad que conforman el modelo. A continuación, se evaluó métrica por métrica del modelo para identificar otros posibles errores que podrían presentarse en el experimento bajo estudio. Por último se consolidó en una única lista los posibles problemas de calidad sobre los datos. Esto se muestra en las Tablas 6 y 7. Debido a que no se utilizan herramientas para el registro de datos, no se implementó la tercera estrategia propuesta.

Se validó con el responsable del experimento la lista de los posibles problemas de calidad sobre los datos, descartando aquellos que no serán considerados por no ser posible su ocurrencia, e incluyendo nuevos que identifique el responsable a partir de su experiencia. Esto se realizó durante una reunión de trabajo entre el analista de calidad y el responsable del experimento.

Durante la replicación, se analizaron nuevamente cada una de las métricas de calidad que conforman el modelo, con el objetivo de identificar si existen diferencias con respecto las métricas que fueron aplicadas para el experimento base (debido a los nuevos datos que comienzan a registrarse). Como resultado del análisis, se encuentra que las métricas a aplicar son las mismas en ambos casos.

# Mét	Métrica de Calidad	# Med	Objetos (tablas y atributos)	Valor de Calidad	Agregación	Método de medición	Problema de Calidad	Cant. datos con problema
M1	Valor fuera de rango	B1.1	Sesión.tiempo_sesion (nro_sesion = 1)	0,913	0,968	Automático (fórmulas de cálculo)	Valor Sospechoso	7
		B1.2	Sesión.tiempo_sesion (nro_sesion = 2)	0,990			Valor Sospechoso	
		B1.3	Ejecución.tiempo_total	1,000				
M2	Falta de estandarización	B2.1	Sesión.tiempo_sesion	1,000	0,941	Automático (fórmulas de cálculo)		8
		B2.2	Ejecución.tiempo_total	1,000				
		B2.3	ResultadoItemCP.resultado ResultadoCasoPrueba.resultado	1,000				
		B2.4	RespuestaItemSatisfaccion.respuesta	1,000				
		B2.5	RespuestaItemSatisfaccion.respuesta	0,706			Error en los datos	
M3	Valor embebido	B3.1	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,615	0,692	Manual (revisión de textos)	Error en los datos	16
		B3.2	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,769			Error en los datos	
M5	Registro con errores	B5.1	Sesión.tiempo_sesion (nro_sesion = 1) Sesión.tiempo_sesion (nro_sesion = 2)	N/E	1,000	Manual (comparación de valores registrados y reales)		
		B5.2	ResultadoItemCP.resultado	N/E				
		B5.3	RespCuestionarioSatisfacción.suma_PEOU	1,000				
		B5.4	RespCuestionarioSatisfacción.suma_PU	1,000				
		B5.5	RespCuestionarioSatisfacción.suma_ITU	1,000				
M7	Falta de precisión	B7.1	Ejecución.resultado_CP	1,000	1,000	Manual (revisión de fórmulas)		
M8	Valor nulo	B8.1	Sesión.tiempo_sesion (nro_sesion = 1)	1,000	1,000	Automático (fórmulas de cálculo)		
		B8.2	Ejecución.tiempo_total	1,000				
		B8.3	ResultadoItemCP.resultado ResultadoCasoPrueba.resultado	1,000				
		B8.4	Ejecución.resultado_CP	1,000				

		B8.5	RespCuestionarioSatisfacción.suma_PEOU Ejecución.media_PEOU	1,000				
		B8.6	RespCuestionarioSatisfacción.suma_PU Ejecución.media_PU	1,000				
		B8.7	RespCuestionarioSatisfacción.suma_ITU Ejecución.media_ITU	1,000				
M9	Información omitida	B9.1	Sesión.tiempo_sesion	1,000	1,000	Automático (fórmulas de cálculo)		
M10	Registro faltante	B10.1	Par Ejecución Problema	1,000	0,996	Automático (fórmulas de cálculo)		1
		B10.2	Par Ejecución ResultadoCasoPrueba	1,000				
		B10.3	Ejecución RespCuestionarioSatisfacción RespuestaDemográfico	0,987			Error en los datos	
M11	Regla de integridad de dominio	B11.1	Ejecución.resultado_CP	1,000	1,000	Automático (fórmulas de cálculo)		
		B11.2	RespCuestionarioSatisfacción.suma_PEOU	1,000				
		B11.3	RespCuestionarioSatisfacción.suma_PU	1,000				
		B11.4	RespCuestionarioSatisfacción.suma_ITU	1,000				
		B11.5	Ejecución.media_PEOU	1,000				
		B11.6	Ejecución.media_PU	1,000				
		B11.7	Ejecución.media_ITU	1,000				
M12	Regla de integridad intra-relación	B12.1	Sesión.tiempo_sesion	0,971	0,971	Automático (fórmulas de cálculo)	Valor Sospechoso	3
M14	Regla de inte-	B14.1	ResultadoItemCP.resultado	0,978	0,857	Automático (fórmulas	Error en los datos	20

	gridad inter-relación		ResultadoItemCP.repetido ResultadoCasoPrueba.resultado			de cálculo) y Manual (revisión de valores registrados)		
		B14.2	Sesión.tiempo_sesion Ejecución.tiempoTotal	1,000				
		B14.3	ResultadoItemCP.resultado ResultadoItemCP.repetido ResultadoCasoPrueba.resultado	1,000				
		B14.4	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,538			Error en los datos	
		B14.5	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	N/E				
		B14.6	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,769			Error en los datos	
M16	Registro duplicado	B16.1	Par, Ejecución, Problema	1,000	1,000	Automático (fórmulas de cálculo)		
		B16.2	Sujeto, Par, Ejecución, RespCuestionarioSatisfacción	1,000				
M18	Estructura de datos	B18.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión planillas de datos)	Oportunidad de mejora	
M19	Formato de datos	B19.1	Conjunto de todos los datos	Bueno	Bueno	Manual (revisión planillas de datos)		
M20	Facilidad de entendimiento	B20.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión planillas de datos)	Oportunidad de mejora	
M21	Metadata	B21.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión planillas de datos)	Oportunidad de mejora	

Tabla 6: Resultado de la aplicación de métricas de calidad sobre los datos del experimento Base MDD

# Mét	Métrica de Calidad	# Med	Objetos (tablas y atributos)	Valor de Calidad	Agregación	Método de Medición	Problema de Calidad	Corrección	Cant. datos con problema
M1	Valor fuera de rango	C1.1	Sesión.tiempo_sesion (nro_sesion = 1)	1,000	0,988	Automático (fórmulas de cálculo)			5
		C1.2	Sesión.tiempo_sesion (nro_sesion = 2)	0,975			Valor Sospechoso	N/A	
		C1.3	Ejecución.tiempo_total	1,000					
		C1.4	Ejercicio.tiempo_ejercicio	1,000					
		C1.5	Ejercicio.cant_compilaciones	1,000					
		C1.6	Sesión.cant_instalaciones	1,000					
		C1.7	Ejercicio.tiempo_instalacion	0,950			Valor Sospechoso	N/A	
		C1.8	Sesión.tiempo_instalacion Sesión.cant_instalaciones	0,975			Valor Sospechoso	N/A	
M2	Falta de estandarización	C2.1	Sesión.tiempo_sesion	1,000	1,000	Automático (fórmulas de cálculo)			
		C2.2	Ejecución.tiempo_total	1,000					
		C2.3	ResultadoItemCP.resultado CasoPrueba.resultado	1,000					
		C2.4	RespuestaItemSatisfaccion.respuesta	1,000					
		C2.5	RespuestaItemSatisfaccion.respuesta	1,000					
		C2.6	Ejercicio.tiempo_ejercicio	1,000					
		C2.7	Sesión.tiempo_instalacion Ejercicio.tiempo_instalacion	1,000					
M3	Valor embebido	C3.1	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	1,000	1,000	Manual (revisión de textos)			
		C3.2	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	1,000					
M5	Registro con errores	C5.1	Ejecución.tiempo_sesion1 Ejecución.tiempo_sesion2	N/E	1,000	Manual (comparación de valores registrados y reales)			
		C5.2	ResultadoItemCP.resultado	N/E					
		C5.3	CuestionarioSatisfacción.suma_PEOU	1,000					

		C5.4	CuestionarioSatisfacción.suma_PU	1,000					
		C5.5	CuestionarioSatisfacción.suma_ITU	1,000					
M7	Falta de precisión	C7.1	Ejecución.resultado_CP	1,000	1,000	Manual (revisión de fórmulas)			
M8	Valor nulo	C8.1	Sesión.tiempo_sesion (nro_sesion = 1)	1,000	0,959	Automático (fórmulas de cálculo)			9
		C8.2	Ejecución.tiempo_total	1,000					
		C8.3	ResultadoItemCP.resultado CasoPrueba.resultado	0,825			Error en los datos	N/A	
		C8.4	Ejecución.resultado_CP	1,000					
		C8.5	CuestionarioSatisfacción.suma_PEOU Ejecución.media_PEOU	1,000					
		C8.6	CuestionarioSatisfacción.suma_PU Ejecución.media_PU	1,000					
		C8.6	CuestionarioSatisfacción.suma_ITU Ejecución.media_ITU	1,000					
		C8.7	Ejercicio.tiempo_ejercicio	0,850			Error en los datos	N/A	
M9	Información omitida	C9.1	Sesión.tiempo_sesion	0,900	0,800	Automático (fórmulas de cálculo)	Error en los datos	Limpieza	9
		C9.2	Ejercicio.tiempo_instalacion Ejercicio.motivo_instalacion Ejercicio.cant_compilaciones	0,500			Error en los datos	N/A	
		C9.3	Sesion.tiempo_instalacion Sesión.cant_instalaciones	1,000					
M10	Registro faltante	C10.1	Par Ejecución Problema	1,000	0,933	Automático (fórmulas de cálculo)			4
		C10.2	Par Ejecución CasoPrueba	0,800			Error en los datos	N/A	

		C10.3	Ejecución CuestionarioSatisfacción RespuestaDemográfico CuestionarioFinal	1,000					
M11	Regla de integridad de dominio	C11.1	Ejecución.resultado_CP	1,000	1,000	Automático (fórmulas de cálculo)			
		C11.2	CuestionarioSatisfacción.suma_PEOU	1,000					
		C11.3	CuestionarioSatisfacción.suma_PU	1,000					
		C11.4	CuestionarioSatisfacción.suma_ITU	1,000					
		C11.5	Ejecución.media_PEOU	1,000					
		C11.6	Ejecución.media_PU	1,000					
		C11.7	Ejecución.media_ITU	1,000					
M12	Regla de integridad intra-relación	C12.1	Sesión.tiempo_sesion	0,963	0,963	Automático (fórmulas de cálculo)	Valor Sospechoso	N/A	3
M14	Regla de integridad inter-relación	C14.1	ResultadoItemCP.resultado ResultadoItemCP.repetido CasoPrueba.resultado	0,971	0,926	Automático (fórmulas de cálculo) y Manual (revisión de valores registrados)	Error en los datos	Limpieza	13
		C14.2	Sesión.tiempo_sesion Ejecución.tiempoTotal	1,000					
		C14.3	ResultadoItemCP.resultado ResultadoItemCP.repetido CasoPrueba.resultado	1,000					
		C14.4	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	1,000					
		C14.5	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,895			Error en los datos	N/A	
		C14.6	RespuestaDemográfico.respuesta PreguntaDemográfico.pregunta	0,842			Error en los datos	N/A	

		C14.7	Ejecución.tiempo_total Ejercicio.tiempo_ejercicio	0,700			Error en los datos	Limpieza	
		C14.8	Ejercicio.tiempo_ejercicio Ejercicio.tiempo_instalacion Sesion.tiempo_instalacion	1,000					
M16	Registro duplicado	C16.1	Par, Ejecución, Problema	1,000	1,000	Automático (fórmulas de cálculo)			
		C16.2	Sujeto, Par, Ejecución, CuestionarioSatisfacción	1,000					
M18	Estructura de datos	C18.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión planillas de cálculo)	Oportunidad de mejora		
M19	Formato de datos	C19.1	Conjunto de todos los datos	Regular	Regular				
M20	Facilidad de entendimiento	C20.1	Conjunto de todos los datos	Regular	Regular				
M21	Metadata	C21.1	Conjunto de todos los datos	Regular	Regular				

Tabla 7: Resultado de la aplicación de métricas de calidad sobre los datos del experimento Replicación MDD

N/E	No Ejecutada
	Diferencias en métricas instanciadas entre experimento base y replicación
	Presenta un problema de calidad (resultado menor a 1,00)
	No presenta un problema de calidad (resultado igual a 1,00)

Tabla 8: Referencias Tablas 6, 7 y 15

Las tablas 6 y 7 muestran cuáles son las métricas de calidad que se aplicaron sobre los datos significativos de cada experimento (base y replicación respectivamente). Para cada métrica de calidad de datos considerada, se definieron los objetos (tablas y atributos) sobre los cuales se aplicarán, y las mediciones correspondientes. La información detallada de las métricas instanciadas que fueron definidas se encuentra en el Anexo C.

De las 21 métricas de calidad que forman parte del Modelo de Calidad de Datos para Experimentos en Ingeniería de Software, las mismas 16 fueron aplicadas sobre los datos de ambos experimentos. Dichas métricas fueron aplicadas sobre 47 objetos particulares del dominio (ya sean celdas, tuplas o el conjunto de datos en su totalidad) en el experimento base, y sobre 59 en la replicación. Las restantes 5 métricas no resultaron aplicables debido a las características de los datos y del contexto bajo estudio. Por ejemplo, al considerar la métrica "Valor fuera de referencial" no se identificó ningún dato que debiera pertenecer a un referencial determinado; o para "Registro contradictorio" no se identificaron casos en los cuales pudieran ocurrir contradicciones entre los datos.

De forma de registrar los resultados de las mediciones de calidad de datos se incluyeron nuevas columnas en las planillas de cálculo que contienen los datos del experimento. Las celdas que presentaban algún problema de calidad se marcaban con un '0' en color rojo, mientras que la celdas que se encontraban libres de errores se marcaban con un '1' en color verde. Cada columna se identificaba con el número de la medición para la cual se estaban registrando los resultados.

Se incluyó también en cada planilla de cálculo una nueva hoja que contiene los resultados de las mediciones, así como otros valores auxiliares que resultaron necesarios para calcular las medidas.

7.2.1 Definición de las métricas de calidad instanciadas

A continuación se describe cómo se aplica cada métrica de calidad a los datos de los experimentos.

La Ilustración 14 muestra gráficamente sobre qué objetos del experimento base se aplican las diferentes métricas de calidad de datos. Como indican las referencias, cada color representa una dimensión de calidad. La aplicación de las métricas de calidad sobre los objetos de la replicación es similar, y no se presenta aquí por motivos de espacio.

Valor fuera de rango

Resulta de interés que los valores de los tiempos (primera sesión, segunda sesión y total) se encuentren dentro de un rango determinado. Estos son ingresados en las planillas de forma manual. En todos los casos, los rangos se establecen a partir de la experiencia (juicio de experto) del experimentador.

Para el caso del tiempo de primera sesión y tiempo total, el rango tiene su mínimo en 01:30hs., considerando que dada la complejidad de los problemas ningún par de sujetos puede haber finalizado la implementación en un tiempo menor. Sin embargo, los estudiantes podrían finalizar su problema sin utilizar el tiempo disponible para la segunda sesión, por lo cual el mínimo del tiempo dedicado a esta se establece en 0.

Para los tiempos de primera y segunda sesión, el máximo es de 02:15hs. La duración de cada sesión es de 2 horas, y se incluye un margen de tolerancia de 15 minutos por si los estudiantes no han finalizado la propuesta. Para el caso del tiempo total, el máximo definido será la suma de las cotas superiores de las dos sesiones (04:30hs). Este rango se define de forma de tener una alerta más sobre los valores que se alejan del rango, y para tener más información sobre lo ocurrido (por ejemplo, en el caso de que un tiempo de sesión se aleje del rango, pero no suceda lo mismo con el tiempo total).

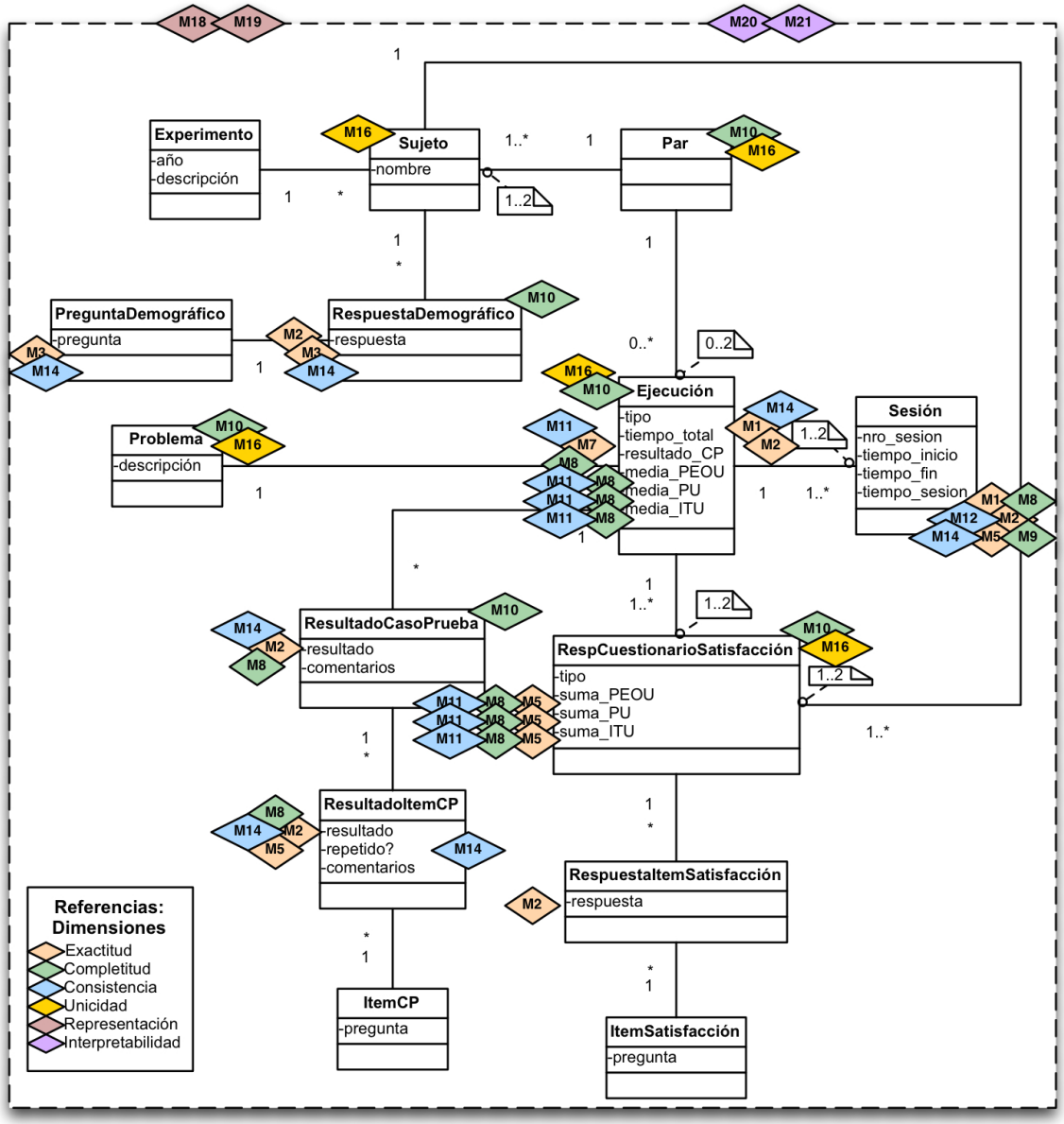

Ilustración 14: Objetos del experimento sobre los cuales se aplican las métricas de calidad de datos

En particular para la replicación, se definieron otras mediciones sobre los nuevos datos que se registran.

Para el caso del tiempo dedicado al ejercicio 1, ya que su alcance coincide con el problema completo planteado para el experimento base, el rango se define igual que para el tiempo total (entre 90 y 270 minutos). No interesa medir el tiempo dedicado a los demás ejercicios, ya que no se toman en cuenta para los análisis estadísticos del experimento.

Para el tiempo de instalación, se define un rango en base a la experiencia del experimentador. En el caso de implementación manual, se establece el mínimo en 0 y como tiempo máximo 30 minutos. Se considera que una pareja no debería dedicar más que este tiempo en las tareas de instalación y configuración inicial. En implementación MDD, se establece un rango en función de la cantidad de

instalaciones que se hayan realizado. Cada instalación debería llevar 5 minutos como mínimo y 30 minutos como máximo. Un valor fuera de este rango podría estar indicando que la pareja tuvo algún inconveniente durante la instalación, o que este tiempo no se registró correctamente.

Se definen rangos también sobre la cantidad de instalaciones y compilaciones. En el caso de implementación manual carecería de sentido que se realice más de una compilación cada 4 minutos, ya que los sujetos estarían demasiado tiempo compilando (y no desarrollando). En implementación MDD el tiempo es de 15 minutos entre compilaciones, al ser más costosas de ejecutar se espera que se realicen con menor frecuencia.

Falta de estandarización

Interesa que los valores de los tiempos de sesiones estén todos registrados en el formato "HH:MM", de forma que sea posible realizar operaciones con los mismos. Para el caso del tiempo total, interesa que sea un entero, calculado como la suma de los tiempos de ambas sesiones (expresado en minutos). En particular para la replicación, interesa también que el tiempo dedicado al ejercicio 1 así como el tiempo de instalación sean valores enteros.

Se identifica además que los resultados de los pasos y casos de prueba deben contener alguno de los valores {0,1} representando si el mismo falló o pasó.

Las respuestas de los sujetos al formulario de satisfacción, al ser en formato múltiple opción, deben contener alguno de los valores {1,2,3,4,5} representando cada una de estos las opciones de respuesta posibles.

Por último, los datos identificatorios de los estudiantes (nombre y apellido) deben estar registrados con la misma sintaxis y formato para todos los cuestionarios, de forma tal de que sea posible identificar qué sujeto respondió a qué cuestionarios. Notar que no existe un identificador asociado al estudiante.

Valor embebido

Los datos demográficos registrados por los sujetos son relevantes para conocer su experiencia, contexto y antecedentes. Sin embargo, muchos de ellos se encuentran embebidos en un texto libre, y por lo tanto deben procesarse caso a caso para poder ser utilizados.

Los datos embebidos son sobre la experiencia (en meses) de los sujetos para cada perfil, y los frameworks para desarrollo web que conocen. Se consideran valores embebidos aquellos que contienen otra información (además de los valores solicitados) en el mismo campo de texto.

Registro con errores

Se busca si existen desviaciones respecto a lo que sucedió realmente en cuanto a los tiempos, los resultados de casos de prueba y en las medias de satisfacción, intentando comparar los valores reales con los registrados.

El tiempo de fin de cada sesión es registrado por los experimentadores observando la hora en que las parejas suben la última versión del programa al repositorio, por lo que podría ocurrir un error al copiar esos datos en la planilla correspondiente.

Por otra parte, interesa que los resultados de los casos de prueba (si falla o pasa) reflejen el estado real de las funcionalidades que fueron implementados por los sujetos. Puede existir una equivocación por parte del experimentador al registrar el resultado {0,1}, o que los criterios utilizados para establecer cuando un caso falla o pasa sean diferentes para distintas parejas.

Por último, interesa que las medias (PU, PEOU, ITU) para medir la satisfacción de los sujetos y parejas hayan sido calculadas considerando las respuestas del cuestionario que corresponden a cada variable.

Falta de precisión

Interesa que el porcentaje obtenido a partir de los resultados de casos de prueba que resultan exitosos por pareja se registren como un entero con dos números decimales, de forma de lograr una mayor precisión en los cálculos y análisis que se realizan a partir de estos valores.

Valor nulo

Interesa conocer si existen valores nulos en el tiempo de la primera sesión y tiempo total. Notar que la pareja puede finalizar el problema durante la primera sesión, por lo que el tiempo de la segunda podría ser nulo. En particular para la replicación, interesa también que el tiempo del ejercicio 1 contenga algún valor no vacío, ya que será utilizado como base para los análisis del experimento.

Por otra parte, interesa que existan los resultados para los pasos y los casos de prueba, de forma de poder calcular a partir de estos el porcentaje de casos de prueba que son exitosos por pareja.

Por último, interesa que las medias (PU, PEOU, ITU) para medir la satisfacción de los sujetos hayan sido calculadas.

Información omitida

En este caso, interesa que si el tiempo de inicio y fin de sesión son no vacíos, entonces el tiempo total por sesión haya sido calculado (y sea también no vacío), tanto para la primera como para la segunda sesión.

En particular para la replicación, para cada ejecución manual el tiempo y motivo de instalación así como la cantidad de compilaciones por ejecución deben ser valores no vacíos. Mientas que en cada ejecución MDD, el tiempo y cantidad de instalaciones por sesión deben contener algún valores diferente de nulo.

Registro faltante

Para cada par de sujetos debe existir un registro de ejecución por cada uno de los dos problemas a implementar; así como un registro de caso de prueba por cada caso definido.

Para cada sujeto debe existir un registro por cada cuestionario respondido (uno para el demográfico, y uno de satisfacción por cada uno de los 2 problemas implementados). En la replicación además debe existir un registro para el cuestionario final por cada sujeto.

Regla de integridad de dominio

Los porcentajes que corresponden a los casos de prueba exitosos por pareja deben ser valores entre 0 y 100.

Por otra parte, se identifica que las variables de satisfacción (PEOU, PU, ITU), dado la forma en que son calculadas, deben pertenecer a un determinado dominio. Esto sucede tanto para las medias calculadas por cada sujeto (PEOU=[6..30], PU=[8..40], ITU=[2..10]) como para cada pareja (PEOU=[12..60], PU=[16..80], ITU=[4..20]).

Regla de integridad intra-relación

Se considera que si una pareja de sujetos dedica menos que el tiempo disponible durante la primera sesión, es porque logró finalizar el problema asignado en este tiempo. En caso contrario, debería haber ocupado todo el tiempo de clase disponible. Se define entonces la regla de integridad que establece que si el tiempo dedicado a la primera sesión es menor a las 2 horas que tienen disponible, entonces durante la segunda sesión no dedican esfuerzo (o sea que el tiempo debería ser igual a 0).

Reglas de integridad inter-relación

Se identifica un conjunto de reglas que deben ser satisfechas sobre los datos de tiempos y casos de prueba. Estas reglas se definen en conjunto con el responsable del experimento a partir del conocimiento de la realidad bajo estudio.

- El tiempo total por pareja es igual a la suma del tiempo dedicado a la primera y a la segunda sesión.
- En la replicación, se debe cumplir que el tiempo registrado para al ejercicio 1 sea menor o igual al tiempo total. El tiempo total es la suma del tiempo dedicado a cada una de las dos sesiones, y por lo tanto durante las mismas se podría trabajar en más de un ejercicio.
- En la replicación, se debe cumplir que el tiempo que dedican a las instalaciones sea menor o igual al tiempo del ejercicio 1. Esta regla verifica que los sujetos no hayan dedicado más tiempo a compilar que a programar.
- Un mismo paso puede ser parte de más de un caso de prueba (se denomina un paso "repetido"). Un paso "repetido" que se considere para calcular el resultado de un caso de prueba, no debe ser considerado para calcular el resultado de otros casos de prueba en los cuales se incluya.
- El resultado de un caso de prueba es 1 (pasa) si y sólo si el resultado de todos los pasos que lo conforman es 1 (pasa), sin considerar los repetidos. En caso contrario, si existe al menos un paso del caso de prueba cuyo resultado sea 0 (falla), entonces el resultado del caso de prueba es 0 (falla).

Con respecto al cuestionario demográfico, existen preguntas que están relacionadas y por lo tanto las respuestas de los sujetos debería ser consistente. Se definen tres reglas:

- Si como respuesta a la pregunta 1 se indica que no se posee experiencia en el desarrollo de software en empresas TIC, entonces la respuesta a la pregunta 2 (que solicita indicar perfil desempeñado y tiempo) debería ser vacía.
- El tiempo que se indica como respuesta a la pregunta 1 (experiencia en el desarrollo de software en empresas TIC) debería ser igual a la suma de los tiempos por perfil que son indicados en la respuesta a la pregunta 2.
- Si como respuesta a la pregunta 10 se indica que no se posee experiencia previa en el desarrollo de aplicaciones Web, entonces la respuesta a la pregunta 11 (frameworks para desarrollo web que dominan) debería ser vacía.

Registro duplicado

A cada par de sujetos se le asigna un problema a resolver (Fotográfico o Electricidad), utilizando un método de desarrollo determinado (manual o MDD). De acuerdo a como fue diseñado el experimento, el criterio de duplicación definido establece que la misma pareja no debe implemente el mismo problema más de una vez con diferentes métodos.

Por otra parte, cada sujeto debe responder un único cuestionario de satisfacción por cada problema a implementar con un método de desarrollo dado.

Estructura de datos

Debido a que no existen restricciones definidas ni documentadas explícitamente sobre los datos, ni son controladas en las planillas de cálculo, las restricciones que son identificadas junto con los experimentadores son incluidas como reglas de consistencia.

Por otra parte, se analiza si la estructura de las planillas de cálculo utilizadas para almacenar los datos resulta adecuada para la realidad bajo estudio.

Formato de datos

Se analiza si se utiliza el mismo formato para representar los mismos datos en las planillas de cálculo que almacenan los datos, así como entre las hojas de una misma planilla.

Facilidad de entendimiento

Para evaluar si los datos son entendibles por alguien ajeno al experimento, el analista de calidad analiza las planillas de cálculo y verifica si comprende el significado de los datos almacenados.

Metadata

Debido a que no se define ni documenta el esquema conceptual para los datos, el mismo fue generado como parte del trabajo de calidad en colaboración con el experimentador, para lograr un mejor entendimiento de la realidad bajo estudio.

Además, se evalúa si las planillas de cálculo contienen metadata e información de trazabilidad.

7.3 Fase 3: Evaluar la calidad de los datos

En las Tablas 6 y 7 se muestra el resultado obtenido luego de la aplicación de cada métrica de calidad sobre los datos de los experimentos (base y replicación respectivamente). A partir de la medida obtenida (valor de calidad), es posible identificar cuáles son los problemas de calidad que están presentes en los datos.

Las mediciones se identifican con la siguiente nomenclatura:
<id_ expe><id_ métrica>'.'<id_ medición>.

Por ejemplo, la medición "Valor Fuera de Rango" sobre el objeto "tiempo_total", se identifica en el caso base como B1.3, y en la replicación como C1.3.

En el experimento base se ejecutan 47 mediciones sobre cada uno de los objetos del dominio definidos en la fase anterior, mientras que en la replicación se ejecutan 59 mediciones. Del total de mediciones ejecutadas:

- El 68% (32 mediciones) se realizan de forma automática en el caso base y 75% (44 mediciones) en la replicación, mediante el uso de fórmulas de cálculo.
- El 26% (12 mediciones) se realizan de forma manual en el caso base y 22% (13 mediciones) en la replicación, por no ser posible o necesaria su automatización. En todos los casos corresponden a revisiones o verificaciones manuales sobre las planillas que contienen los datos. Como parte de estas mediciones, se incluyen las 4 referentes a la Representación e Interpretabilidad de los datos.
- Las mediciones correspondientes al restante 6% (3 mediciones) y 3% (2 mediciones) respectivamente, son aquellas que no es posible su ejecución, ya sea porque corresponden a mediciones manuales con alto costo (esfuerzo) de implementación asociado, o que por su relación costo/beneficio no amerita realizarlas. Esto sucede en ambos experimentos para el caso de Registro con errores (mediciones B-C5.1 y B-C5.2), y en el caso base para Reglas de integridad inter-relación (medición B14.5).

Como resultado se encuentra que la medida obtenida es menor a 1,00 o 'Regular' para 13 de las 47 mediciones ejecutadas en el caso base y para 17 de las 59 en la replicación, indicando la presencia de un problema de calidad en los datos. Estas mediciones corresponden a 9 métricas de calidad diferentes en el primer caso, y a 10 en el segundo. Se analizan los resultados obtenidos y se identifican los datos que contienen algún problema de calidad, clasificándose como sigue.

7.3.1 Errores en los datos identificados en el experimento base

A partir de la ejecución de 7 de las 13 mediciones se identifican datos erróneos. Se detalla cada caso a continuación.

Falta de estandarización: errores sintácticos en nombres y apellidos de los sujetos (medición B2.5)

Se identifican 8 registros de respuestas al cuestionario de satisfacción que contienen nombres y apellidos que no fueron ingresados siguiendo el mismo formato ni sintaxis.

Los casos identificados son los siguientes:

- Nombres y apellidos registrados en celdas intercambiadas (1 registro).
- Inclusión de primer apellido en algunas casos y segundo apellido en otros (3 registros).
- Errores de sintaxis, incluyendo falta de tildes y letras trabucadas (3 registros).
- Hay 1 registro que coincide con el resultado de la medición de "Registro faltante".

Si se requiere identificar las respuestas de un sujeto para ambos cuestionarios, sería necesario llevar a cabo una corrección manual de los datos, o incluir un identificador de sujeto.

Valor embebido: información sobre la experiencia de los sujetos embebida en las respuestas al cuestionario demográfico (mediciones B3.1 y B3.2)

Se identifican 10 registros que contienen datos sobre la experiencia de los sujetos (perfiles en los que han trabajado y tiempo) embebidos en un texto libre. Esto dificulta conocer información como el tiempo de experiencia que tienen los sujetos por cada perfil.

Por otra parte, se identifican 6 registros de respuesta al cuestionario demográfico que contienen datos referentes al conocimiento previo de los sujetos (*frameworks* de desarrollo que dominan) también embebidos en un texto libre. Esto dificulta conocer información como el *framework* para desarrollo de aplicaciones web más conocido por los sujetos. Hay 4 sujetos que han ingresado valores embebidos en ambas respuestas.

Si se requiere manipular los datos de forma automática, sería necesario separar los valores en celdas diferentes. Si la cantidad de sujetos fuese mayor, el esfuerzo asociado al análisis manual de estos datos se vería incrementado de forma significativa.

Reglas de integridad inter-relación: incumplimiento de reglas de consistencia sobre las respuestas al cuestionario demográfico (mediciones B14.4 y B14.6)

Se identifican 12 registros en los cuales los sujetos indican no tener experiencia en el desarrollo software en empresas TIC, y sin embargo ingresan el perfil y/o tiempo en los que trabajaron.

Hay 6 registros de respuestas al cuestionario demográfico en los cuales los sujetos indican nunca haber desarrollado una aplicación web, y sin embargo detallan qué *frameworks* para desarrollo web dominan. Hay 5 sujetos que han ingresado inconsistencias en ambas respuestas.

Un motivo por el cual suceden estas inconsistencias puede ser el hecho de que todas las preguntas son obligatorias, por lo cual no era posible dejar ninguna sin respuesta. Debido a que son pocos sujetos, los experimentadores optaron por incluir preguntas abiertas y obligatorias, con respuestas que contienen texto libre, y luego analizar los datos manualmente.

Se identifican además otros dos casos que fueron tratados por el responsable del experimento de manera previa al análisis de la calidad, estos son:

Registro faltante: inexistencia de respuesta al segundo cuestionario de satisfacción (medición B10.3)

Se identifica que para un estudiante no existe el registro de su respuesta al segundo cuestionario de satisfacción (caso de implementación MDD). Se consultó con el responsable del experimento indicando que el motivo es que el estudiante abandonó el curso en ese momento.

Para calcular las variables de satisfacción de la pareja a la cual pertenecía ese estudiante, los experimentadores consideran dos veces los datos de su compañero (suponiendo que el estudiante que abandonó habría respondido de igual o similar forma que su pareja). No se descartan el resto de los datos que son ingresados por el estudiante que abandona el curso, sino que sus respuestas al cuestionario demográfico y al primer cuestionario de satisfacción (implementación tradicional) son consideradas para los análisis. Los experimentadores asumen que este hecho no tiene impacto en los datos de tiempos ni en los resultados de los casos de prueba de la pareja.

Reglas de integridad inter-relación: incumplimiento de reglas de consistencia asociadas a los resultados de pasos y casos de prueba (medición B14.1)

Se identifican 2 registros de casos de prueba en los cuales existe un paso que falla (su resultado es igual a 0), y sin embargo el caso de prueba es exitoso (su resultado es igual a 1). En ambos casos el paso que falla es "Introducir un id".

Se consulta con el responsable del experimento y se indica que ese paso no fue considerado para calcular el resultado final de los casos de prueba. Por lo tanto, no corresponde a un error en los datos y no requiere ser corregido. Notar que esto no está documentado ni indicado mediante metadata en la planilla de cálculo.

7.3.2 Errores en los datos identificados en la replicación

Como resultado de la ejecución de 9 de las 17 mediciones se identifican datos erróneos. Se detalla cada caso a continuación.

Valor nulo: en los pasos de casos de prueba y tiempo del primer ejercicio (mediciones C8.3 y C8.7)

Se identifican 6 registros con valores nulos en los pasos de los casos de prueba. Se consulta con el responsable del experimento, indicando que corresponden a funcionalidades que no fueron implementadas y por lo tanto tampoco fueron probadas.

Por otra parte, se identifican 3 casos en los cuales existe el tiempo del primer ejercicio en minutos, pero este tiempo no fue registrado en formato HH:MM. Esto hace dudar sobre la forma en que se obtuvo este dato. Se consulta con el responsable del experimento, indicando que en estos casos los equipos no han terminado con la implementación del primer ejercicio, por lo que se registra como tiempo total las 4 horas (240 minutos) de clase completas.

Información omitida: sobre los tiempos de sesión, y sobre los datos de instalaciones y compilaciones (mediciones C9.1 y 9.2)

Se identifican 4 registros (ejecución manual) en los que se ingresa la hora de inicio y fin de la sesión, pero no se calcula el tiempo dedicado en dicha sesión. Esto impacta directamente en el cálculo de la variable de respuesta esfuerzo, ya que el tiempo total para el problema solo considera el tiempo de una sesión y no de ambas. El motivo de la existencia de este error es que se omitió realizar el cálculo del tiempo de la sesión por parte del experimentador, y por lo tanto requiere ser corregido.

Por otra parte, se identifica que para 5 de las 10 parejas se omite el ingreso de algunos de los datos correspondientes a la instalación y compilación (tiempo, motivo y/o cantidad de instalaciones). En principio estos datos no serán utilizados para los análisis del experimento.

Registro faltante: de casos de prueba (medición C10.2)

Se identifican 4 casos para los cuales no existen los registros de resultados de casos de prueba. Todos ellos coinciden con alguno de los casos identificados para la medición C8.3 de Valor Nulo. La omisión de registros de casos de prueba se debe también a que las funcionalidades no fueron implementadas, y por lo tanto no se ejecutan esos casos de prueba.

Regla de integridad inter-relación: incumplimiento de reglas de consistencia sobre los resultados de pasos y casos de prueba, tiempos, y respuestas al cuestionario demográfico (mediciones C14.1, C14.5, C14.6 y C14.7)

Se identifican 13 registros que no satisfacen alguna de las reglas planteadas.

En la medición C14.1, hay 2 casos en los que se registra el resultado del caso de prueba, pero no existen los resultados de los pasos que lo conforman. Esto hace dudar sobre la forma en la que se obtiene el resultado del caso de prueba (sin conocer los resultados de sus pasos). El motivo de la existencia de este error es que las funcionalidades correspondientes a esos casos de prueba no fueron implementados, por lo tanto el resultado debería ser vacío o '0' (dependiendo de la convención que se tome para funcionalidades no implementadas). Este error impacta directamente en el cálculo de la variables de respuesta calidad de software, por lo que debe ser atendido.

Para la medición C14.7, hay 6 pares de sujetos (ejecución manual) para los cuales el tiempo del primer ejercicio es mayor que el tiempo total. Hay 4 de los 6 casos que coinciden con los encontrados en la medición C9.1. Además, para 3 de los 6 casos sucede que, dado que no llegan a finalizar la implementación del ejercicio, se toma como convención el ingreso de la máxima cantidad de minutos de una sesión (240 = 4 horas). Sin embargo, los sujetos registran igualmente la hora de finalización de las sesiones. Mientras que el tiempo total es calculado en base a los datos registrados por los sujetos, el tiempo del ejercicio 1 se registra por convención como 240 minutos, dando lugar a la inconsistencia detectada. Este error impacta directamente en el cálculo de la variables de respuesta esfuerzo, por lo que debe ser atendido.

Las mediciones C14.5 y C14.6 refieren a datos demográficos ingresados por los sujetos. En la primera medición se identifican dos casos en los cuales la cantidad total de meses de experiencia no es igual a la suma de los meses de experiencia considerando los diferentes perfiles. En la segunda medición hay tres casos en los cuales se dice tener experiencia en desarrollo web pero no se indica el framework de desarrollo web conocido; o viceversa.

7.3.3 Valores sospechosos identificados en el experimento base

A partir de la ejecución de 3 mediciones se detecta la presencia de 10 valores sospechosos en los datos, los cuales fueron analizados de manera separada junto con el responsable del experimento. Entre los problemas identificados se incluyen:

Valor fuera de rango: en los tiempos de primera y segunda sesión (mediciones B1.1 y B1.2)

Como resultado de la medición se obtienen 7 registros de sesiones cuyos valores en los tiempos caen fuera del rango considerado como válido: 6 corresponden al tiempo de primera sesión, y 1 al tiempo de segunda sesión. Notar que no se identifican valores fuera de rango en el tiempo total.

Como se observa en la Tabla 9, en 6 de los 7 casos los valores se exceden por menos de 15 minutos del rango definido, lo cual se considera un margen de tolerancia aceptable. Podrían ocurrir retrasos o contratiempos que ocasionen que los sujetos dediquen unos minutos más de clase para finalizar los ejercicios planteados.

Sin embargo, hay un caso que llama la atención ya que el tiempo de la primera sesión excede en casi 2 horas la cota superior. El experimentador supone que los sujetos no podrían asistir a la segunda sesión ya que sólo dedican 13 minutos en esta, por lo cual invierten la mayor parte del esfuerzo durante la primera. No existen notas ni registro sobre lo sucedido. De todas maneras el tiempo total para esta pareja se sitúa dentro del rango considerado válido (04:27).

Id Pareja	Ejecución	Tiempo	Valor (HH:MM)	Diferencia (minutos)
1	Manual	Primera Sesión	02:18	3
2	Manual	Primera Sesión	02:28	13
7	Manual	Primera Sesión	04:14	119
8	Manual	Primera Sesión	02:18	3
8	MDD	Primera Sesión	02:16	1
11	MDD	Primera Sesión	02:16	1
2	MDD	Segunda Sesión	02:20	5

Tabla 9: Tiempos de sesiones fuera de rango

Regla de integridad intra-relación: reglas de consistencia sobre tiempos de sesiones (medición B12.1)

Se obtienen 3 registros de sesiones que no cumplen con la regla especificada.

Como se observa en la Tabla 10, en todos los casos el tiempo de la primera sesión no es más que 15 minutos inferior que la cota planteada, lo cual se considera un margen de tolerancia aceptable. Podría ocurrir que los sujetos finalicen unos minutos antes que las 2 horas disponibles, por ejemplo si el tiempo de compilación fuera mayor al tiempo de clase restante.

Id Pareja	Ejecución	Tiempo 1era. Sesión	Tiempo 2da. Sesión	Diferencia (minutos)
13	Manual	01:49	00:59	11
3	MDD	01:59	01:37	1
7	MDD	01:48	09:49	12

Tabla 10: Tiempos de sesiones que no cumplen con la regla de integridad intra-relación

7.3.4 Valores sospechosos identificados en la replicación

A partir de la ejecución de 4 mediciones se identifica la presencia de 8 valores sospechosos. Entre los problemas identificados se incluyen:

Valor fuera de rango: en los tiempos de segunda sesión y de instalación (mediciones C1.2, C1.7 y C1.8)

Como resultado de la medición se obtienen 5 registros cuyos valores en los tiempos caen fuera de los rangos considerados como válidos.

Hay 2 casos en el tiempo de la segunda sesión y 1 en el tiempo de instalación que se exceden en menos de 15 minutos, lo cual se considera un margen de tolerancia aceptado.

Hay 2 casos para los cuales el tiempo por instalación es menor que el mínimo definido (menos de 1 minuto por cada instalación), lo cual se considera un margen de tolerancia aceptado.

Regla de integridad intra-relación: regla de consistencia sobre tiempos de sesiones (medición C12.1)

Hay 3 casos para los cuales el tiempo de la primera sesión no es más que 10 minutos inferior que la cota planteada, lo cual se considera un margen de tolerancia aceptado.

Se revisan todos los casos de valores sospechosos junto con el responsable del experimento, y no se identifica ninguno que pueda ser corregido.

7.3.5 Oportunidades de mejora

Se identifican propuestas de mejora a considerar sobre las 4 mediciones correspondientes a las métricas de calidad que refieren al conjunto de datos completo.

Estructura de datos (medición B-C18.1)

Los datos se registran en planillas de cálculo, en un formato que se considera adecuado y consistente para la realidad que es representada, considerando la baja cantidad y complejidad de los datos. Sin embargo, la definición e implementación de un esquema de base de datos podría prevenir la ocurrencia de algunos de los problemas de calidad encontrados (tales como valores fuera de rango, no estandarizados o inconsistentes), mediante la definición de reglas y restricciones sobre los datos.

Formato de datos (medición C19.1), aplica solamente para la replicación

En general las planillas utilizadas y las diferentes hojas que las componen conservan el mismo formato para registrar los mismos datos. Sin embargo, hay algunos datos (sobre tiempos, compilaciones) que se registran de forma diferente según sea implementación manual o MDD.

Se recomienda utilizar el mismo formato para representar los mismos datos, de forma de contribuir a la comprensión de los mismos.

Facilidad de entendimiento (medición B-C20.1)

Fue necesario consultar al experimentador para poder interpretar el significado de ciertos datos registrados en las planillas. Algunos ejemplos: no está claramente identificada la lista de sujetos que participaron del experimento (en el caso base uno abandonó el curso pero no está especificado); los pasos de casos de prueba tienen sombras de diferentes colores pero se desconoce su significado.

Se recomienda utilizar referencias para que sea posible facilitar el entendimiento de los datos.

Metadata (medición B-C21.1)

No se registra metadata sobre la historia, origen o trazabilidad de los datos. A modo de ejemplo, no se identifica quién registró cada dato (sujeto o experimentador) ni de qué forma (directamente en las planillas o mediante formulario web).

7.3.6 Fase 4: Ejecutar acciones correctivas sobre los datos

Se analiza en conjunto con el responsable del experimento si para los problemas de calidad identificados es posible aplicar acciones de corrección. El análisis detallado de estos casos se encuentra en el Anexo C.

Para el caso de los errores en los datos presentes en el experimento base, no se identifican posibles correcciones o limpiezas a aplicar. Estos errores suceden sobre datos que no tienen un impacto significativo en los resultados. Corresponden a datos registrados por los estudiantes (respuestas a los cuestionarios demográfico y de satisfacción). El responsable del experimento puede realizar una depuración manual sobre estos datos previo a su análisis sin mayor esfuerzo, debido a la baja cantidad de datos.

Sin embargo, en la replicación se comprueba que 3 de los 9 casos con errores en los datos deben ser corregidos. Los otros 6 casos son revisados con el responsable del experimento y no se identi-

fican formas de corrección posible ya que no conoce la información real. Notar que todos los casos de error corresponden a datos que son registrados o calculados por parte del experimentador, y se detallan a continuación.

Información omitida referente al tiempo de segunda sesión (medición C9.1)

La corrección consiste en realizar los cálculos faltantes. Se actualiza el valor del tiempo de segunda sesión (calculado como la diferencia entre la hora de fin e inicio de la sesión) y del tiempo total (calculado como la suma de tiempos de primera y segunda sesión). Esto se debe implementar de forma manual para las parejas 2, 3, 4 y 5, como se muestra en la Tabla 11.

Id Pareja	Antes corrección		Después corrección	
	Tiempo 2da. Sesión	Tiempo Total	Tiempo 2da. Sesión	Tiempo Total
2	NULL	02:00	02:03	04:03
3	NULL	02:03	01:58	04:01
4	NULL	02:00	02:00	04:00
5	NULL	01:58	02:01	03:59

Tabla 11: Correcciones para la medición C9.1

Regla de integridad inter-relación sobre el resultado de pasos y casos de prueba (medición C14.1)

La forma de corregir este error consiste en sustituir los valores de los resultados de casos de prueba por '0', y realizar nuevamente los cálculos de porcentajes de casos de prueba exitosos.

Como se muestra en la Tabla 12, la corrección aplicada sobre la pareja 10 impacta en el resultado obtenido (ya que su resultado era '1' en vez de '0'), mientras que en la pareja 8 se mantiene igual.

Id Pareja	Caso de Prueba (CP)	Antes corrección		Después corrección	
		Resultado CP	% CP exitosos	Resultado CP	% CP exitosos
10	Aprobar Solicitudes	1	25	0	0
8	Promocionar Fotográfico	0	75	0	75

Tabla 12: Correcciones para la medición C14.1

Regla de integridad inter-relación sobre el tiempo del primer ejercicio (medición C14.7)

Hay 4 de los 6 casos que coinciden con los encontrados en la medición C9.1, por lo tanto son corregidos al realizar la limpieza antes propuesta. Para 3 casos es necesario además sustituir el valor del tiempo del ejercicio 1 (ingresado por convención) por el valor del tiempo total (que refleja de mejor manera la realidad por ser el registrado por los propios sujetos). Esto se ejecuta en forma manual para las parejas 5, 6 y 10, como se muestra en la Tabla 13.

Id Pareja	Después de corrección C9.1		Después corrección C14.7	
	Tiempo 1er Ejercicio	Tiempo Total	Tiempo 1er Ejercicio	Tiempo Total
2	180	243	180	243
3	240	241	240	241
4	150	240	150	240
5	240	239	239	239
6	240	237	237	237
10	240	237	237	237

Tabla 13: Correcciones para la medición C14.7

Para el caso de los valores sospechosos, debido a que no existen fuentes de datos sobre las cuales se pueda comparar el valor registrado con el real, no es posible asegurar la existencia de un error ni aplicar correcciones en ninguna de las dos ejecuciones.

Se proponen actividades de prevención para todos los casos en los que se identifica la presencia de un problema de calidad, que podrían ser incorporadas para futuras replicaciones del experimento para contribuir en la mejora de su calidad.

7.4 Análisis y discusión

En este trabajo se aplica la metodología definida y se instancia el modelo de calidad propuesto en dos ejecuciones de un experimento, persiguiendo dos propósitos específicos. Por un lado, se busca conocer, analizar y mejorar la calidad de los datos del experimento bajo estudio. Por otra lado, se busca evaluar si el modelo y la metodología de calidad de datos propuestos son aplicables a este experimento en particular, y así obtener retroalimentación para continuar con el ciclo de ajustes y mejoras al modelo.

Tanto la metodología de trabajo como el modelo de calidad propuestos fueron aplicados sin dificultades en ambos experimentos. Contar con la disposición del responsable del experimento para las validaciones y evacuar las dudas que fueron surgiendo, fue fundamental para llevar a cabo el trabajo de forma exitosa, logrando el cumplimiento de los objetivos establecidos. La aplicación de la metodología y modelo de calidad resultó más simple y requirió menos esfuerzo en la replicación, debido a que el diseño experimental, el contexto y los datos ya eran conocidos.

Varios de los problemas de calidad encontrados podrían ser fácilmente identificados y corregidos de manera manual, principalmente debido a la baja cantidad de sujetos y datos existentes. Sin embargo entendemos que si se detectara la presencia de algunos de estos problemas sobre una mayor cantidad de datos o en datos de mayor complejidad, probablemente tendrían un impacto importante en los resultados del experimento. Además, en esos casos, algunas de las correcciones planteadas de forma manual debieran ser automatizadas para poder llevarse a cabo.

Como la mayoría de los datos son registrados por los propios responsables del experimento, su ingreso se realiza de forma más controlada y no resulta sorprendente que se cometa una menor cantidad de errores sobre los datos. Los responsables del experimento son los primeros interesados en que los datos obtenidos reflejen lo más fielmente posible la realidad, y por lo cual suelen ser sumamente cuidadosos al momento de registrarlos.

Como resultado de la aplicación del modelo de calidad se encontraron problemas de calidad sobre los datos de ambos experimentos. En el caso base no se identificaron posibles limpiezas a aplicar sobre los mismos. Uno de los motivos por los cuales no se identificaron errores en los datos que requieran ser corregidos puede deberse a la baja cantidad de sujetos participantes, y a la baja canti-

dad y complejidad de los datos. De hecho, a partir de la aplicación del modelo de calidad sobre los datos de otros experimentos (presentados en el Capítulo 8), se observa que sí se identifican problemas de calidad que requieren corrección. Además, muchos de los errores identificados ocurren sobre los datos demográficos de los sujetos que no son utilizados para realizar los análisis estadísticos. A pesar de que es importante analizar y conocer la calidad también de estos datos, se debería evaluar si el esfuerzo (manual) que se debe invertir en su corrección permite obtener un beneficio acorde.

En la replicación, sin embargo, se realizaron limpiezas para corregir algunos de los errores identificados sobre los datos. Debido a que los errores ocurren sobre datos que son significativos para los análisis (tiempos y resultados de casos de prueba), su corrección podría impactar en los resultados del experimento. Es posible aplicar correcciones para 3 de los 9 casos en que se identifican errores en los datos. En todos los casos las correcciones se realizan de forma manual, ya que la cantidad de datos involucrada es baja.

Con respecto a los valores sospechosos no fue posible conocer el valor real en ninguno de los dos casos, por lo tanto no se aplicaron correcciones.

También es importante considerar el momento en el cual se lleva a cabo el análisis de la calidad de los datos. Mientras que en el caso base se realiza de forma posterior a la obtención de los resultados del experimento, en la replicación se realiza antes. Según la metodología propuesta, la aplicación del modelo de calidad sobre los datos de experimentos debiera llevarse a cabo antes de realizar el análisis estadístico, de forma tal de poder corregir los errores identificados y ejecutar los análisis con los datos "limpios". Este podría ser un motivo por el que se encuentren errores en la replicación que no se presentan en el experimento base. Al ejecutar los análisis, dada que es una baja cantidad de datos, los mismos experimentadores podrían identificar ciertos errores en los datos de forma manual y realizar las correcciones en el mismo momento. De hecho, esto sucede en el experimento base para las mediciones B10.3 y B14.1, como se detalla en la sección anterior. Sin embargo, este análisis ad-hoc no se aplica de manera sistemática y por lo tanto no resulta repetible en caso que se quiera volver a aplicar sobre los datos de otro experimento. Además, al no tener en cuenta un conjunto de métricas predefinidas basadas en los conceptos de Calidad de Datos, es probable que muchos aspectos de calidad se estén dejando por fuera del análisis. En resumen, esta forma de proceder ad hoc no permite asegurar que los errores sean necesariamente identificados y corregidos, ya que depende de quién esté analizando los datos (sus conocimientos y habilidades), así como la cantidad y complejidad de los mismos. La aplicación del modelo de calidad y la metodología propuesta permite que se siga una forma de trabajo sistemática, disciplinada y estructurada en las diferentes experiencias empíricas, que maximiza la probabilidad de encontrar problemas de calidad sobre los datos que serán luego corregidos.

A partir de los resultados obtenidos se observa que la métrica de calidad para la cual se obtiene el menor valor de calidad en el experimento base corresponde a datos ingresados por los propios estudiantes. Resulta llamativo considerando que los datos registrados por los sujetos son sólo los correspondientes a los cuestionarios (demográfico y satisfacción), una proporción bastante menor en relación a la cantidad de datos totales que se recolectan. Sin embargo, en la replicación la métrica con menor valor de calidad corresponde a la omisión de un cálculo por parte del experimentador (tiempo de segunda sesión). En este caso, las respuestas de los sujetos a los cuestionarios demográficos son "pre-procesadas" por el experimentador al copiarlas manualmente a las planillas (no se exportan desde el formulario web), por lo cual algunos errores de calidad podrían ser detectados y corregidos en ese momento. En la replicación existe además una mayor cantidad de datos que son registrados por los sujetos. Se puede ver que sobre alguno de estos nuevos datos (tiempos, información sobre compilaciones e instalaciones) se identifican también problemas de calidad.

También es de notar que el hecho de que los datos se registren y almacenen en planillas de cálculo y no en una base de datos relacional, puede dificultar el análisis no solo de su calidad sino también de los cálculos que se realicen a partir de los mismos. Esto sucede en muchos experimentos en los cuales la cantidad de datos registrados y la complejidad de la realidad no amerita el diseño y construcción de una base de datos relacional. Sin embargo, podría ser una causante de errores de calidad al dificultar la correcta interpretación de los datos registrados y las nomenclaturas utilizadas, el entendimiento de las referencias entre los propios datos, la falta de metadata, la falta de controles (ya sea durante el ingreso o análisis de los datos), etc. En particular, a raíz de esta aplicación particular se introdujeron nuevas métricas al modelo de calidad de datos (asociadas a las dimensiones de representación e interpretabilidad) que no habían sido consideradas antes.

Por otra parte, existen algunas oportunidades de mejora que fueron planteadas para el experimento base que son aplicadas sobre la replicación como iniciativa del propio experimentador (de manera previa e independiente al análisis de calidad). La aplicación de estas propuestas impacta en una mejora significativa en los resultados de las mediciones relacionadas a las mismas, sobre todo las referentes al cuestionario demográfico. De esta forma, se observa que si las actividades de prevención son consideradas y aplicadas, podrían existir mejoras en la calidad de los datos que se analizan y por lo tanto, mejorar la confianza y calidad de los resultados que se obtienen.

Resultó interesante poder aplicar el modelo y métricas de calidad sobre los datos de dos ejecuciones que corresponden al mismo experimento. De esta forma, resultó posible instanciar las mismas métricas sobre los datos de diferentes casos, realizando las adaptaciones y ajustes correspondientes (como sucedió en el caso de la replicación). Si pensamos en una generalización del modelo y métricas de calidad, donde se busca la aplicación de las métricas propuestas sobre los datos de otros experimentos, podemos considerar que de la misma manera se podrían adecuar estas métricas a otros datos y otra realidad. También nos hace pensar que la calidad de los datos es dependiente de la propia ejecución y no del diseño del experimento ni de los experimentadores, ya que para dos ejecuciones diferentes del mismo experimento los resultados de calidad obtenidos fueron diferentes.

La Tabla 14 muestra cuál fue el esfuerzo invertido (en horas) por cada participante (Analista en Calidad de Datos y Responsable del Experimento) en cada fase de la metodología, por cada experimento. El analista de calidad invierte un 89% y 85% del esfuerzo total en cada experimento respectivamente, participando durante todo el ciclo y con mayor dedicación debido a las actividades que realiza. El esfuerzo (bajo) que se invierte en aplicar la metodología de trabajo propuesta se debe principalmente a la baja cantidad y complejidad de los datos.

En el experimento base, las horas dedicadas a las primeras fases se vieron incrementadas por la falta de entendimiento y dificultad en la interpretación de ciertos datos registrados en las planillas de cálculo. También se trabajó durante estas etapas en la elaboración del modelo conceptual de forma de entender de mejor manera la realidad y los datos. Sin embargo, durante la replicación el esfuerzo dedicado a las primeras dos fases se vio reducido significativamente, debido a que ya se contaba con el conocimiento requerido.

La fase de evaluación de la calidad de datos fue la que requirió mayor esfuerzo en ambos casos, ya que es cuando se implementan las mediciones definidas. La reutilización de las fórmulas de cálculo usadas para el caso base, disminuye el esfuerzo dedicado en la replicación durante esta fase.

Debido a que no se identificaron errores sobre los datos que deban ser corregidos en el caso base, la última fase es la que requirió menor esfuerzo. En la replicación, sin embargo, se identifican errores sobre los datos que deben ser corregidos, por lo cual se incrementa el esfuerzo. Por otra parte, varias de las acciones de mejora que podrían aplicarse para futuras replicaciones del experimento co-

inciden con las planteadas para el experimento base, lo cual disminuye el esfuerzo dedicado en esta actividad.

Fase de la Metodología		**Esfuerzo (horas) – Exp. Base**		**Esfuerzo (horas) – Replicación**	
		Analista Calidad Datos	**Responsable Experimento**	**Analista Calidad Datos**	**Responsable Experimento**
1	Generar conocimiento del experimento	26	7	6	4
2	Instanciar el modelo de calidad de datos	27	2	16	2
3	Evaluar la calidad de los datos	42	2	32	2
4	Ejecutar acciones correctivas sobre los datos	16	3	11	4
TOTAL		**111**	**14**	**65**	**12**
		125		**77**	

Tabla 14: Esfuerzo dedicado por fase en los Experimentos MDD-UPV

En la Tabla 15 se muestra, a modo de resumen, la comparación entre los resultados obtenidos mediante la aplicación del modelo de calidad a ambos experimentos de MDD-UPV (base y replicación).

		Mediciones			Expe Base	Expe Replic.	Expe Base	Expe Replic.	Análisis de resultados
# Mét	**Métrica de Calidad**	**# Med**	**Exp. Base**	**Exp. MDD**	**Valor de Calidad**		**Valor de Calidad por métrica**		**Observaciones**
M1	Valor fuera de rango	1.1	Sí	Sí	0,913	1,000	0,968	0,983	En ambos casos corresponden a valores sospechosos sobre los tiempos para los cuales no es posible asegurar la existencia de un error, al no contar con datos de la realidad que permitan verificarlo.
		1.2	Sí	Sí	0,990	0,975			
		1.3	Sí	Sí	1,000	1,000			
		1.4	No	Sí		1,000			
		1.5	No	Sí		0,950			
		1.6	No	Sí		0,975			
M2	Falta de estandarización	2.1	Sí	Sí	1,000	1,000	0,941	1,000	El error detectado corresponde a datos del cuestionario de satisfacción. En este caso no se realiza corrección. Mientras que en el caso base los datos son ingresados directamente por los sujetos en un formulario web, en la replicación el responsable del experimento realiza un "pre-procesamiento" de los datos (al copiarlos desde las hojas de papel a las planillas). Esto hace que el registro de los datos se haga con el mismo formato, minimizando la cantidad de errores asociados a los mismos.
		2.2	Sí	Sí	1,000	1,000			
		2.3	Sí	Sí	1,000	1,000			
		2.4	Sí	Sí	1,000	1,000			
		2.5	Sí	Sí	0,706	1,000			
		2.6	No	Sí		1,000			
		2.7	No	Sí		1,000			
M3	Valor embebido	3.1	Sí	Sí	0,615	1,000	0,692	1,000	Ídem anterior.
		3.2	Sí	Sí	0,769	1,000			
M5	Registro con errores	5.1	Sí	Sí	N/E	N/E	1,000	1,000	
		5.2	Sí	Sí	N/E	N/E			
		5.3	Sí	Sí	1,000	1,000			
		5.4	Sí	Sí	1,000	1,000			
		5.5	Sí	Sí	1,000	1,000			
M7	Falta de precisión	7.1	Sí	Sí	1,000	1,000	1,000	1,000	
M8	Valor nulo	8.1	Sí	Sí	1,000	1,000	1,000	0,959	El error detectado corresponde a la omisión del ingreso de resultados de pasos de casos de prueba, que indican que las funcionalida-
		8.2	Sí	Sí	1,000	1,000			

		8.3	Sí	Sí	1,000	0,825			
		8.4	Sí	Sí	1,000	1,000			
		8.5	Sí	Sí	1,000	1,000			
		8.6	Sí	Sí	1,000	1,000			
		8.7	Sí	Sí	1,000	1,000			
		8.8	No	Sí		0,850			des no han sido implementadas. En este caso no se requiere realizar corrección.
M9	Información omitida	9.1	Sí	Sí	1,000	0,900	1,000	0,800	En la replicación se omite el cálculo de tiempos de sesiones, lo cual impacta directamente en el valor del tiempo total. Esto corresponde a un error en los datos que debe ser corregido.
		9.2	No	Sí		0,500			
		9.3	No	Mod		1,000			
M10	Registro faltante	10.1	Sí	Sí	1,000	1,000	0,996	0,933	En el experimento base se omite la respuesta de un sujeto a un cuestionario de satisfacción, debido a que abandonó el curso. En la replicación faltan resultados a los casos de prueba, debido a que las funcionalidades no fueron implementadas. Ninguno de los casos requieren ser corregidos. Se modifica la medición 10.3 ya que se incluye un nuevo cuestionario final.
		10.2	Sí	Sí	1,000	0,800			
		10.3	Sí	Mod	0,987	1,000			
M11	Regla de integridad de dominio	11.1	Sí	Sí	1,000	1,000	1,000	1,000	
		11.2	Sí	Sí	1,000	1,000			
		11.3	Sí	Sí	1,000	1,000			
		11.4	Sí	Sí	1,000	1,000			
		11.5	Sí	Sí	1,000	1,000			
		11.6	Sí	Sí	1,000	1,000			
		11.7	Sí	Sí	1,000	1,000			
M12	Regla de integridad intra-relación	12.1	Sí	Sí	0,971	0,963	0,971	0,963	En ambas instancias se encuentran valores de tiempos que no cumplen con la regla de integridad planteada, pero ninguno corresponde a errores en los datos que puedan ser corregidos.

M14	Regla de integridad inter-relación	14.1	Sí	Sí	0,978	0,971	0,857	0,926	Las inconsistencias entre respuestas identificadas para el caso base (medición 14.4) se deben principalmente a que el formulario web no permite el ingreso de valores nulos. Sin embargo, en la replicación el formulario es en papel y esto no sucede. En ambas instancias se identifican datos que no cumplen con alguna de las reglas de integridad planteadas. En la replicación se identifican errores en los datos que deben ser corregidos (14.1 y 14.7), no así para el experimento base. Debido a que en la replicación no existen valores embebidos, fue posible ejecutar la medición 14.5.
		14.2	Sí	Sí	1,000	1,000			
		14.3	Sí	Sí	1,000	1,000			
		14.4	Sí	Sí	0,538	1,000			
		14.5	Sí	Sí	N/E	0,895			
		14.6	Sí	Sí	0,769	0,842			
		14.7	No	Sí		0,700			
		14.8	No	Sí		1,000			
M16	Registro duplicado	16.1	Sí	Sí	1,000	1,000	1,000	1,000	
		16.2	Sí	Sí	1,000	1,000			
M17	Estructura de datos	17.1	Sí	Sí	Regular	Regular			En todos los casos corresponden a propuestas de mejora sobre el conjunto de datos y la forma de almacenamiento (planilla de cálculo)
M18	Formato de datos	18.1	Sí	Sí	Bueno	Regular			
M19	Facilidad de entendimiento	19.1	Sí	Sí	Regular	Regular			
M20	Metadata	20.1	Sí	Sí	Regular	Regular			

Tabla 15: Comparación de los resultados de la aplicación de métricas de calidad sobre los datos del experimento Base y Replicación

Capítulo 8: Aplicación de la Metodología y Modelo de Calidad sobre los Datos de Experimentos de Técnicas de Verificación

En este capítulo se presenta cómo se aplicó la metodología de trabajo y el modelo de calidad propuesto sobre los datos de dos experimentos que evalúan la Efectividad de Técnicas de Verificación. Sus diseños experimentales tienen algunas similitudes, a pesar de que fueron ejecutados por diferentes Centros de Investigación. Uno de ellos se ejecutó en el año 2004-2005 por el Grupo de Investigación en Ingeniería de Software Empírica (GrISE) [95] de la Universidad Politécnica de Madrid (UPM), y el otro fue ejecutado en el año 2008-2009 por el Grupo de Ingeniería de Software (GrIS) [96] de la Universidad de la República (UdelaR). A lo largo de este capítulo los llamaremos Expe-UPM y Expe-UdelaR respectivamente.

Por motivos de espacio, estos casos de aplicación se presentan de forma resumida. La información detallada de cada aplicación se encuentra en el Anexo A y Anexo B.

8.1 Fase 1: Generar conocimiento del experimento

Con el fin de generar conocimiento sobre los experimentos se llevaron a cabo reuniones de trabajo con sus respectivos responsables, se analizó el material disponible sobre los mismos [97]–[99], y se realizaron algunas consultas puntuales (de forma personal o vía mail).

8.1.1 Diseño experimental

Ambos experimentos se desarrollaron en un contexto académico, en el marco de una asignatura de grado.

El principal objetivo de Expe-UdelaR era conocer las relaciones de efectividad entre las técnicas de verificación utilizadas y los distintos tipos de defectos. Participaron 14 estudiantes utilizando 5 técnicas de verificación diferentes, sobre 4 programas (archivos de software) de distinta naturaleza. El diseño del experimento distribuye a los 14 participantes en 40 experiencias de verificación (experimentos unitarios), cada una con un programa y una técnica de verificación a ejecutar. Los verificadores registraban los defectos detectados y los clasificaban según dos taxonomías: ODC [100] y Beizer [101]. La taxonomía de ODC es una clasificación ortogonal de defectos, mientras la taxonomía de Beizer es jerárquica.

El objetivo de Expe-UPM era comparar la efectividad relativa de técnicas de pruebas dinámicas (estructural y funcional) y estáticas (revisión), y relacionarlas con los tipos de faltas que detectan. El estudio estaba compuesto de dos experimentos. El análisis de calidad se realizó sobre los datos del segundo experimento ejecutado. En el segundo experimento participaron 32 estudiantes divididos en 6 grupos de trabajo. Los integrantes de cada uno de los grupos ejecutaban una de las técnicas asignadas sobre el mismo programa, de manera tal que todos los sujetos ejecutaban cada una de las tres técnicas sobre cada uno de los tres programas. El experimento incluía la documentación de las instrucciones y guías a seguir para cada una las técnicas.

8.1.2 Datos recolectados y almacenados

En Expe-UdelaR se registraba por cada experiencia de verificación: fecha y hora de comienzo y finalización y tiempo de diseño y ejecución de casos de prueba. Además, por cada defecto detectado se registraba: archivo de software que contiene la clase con el defecto, número de línea de código donde se encuentra el defecto, clasificación en ODC y Beizer, estructura (IF, FOR, WHILE, etc.) y el número de línea en el código en que comienza la misma, tiempo de detección (tiempo para hallar un defecto en el código), y descripción del defecto. Para la recolección de datos se utilizó una herramienta disponible vía web llamada Tverificar (o Grillo), construida a medida para la recolección de datos del experimento [102]. Cada verificador accedía a las experiencias que tenía asignadas en la herramienta y registraba los datos requeridos. Los datos son almacenados en una base de datos relacional.

En Expe-UPM se ingresaban los siguientes datos: experiencia de los sujetos (relativa con lenguaje C, y absoluta); tiempo requerido para aplicar la técnica (en construir abstracciones para revisión, elaborar casos de prueba para estructural y funcional); tiempo requerido para la revisión, tiempo de ejecución de casos de prueba, y tiempo de detección de fallos; cantidad de niveles de abstracción (revisión), de clases de equivalencia (funcional), y de casos de prueba; estimación de porcentaje de fallos encontrados, y de confianza sobre la correcta aplicación de la técnica; descripción de la falla. Notar que los datos ingresados dependen de la técnica que está siendo aplicada. Los datos eran ingresados por los sujetos en formularios (en papel) provistos por los experimentadores. Debían completar un formulario por cada técnica aplicada. Estos datos eran transcriptos por el responsable del experimento en una planilla de cálculo y se ingresaban en un fichero SPSS.

8.2 Fase 2: Instanciar el modelo de calidad de datos

Durante esta fase se llevaron a cabo las estrategias propuestas por la metodología de aplicación del modelo de calidad de datos. En Expe-UdelaR se aplicaron las 3 estrategias definidas en dicha metodología. En Expe-UPM no se utilizaron herramientas para el registro de datos, por lo cual no se aplica la estrategia 3.

Las Tablas 16 y 17 muestran cuáles son las métricas de calidad que se aplican sobre los datos de cada experimento, y sobre qué objetos particulares del dominio se instancian.

De las 21 métricas de calidad que forman parte del Modelo de Calidad de Datos, en Expe-UdelaR se aplicaron 15 métricas sobre 51 objetos particulares del dominio, mientras que en Expe-UPM se aplicaron 16 métricas sobre 71 objetos (hay 11 métricas en común entre ambas aplicaciones). Las restantes métricas no resultaron aplicables debido a las características de los datos y del contexto bajo estudio. Por ejemplo, en Expe-UdelaR no se identificaron reglas de integridad a considerar entre datos de diferentes relaciones, ni interesó medir la precisión de los datos. En Expe-UPM no se identificaron valores que deban ser únicos, ni casos de contradicciones entre los datos.

La forma de registrar los resultados de las mediciones de calidad es diferente en cada caso. En Expe-UPM se utilizan planillas de cálculo, y su forma de registro coincide con el caso de los experimentos MDD-UPV. En Expe-UdelaR se creó una nueva tabla por cada métrica de calidad aplicada en la misma base del experimento. Cada una de estas tablas contiene el resultado obtenido a partir de la aplicación de la métrica sobre los objetos definidos. La estructura de cada tabla de registro depende de la métrica a registrar, y contiene la información relevante así como las referencias (*foreign keys*) a las tablas de la base involucradas. Se crea además un catálogo de métricas de calidad (*reg_ catalogo_ errores*) que contiene información específica sobre cada métrica aplicada.

8.3 Fase 3: Evaluar la calidad de los datos

Las Tablas 16 y 17 muestran el resultado obtenido luego de la aplicación de cada métrica de calidad sobre los datos de ambos experimentos. En Expe-UdelaR se ejecutan 51 mediciones sobre cada uno de los objetos del dominio definidos en la fase anterior, mientras que en Expe-UPM se ejecutan 71 mediciones. Del total de mediciones ejecutadas:

- El 74% (38 mediciones) se realizan de forma automática en Expe-UdelaR mediante la ejecución de consultas SQL y algoritmos programados; mientras que en Expe-UPM se automatiza el 82% (58 mediciones) mediante el uso de fórmulas de cálculo.
- El 18% (9 mediciones) se realizan de forma manual en Expe-UdelaR y el 14% (10 mediciones) en Expe-UPM, por no ser posible o necesaria su automatización. En todos los casos corresponden a revisiones o verificaciones manuales sobre datos particulares o su conjunto.
- Las mediciones del restante 8% (4 mediciones) y 4% (3 mediciones) respectivamente son aquellas para las cuales no es posible su ejecución (indicadas con "N/E" en las Tablas 16 y 17). Esto sucede para las mediciones manuales con alto costo de implementación. Por ejemplo, en Expe-UdelaR sucede para el caso de Valor fuera rango (A1.1 y A1.2), Registro inexistente (A3.1) y Registro con errores (A4.1).

Como resultado se encuentra que la medida obtenida es menor a 1,00 o 'Regular' para 20 de las 51 mediciones ejecutadas en Expe-UdelaR y para 37 de las 71 en Expe-UPM, indicando la presencia de un problema de calidad en los datos. Estas mediciones corresponden a 11 métricas de calidad diferentes en el primer caso, y a 9 en el segundo.

A continuación, se analizan los resultados obtenidos y se clasifican los problema de calidad identificados.

8.3.1 Errores en los datos

En Expe-UdelaR se identifican datos erróneos a partir de la ejecución de 19 de las 20 mediciones, y en Expe-UPM para 24 de las 37 mediciones. En las Tablas 16 y 17 se muestra la cantidad de objetos con error que se encuentran en cada caso.

8.3.2 Valores sospechosos

En ambos experimentos los valores sospechosos corresponden a la métrica Valor Fuera de Rango. Debido a que la definición de los rangos es arbitraria, no se puede asegurar la existencia de errores en los datos hasta compararlos (en caso de ser posible) con valores reales.

En Expe-UdelaR se detecta la presencia de 1 valor sospechoso, a partir de la medición ejecutada sobre el tiempo de detección de defectos. En Expe-UPM el resultado de 9 mediciones sobre los tiempos y cantidades muestran la presencia de valores sospechosos. Todos estos casos son analizados con el responsable del experimento y, siempre que sea posible, comparados con otras fuentes de datos para identificar si corresponden o no a errores en los datos.

# Mét	Métrica de Calidad	# Med	Objetos (tablas y atributos)	Valor de Calidad	Agregación	Método de medición	Problema de Calidad	Corrección	Cant. datos con problema	Cant. datos corregidos
M1	Valor fuera de rango	A1.1	Experimento.time_casos	N/A	0,971	Automático (consultas SQL)				
		A1.2	Experimento.time_ejecucion	N/A						
		A1.3	Registro_Defecto.time_deteccion	0,971			Valor Sospechoso	Etapa 2 – Manual	21	4
M2	Falta de estandarización	A2.1	Experimento.time_casos	1,000	1,000	Automático (consultas SQL)				
		A2.2	Experimento.time_ejecucion	1,000						
		A2.3	Registro_Defecto.time_deteccion	1,000						
		A2.4	Registro_Defecto.linea	1,000						
		A2.5	Registro_Defecto.linea_estructura	1,000						
M4	Registro inexistente	A4.1	Registro_Defecto	N/A	0,979	Manual (revisión de fuentes)				
		A4.2	Archivo	0,979			Error en datos	Etapa 2 – Manual	1	1
M5	Registro con errores	A5.1	Registro_Defecto.linea Registro_Defecto.linea_estructura Registro_Defecto.time_deteccion Registro_Defecto.descripcion Registro_Defecto.estructura_id Registro_Defecto.archivo_id Registro_Defecto.tipo_id Registro_Defecto.valor_categoria_id	N/E	N/E					
M6	Valor fuera de referencial	A6.1	Tecnica.nombre	1,000	0,874	Manual (revisión de valores) y Automático (consultas SQL)				
		A6.2	Software.nombre	1,000						
		A6.3	Tipo_Defecto.nombre	1,000						
		A6.4	Estructura.nombre	1,000						
		A6.5	Taxonomia.nombre	1,000						
		A6.6	Categoria.nombre	0,500			Error en datos	Etapa 1 – Auto-	6	6

		A6.7	Registro_Taxonomia	0,616			Error en datos	mática	3721	3721
M8	Valor nulo	A8.1	Usuario.perfil_id	1,000	0,973	Automático (consultas SQL)				
		A8.2	Registro_Defecto.tipo_id	1,000						
		A8.3	Registro_Defecto.linea	1,000						
		A8.4	Experimento.nombre	1,000						
		A8.5	Registro_Defecto.time_deteccion	1,000						
		A8.6	Registro_Taxonomia.registro_id	1,000						
		A8.7	Experimento.time_ejecucion	0,932			Error en datos	Etapa 2 – Manual	3	3
		A8.8	Registro_Taxonomia.taxonomia_id	0,863			Error en datos	Etapa 1 – No requerida	1324	1324
		A8.9	Registro_Taxonomia.valor_categoria_id	0,999			Error en datos	Etapa 2 – No requerida	2	2
		A8.10	Experimento.time_casos (técnica dinámica)	0,932			Error en datos	Etapa 2 – Manual	3	3
M9	Información omitida	A9.1	Registro_Defecto (para ODC)	0,997	0,999	Automático (consultas SQL)	Error en datos	Etapa 2 – No ejecutada	3	0
		A9.2	Registro_Defecto (para Beizer)	1,000						
M11	Regla de integridad de dominio	A11.1	Registro_Defecto.linea	0,992	0,981	Automático (consultas SQL)	Error en datos	Etapa 2 – Manual	8	4
		A11.2	Registro_Defecto.linea_estructura	0,997			Error en datos		3	2
		A11.3	Experimento.time_ejecucion	0,955			Error en datos		2	1
M12	Regla de integridad intra-relación	A12.1	Experimento.time_casos (técnica estática)	1,000	0,978	Automático (consultas SQL)				
		A12.2	Experimento.time_casos (técnica dinámica)	1,000						
		A12.3	Registro_Defecto.time_deteccion	0,911			Error en datos	Etapa 1 – Semi-automática	90	Depende decisión usuario
		A12.4	Registro_Defecto.linea_estructura	1,000						

M13	Valor único	A13.1	Categoria.nombre	1,000	1,000	Automático (consultas SQL)				
		A13.2	Valor_Categoria.nombre, categoria_padre	1,000						
		A13.3	Experimento.nombre	1,000						
M15	Referencia inválida	A15.1	Registro_Taxonomia.taxonomia_id	1,000	0,950	Automático (consultas SQL)				
		A15.2	Registro_Taxonomia.registro_id	0,951			Error en datos	Etapa 1 – Automático	472	472
		A15.3	Valor_Categoria.categoria_padre	0,900			Error en datos	Etapa 1 – Automático	19	19
M16	Registro duplicado	A16.1	Registro_Defecto	1,000	0,947	Automático (consultas SQL)				
		A16.2	Registro_Taxonomia (para ODC)	0,842			Error en datos	Etapa 1 – Automática	1534	1534
		A16.3	Registro_Taxonomia (para Beizer)	1,000						
M17	Registro contradictorio	A17.1	Archivo_Software	0,958	0,980	Automático (consultas SQL y programación)	Error en datos	Etapa 1 – Automática y Etapa 2 – Manual	2	2
		A17.2	Registro_Taxonomia (para ODC)	0,987			Error en datos	Etapa 1 – Semi-automática	122	122
		A17.3	Registro_Taxonomia (para Beizer)	0,996			Error en datos		43	43
M18	Estructura de datos	A18.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión de esquema y base de datos)	Error en datos	Etapa 0 - Manual		
M21	Metadata	A21.1	Conjunto de todos los datos	Aceptable	Aceptable					

Tabla 16: Resultado de la aplicación de métricas de calidad sobre los datos del experimento de Técnicas de Verificación de UdelaR

# Mét	Métrica de Calidad	# Med	Objetos (tablas y atributos)	Valor de Calidad	Agregación	Método de medición	Problema de Calidad
M1	Valor fuera de rango	D1.1	SubjectData.TechniqueApplicationTime	0,980	0,912	Automático (fórmulas de cálculo)	Valor Sospechoso
		D1.2	SubjectData.TechniqueApplicationTime	0,992			
		D1.3	SubjectData.TestCaseExecutionTime	0,934			
		D1.4	SubjectData.TestCaseExecutionTime	0,984			
		D1.5	SubjectData.Failure/FaultDetectionTime	0,961			
		D1.6	SubjectData.NoAbstraction	0,813			
		D1.7	SubjectData.NoClasses	0,781			
		D1.8	SubjectData.NoTestCases	0,688			
		D1.9	SUMA(TechniqueApplicationTime+TestCaseExecutionTime+Failure/FaultDetectionTime)	0,988			
		D1.10	SUMA(TechniqueApplicationTime+TestCaseExecutionTime)	1,000			
M2	Falta de estandarización	D2.1	SubjectData.TechniqueApplicationTime	1,000	0,968	Automático (fórmulas de cálculo)	
		D2.2	SubjectData.TestCaseExecutionTime	1,000			
		D2.3	SubjectData.Failure/FaultDetectionTime	1,000			
		D2.4	SubjectData.NoAbstraction	0,938			Error en los datos
		D2.5	SubjectData.NoClasses	0,990			
		D2.6	SubjectData.NoTestCases	1,000			
		D2.7	ObservableFaults (F1 a F7)	1,000			
		D2.8	FailureVisibility (F1 a F7)	1,000			
		D2.9	SubjectData.RelativeExperience	0,802			Error en los datos
		D2.10	SubjectData.AbsoluteExperience	0,938			
		D2.11	SubjectData.EstimatedDefects	0,990			
		D2.12	SubjectData.Confidence	0,958			
M5	Registro con errores	D5.1	ObservableFaults (F1 a F7)	N/A	N/A	Manual (comparación	

		D5.2	FailureVisibility (F1 a F7)	N/A		de valores registrados y reales)	
		D5.3	SubjectData.TechniqueApplicationTime SubjectData.TestCaseExecutionTime SubjectData.Failure/FaultDetectionTime	N/A			
M6	Valor fuera de referencial	D6.1	SubjectData.Program ObservableFaults.Program FailureVisibility.Program	1,000	1,000	Automático (fórmulas de cálculo)	
		D6.2	SubjectData.Technique ObservableFaults.Technique FailureVisibility.Technique	1,000			
		D6.3	SubjectData.Version ObservableFaults.Version FailureVisibility.Version	1,000			
M7	Falta de precisión	D7.1	ObservableFaults.Porcentage	1,000	1,000	Manual (verificación de formato de celdas)	
		D7.2	FailureVisibility.Porcentage	1,000			
M8	Valor nulo	D8.1	SubjectData.RelativeExperience	0,990	0,913	Automático (fórmulas de cálculo)	Error en los datos
		D8.2	SubjectData.AbsoluteExperience	0,802			
		D8.3	SubjectData.TechniqueApplicationTime	0,938			
		D8.4	SubjectData.TestCaseExecutionTime	0,917			
		D8.5	SubjectData.Failure/FaultDetectionTime	0,891			
		D8.6	SubjectData.NoAbstraction	0,844			
		D8.7	SubjectData.NoClasses	0,969			
		D8.8	SubjectData.NoTestCases	0,922			
		D8.9	SubjectData.EstimatedDefects	0,854			
		D8.10	SubjectData.Confidence	0,885			
		D8.11	ObservableFaults.Porcentage	0,948			
		D8.12	FailureVisibility.Porcentage	1,000			
M9	Información omitida	D9.1	ObservableFaults (F1 a F7)	0,948	0,896	Automático (fórmulas	Error en los datos

		D9.2	FailureVisibility (F1 a F7)	0,844		de cálculo)	
M10	Registro faltante	D10.1	SubjectData	1,000	1,000	Automático (fórmulas de cálculo)	
		D10.2	ObservableFaults	1,000			
		D10.3	FailureVisibility	1,000			
M11	Regla de integridad de dominio	D11.1	SubjectData.RelativeExperience	1,000	1,000	Automático (fórmulas de cálculo)	
		D11.2	SubjectData.AbsoluteExperience	1,000			
		D11.3	SubjectData.EstimatedDefects	1,000			
		D11.4	ObservableFaults.Porcentage	1,000			
		D11.5	FailureVisibility.Porcentage	1,000			
		D11.6	SubjectData.Confidence	1,000			
M12	Regla de integridad intra-relación	D12.1	ObservableFaults.Porcentage	1,000	0,955	Manual (verificación de fórmula)	
		D12.2	FailureVisibility.Porcentage	1,000			
		D12.3	ObservableFaults.Total	1,000			
		D12.4	FailureVisibility.Total	1,000			
		D12.5	FailureVisibility.Technique	1,000		Automático (fórmulas de cálculo)	
		D12.6	ObservableFaults.F1 FailureVisibility.F1	0,938			Error en los datos
		D12.7	SubjectData.Technique SubjectData.NoAbstraction	0,984			
		D12.8	SubjectData.Technique SubjectData.NoClasses	0,984			
		D12.9	SubjectData.Technique SubjectData.Failure/FaultDetectionTime SubjectData.NoTestCases	1,000			
		D12.10	SubjectData.RelativeExperience SubjectData.Technique	0,906			Error en los datos
		D12.11	SubjectData.AbsoluteExperience SubjectData.Technique	0,688			

M14	Regla de integridad inter-relación	D14.1	SubjectData, ObservableFaults, FailureVisi-bility	1,000	1,000	Automático (fórmulas de cálculo)	
M16	Registro duplicado	D16.1	SubjectData	1,000	1,000	Automático (fórmulas de cálculo)	
		D16.2	SubjectData	1,000			
M18	Estructura de datos	B18.1	Conjunto de todos los datos	Regular	Regular	Manual (revisión plani-llas de datos)	Oportunidad de mejo-ra
M19	Formato de datos	B19.1	Conjunto de todos los datos	Regular	Regular		
M20	Facilidad de entendimien-to	B20.1	Conjunto de todos los datos	Regular	Regular		
M21	Metadata	B21.1	Conjunto de todos los datos	Regular	Regular		

Tabla 17: Resultado de la aplicación de métricas de calidad sobre los datos del experimento de Técnicas de Verificación de UPM

N/A	No Aplica
N/E	No Ejecutada
	Presenta un problema de calidad (resultado menor a 1,00)
	No presenta un problema de calidad (resultado igual a 1,00)

Tabla 18: Referencias Tablas 16 y 17

8.3.3 Oportunidades de mejora

En Expe-UdelaR no se identifican oportunidades de mejora asociados a los problemas de calidad encontrados. En Expe-UPM se identifican propuestas de mejora a considerar sobre las 4 mediciones que refieren a las métricas aplicadas al conjunto de datos (Estructura de datos, Formato de datos, Facilidad de entendimiento, Metadata).

8.4 Fase 4: Ejecutar acciones correctivas sobre los datos

Se analiza en conjunto con cada responsable del experimento si para los problemas de calidad identificados es posible aplicar acciones correctivas.

En el caso de Expe-UdelaR, se ejecutan limpiezas para la mayoría de los errores identificados sobre los datos. En la Tabla 16 se muestra cuáles son los problemas de calidad que se corrigen de forma automática, semi-automática y manual.

El primer paso (etapa 0) para la limpieza de los datos consiste en la corrección de errores en el diseño de la base. Para esto se analiza el problema de calidad Estructura de datos (A18.1) y se define un nuevo esquema que los corrige. Luego se ejecutan las limpiezas de todos los errores que pueden ser corregidos de forma automática o semi-automática (etapa 1) acompañando la migración de los datos hacia el nuevo esquema [103]. Se construye una aplicación específicamente con el fin de ejecutar las actividades de limpieza y migración de los datos. La corrección de errores de forma manual (etapa 2) se ejecuta sobre aquellos casos particulares que es necesario corregir pero no es posible automatizar. Estas limpiezas se realizan luego de finalizada la ejecución del programa de limpieza y migración, mediante un *script* directamente sobre la base de datos destino.

En particular, para los valores sospechosos se siguen dos estrategias de corrección: se consulta a los sujetos del experimento si creen o recuerdan que existe un error en esos datos, y se comparan los valores fuera de rango con los registrados en planillas de cálculo (en paralelo a la herramienta Grillo). De esta forma, se logran corregir 4 de los 21 casos identificados como sospechosos.

De los 20 objetos que contienen algún problema de calidad, se logra corregir el total de datos con errores en 15 casos y parcialmente en 4. Se aplican diferentes forma de corrección:

- 8 objetos se corrigen de forma automática, acompañando la migración de los datos (etapa 1);
- 3 objetos se corrigen de forma semi-automática, mediante consultas al usuario de la herramienta (etapa 1);
- 5 objetos se corrigen de forma manual, 4 luego de ejecutar el programa de limpieza y migración directamente sobre la base de datos destino (etapa 2) y 1 de forma previa (etapa 0);
- 1 objeto se corrige parte automática y parte manualmente (etapas 1 y 2);
- en 2 casos no es necesario realizar limpiezas, ya que los problemas de calidad se corrigen debido a la ejecución de otras limpiezas;
- en 1 caso no es posible aplicar limpieza, debido a que no se conoce el valor real.

En el caso de Expe-UPM no se identifican posibles limpiezas a aplicar sobre los errores en los datos ni valores sospechosos encontrados. En todos los casos sería necesario comparar manualmente los datos registrados por los sujetos en los formularios con los datos ingresados por los experimentadores en las planillas de cálculo, ya que puede haber ocurrido un error al transcribir los valores en las planillas. Por razones de costo (esfuerzo) asociado, no se llevó a cabo esta acción.

Muchos de los errores identificados suceden sobre datos registrados en la hoja "*Subject Data*". Esta hoja contiene los valores de tiempos y cantidades registrados por los sujetos. El experimentador indicó que esos datos no son utilizados para realizar análisis estadísticos debido a que no se conside-

ran “datos fiables”. Los resultados obtenidos a partir de las mediciones de calidad comprueban este hecho.

En ambos casos se proponen actividades de prevención que consideran a todos los problemas de calidad identificados, y que podrían ser incorporadas para futuras replicaciones de los experimentos de forma de contribuir en la mejora de su calidad. Estas se encuentran detalladas en [103].

8.5 Análisis y discusión

Estos experimentos constituyeron el primer y último caso de aplicación de la metodología y modelo de calidad realizados en el marco del trabajo de tesis. Sus diseños experimentales tienen similitudes, a pesar de que fueron ejecutados por grupos de investigación diferentes. De todas formas, algunas métricas fueron aplicadas en un experimento y en el otro no, obteniendo también diferentes resultados en cada caso.

Durante la aplicación del modelo de calidad a los datos del Expe-UdelaR se construyó la primera versión del modelo de calidad de datos, y se definió e instanció por primera vez la metodología de trabajo propuesta. Al ser la primera experiencia y debido a la gran cantidad de datos involucrados en el análisis, es el caso que requirió más esfuerzo por parte de todos los roles participantes (ver Tabla 19).

En Expe-UdelaR se contó con el tiempo y disposición del responsable del experimento para las validaciones y despejar las dudas que fueron surgiendo, por lo cual el trabajo pudo llevarse a cabo sin dificultades. Una dificultad enfrentada en el caso del Expe-UPM fue que se trabajó a distancia con el responsable del experimento, ya que no se encontraba físicamente en el mismo lugar de trabajo que el analista de calidad. La dificultad radicó fundamentalmente en la coordinación y ejecución de las reuniones de trabajo que son propuestas por la metodología, principalmente en lo que refiere a las validaciones. Por ejemplo, no fue posible realizar la última reunión de validación en la que se identifican las posibles acciones de corrección sobre los datos. Esto se ve reflejado en la baja dedicación por parte del responsable del experimento (Tabla 19) y en la fase 4 en general.

Una de las diferencias identificadas entre ambos casos de aplicación es la forma de recolección y almacenamiento de los datos. Mientras que en el Expe-UPM los datos se recolectan manualmente y se registran en planillas de cálculo, en Expe-UdelaR los datos se recolectan mediante el uso de una herramienta y se registran en una base de datos relacional. En el caso de Expe-UPM existen dos fuentes de datos para el experimento: una planilla de cálculo y un fichero SPSS. Debido a que se desconoce la relación que existe entre los datos registrados en ambos repositorios (a pesar de que se consultó con el experimentador), solamente se utilizó la planilla de cálculo como repositorio de datos base para el trabajo de calidad.

En ambos casos se identificaron problemas de calidad sobre los datos de los experimentos. Estos problemas fueron analizados por parte del responsable de forma de evaluar su impacto en los resultados del experimento. En caso de identificar acciones correctivas, se aplicaron las correcciones necesarias.

En el caso de Expe-UdelaR se observa que varios de los problemas identificados sobre los datos se deben a errores en el diseño de la base de datos y/o en la herramienta utilizada para el registro de datos, que se traducen en errores en los datos. Estos problemas se identifican mediante la aplicación de las métricas *Valor Nulo, Referencia Inválida, Registro Duplicado* y *Registro Contradictorio*. Todos los datos que presentan alguno de estos problemas de calidad lograron ser corregidos. Además, la definición del nuevo esquema de base de datos evitará que vuelvan a presentarse en futuras replicaciones. También se encuentran problemas de calidad sobre los datos ingresados por los administradores de la herramienta (responsables del experimento). Esto sucede para los casos de *Registro ine-*

xistente y *Valor fuera de referencial*. El resto de los datos que contienen algún problema de calidad son ingresados por los sujetos. Algunos de ellos podrían haberse prevenido mediante la definición de restricciones sobre los datos, ya sea a nivel de la base de datos o de la misma herramienta. Esto sucede para los casos de *Valor fuera de rango*, *Reglas de integridad de dominio, Reglas de integridad intra-relación, Valor nulo* e *Información omitida*.

Una diferencia significativa entre ambas aplicaciones se puede apreciar en la etapa de corrección de datos. Mientras que en el caso del Expe-UdelaR se logran corregir la mayoría de los errores identificados, en el caso de Expe-UPM no se aplicaron acciones correctivas sobre los datos. Esto puede deberse a varios motivos. Uno de ellos es debido a la naturaleza y origen de los problemas de calidad identificados en cada caso. En el Expe-UPM muchos errores ocurren sobre datos ingresados por los sujetos (valores nulos, valores fuera de rango, incumplimiento de reglas de consistencia). Una posible forma de corrección podría ser entonces comparar manualmente los datos ingresados en la planilla de cálculo con los datos registrados por los sujetos en los formularios de papel. Debido al alto esfuerzo asociado, no se llevó a cabo esta acción. Además, se consultó con el responsable del experimento sobre los casos en los que los problemas de calidad suceden sobre datos ingresados por éste (omisión de información, incumplimiento de reglas de consistencia). El experimentador no recuerda la ocurrencia de algún suceso fuera de lo común que ocasione la presencia de esos problemas de calidad. Otro de los motivos por los cuales pueden no haberse identificado acciones correctivas en Expe-UPM es que no fue posible realizar la última reunión de validación con el responsable del experimento. Finalmente, la cantidad de objetos que presentan algún problema de calidad es significativamente mayor en el Expe-UdelaR, por lo cual se considera necesario automatizar las acciones de limpieza sobre los datos.

Los principales aspectos a destacar del trabajo de calidad ejecutado sobre los datos del Expe-UdelaR son:

- Se aplicó por primera vez la metodología, modelo y métricas de calidad definidas a los datos de un experimento en ingeniería de software obteniendo resultados de valor para los responsables del experimento.
- Se ejecutaron las mediciones definidas, y como resultado se identificaron problemas de calidad sobre los datos que requerían ser corregidos.
- Se logró limpiar más del 98% de los problemas de calidad identificados.
- A pesar de que aún no se han realizado los análisis estadísticos con los datos limpios, dada la naturaleza de los problemas de calidad identificados y la cantidad de objetos afectados por los mismos, creemos que es altamente probable que los resultados del experimentos se vean impactados.
- Se ejecutó la migración de los datos a un nuevo esquema de base que se adecua mejor a la realidad planteada. Este es el único caso en el cual resultó necesario definir un nuevo esquema de base de datos con el fin de corregir los errores identificados a nivel del diseño de la base de datos, y llevar a cabo la migración de datos correspondiente.
- Se construyó un programa que automatiza la gran mayoría de los procesos de limpieza y la totalidad de la migración de los datos. En este caso resultó imprescindible que la medición y corrección se realizara de forma automática debido a la gran cantidad de datos.

Por otra parte, se destacan los siguientes beneficios obtenidos a partir de la aplicación sobre el Expe-UPM.

- Se cuenta con un nuevo caso de aplicación del modelo y métricas de calidad, mostrando una vez más que es posible aplicar la metodología propuesta sobre los datos de un experimento en ingeniería de software.

- Se aplicó el modelo y métricas de calidad sobre un experimento que está inmerso en un contexto experimental diferente (nuevo grupo de investigación y responsable), y en una modalidad también diferente (a distancia).
- El diseño del experimento tiene similitudes con el Expe-UdelaR, lo que permite comparar los resultados en ambas aplicaciones.
- Un punto a destacar es que mediante este trabajo se confirmó una suposición que tenía el experimentador sobre el nivel de calidad de ciertos datos que consideraba "no fiables", y por lo tanto no se utilizaron para los análisis del experimento. Entendemos que el análisis de calidad realizado agregó objetividad, sistematización y disciplina, a la subjetividad planteada por el experimentador. Los resultados de las mediciones ejecutadas muestran que sobre esos datos es sobre los que se detecta la presencia de una mayor cantidad de problemas de calidad.

La Tabla 19 muestra cuál fue el esfuerzo asociado (en horas) por cada participante en cada fase de la metodología (por cada experimento). En ambos casos el analista de calidad invierte un 93% del esfuerzo. Se observa que el esfuerzo total invertido en el caso del Expe-UdelaR es 5 veces mayor en comparación al Expe-UPM. Es el caso en el que que se dedica más esfuerzo considerando las 4 aplicaciones (ver Tabla 14 del Capítulo 7). Esto se debe a varios motivos. Por un lado, al ser el primer caso de aplicación, los analistas de calidad no tenían experiencia en el tema por lo que parte del esfuerzo puede deberse al propio aprendizaje. Además, parte del esfuerzo dedicado a desarrollar el modelo de calidad y la metodología de trabajo fue incluido en el tiempo dedicado por los analistas. Por otra parte, la cantidad de datos que se analizan es considerablemente mayor, lo cual incrementa el esfuerzo. En este caso sería sumamente difícil poder almacenar los datos recolectados durante el experimento en una planilla de cálculo debido a su cantidad y complejidad, por lo cual se utiliza una base de datos relacional. Otro factor que influye es que justamente debido a la cantidad y complejidad de los datos participan 2 analistas de calidad, incrementando el esfuerzo de este rol sobre todo por las actividades de coordinación.

Se observa que la fase en la cual se invierte más esfuerzo en el Expe-UdelaR es la de corrección (fase 4). Esto se debe principalmente a que fue necesario implementar una herramienta que automatice la limpieza y migración de los datos a un nuevo esquema. Además, en este caso se aplican acciones correctivas sobre la mayoría de los problemas de calidad identificados.

Por otra parte, la baja cantidad y complejidad de los datos recolectados en el Expe-UPM es un factor influyente en el bajo esfuerzo dedicado, tanto por el analista de calidad como por el experimentador. En este caso, el esfuerzo dedicado es el menor de entre todas las aplicaciones, por parte de ambos roles participantes.

Fase de la Metodología		Esfuerzo (horas) – Expe-UdelaR		Esfuerzo (horas) – Expe-UPM	
		Analista Calidad de Datos	**Responsable Experimento**	**Analista Calidad de Datos**	**Responsable Experimento**
1	Generar conocimiento del experimento	47	4.5	14	3
2	Instanciar el modelo de calidad de datos	49.5	9	21	1
3	Evaluar la calidad de los datos	108.5	2.5	30	1
4	Ejecutar acciones correctivas sobre los datos	132	7	2	0
TOTAL		337	23	67	5
		360		**72**	

Tabla 19: Esfuerzo dedicado por fase en el Experimento

Capítulo 9: Resultados y Discusión

Durante este trabajo fue propuesto y aplicado un modelo de calidad y una metodología de trabajo (que utiliza dicho modelo) sobre los datos recolectados por cuatro experimentos en Ingeniería de Software que involucran sujetos humanos.

La Tabla 20 muestra las principales características así como los resultados obtenidos en cada caso de aplicación. Cuando decimos "objeto" nos referimos al conjunto de datos sobre el que se aplicó la métrica, por ejemplo una tabla o columna.

Experimento	UdelaR	UPV-Base	UPV-Replicación	UPM
Grupo de Investigación	GrIS	PROS	PROS	GrISE
Objetivo	Conocer y comparar la efectividad de técnicas de verificación y tipos de defectos que detectan	Comparar el paradigma MDD con métodos de desarrollo de software tradicional	Comparar el paradigma MDD con métodos de desarrollo de software tradicional	Conocer y comparar la efectividad de técnicas de verificación y tipos de defectos que detectan
Repositorio de datos	Base de datos relacional	Planillas de cálculo	Planillas de cálculo	Planillas de cálculo
Recolección de datos	Herramienta web	Manual (papel) y formulario web	Manual (papel)	Manual (papel)
Cantidad de sujetos	14	26	19	32
Cantidad de métricas aplicadas	15	16	16	16
Cantidad de métricas instanciadas sobre objetos del dominio	51	47	59	71
Cantidad de mediciones ejecutadas	Automáticas: 38 Semi-automáticas: Manuales: 9	Automáticas: 32 Manuales: 12	Automáticas: 44 Manuales: 13	Automáticas: 58 Manuales: 10
Cantidad de problemas de calidad identificados	11	9	10	9
Cantidad de objetos y datos con problemas de calidad	Objetos: 20 Datos: 7379	Objetos: 13 Datos: 55	Objetos: 17 Datos: 43	Objetos: 37 Datos: 219
Cantidad de objetos y datos corregidos	Objetos: 19 Datos: 7353	0	Objetos: 3 Datos: 12	0

Tabla 20: Comparación de características y resultados de los experimentos

En las siguientes secciones se abordan y comparan los diferentes aspectos, características y resultados presentados en la Tabla 20, incluyendo la instanciación de las métricas de calidad y qué problemas se presentan en estos casos. Luego se analiza y discute acerca de los resultados obtenidos, y finalmente se muestra la opinión y experiencia de los experimentadores respecto al uso y aplicación del modelo y la metodología de calidad.

9.1 Instanciación de las métricas de calidad sobre los datos de los experimentos

Las 21 métricas de calidad que conforman el modelo de calidad propuesto fueron aplicadas en alguno de los casos de aplicación. En la Tabla 21 se muestra y compara la cantidad de métricas de calidad que son aplicadas en común entre los 4 experimentos. El detalle de qué métrica fue aplicada en cada experimento se encuentra en la Tabla 23.

Experimentos	UPV-Base	UPV-Replicación	UPM
UdelaR	10	10	11
UPV-Base	N/A	16	15
UPV-Replicación	N/A	N/A	15

Tabla 21: Cantidad de métricas aplicadas en común entre experimentos

Observamos que en ambas experiencias de MDD-UPV se aplicaron las mismas 16 métricas de calidad. Entendemos que se debe a que el diseño experimental es el mismo en ambos casos, y los datos que se registran coinciden en su mayoría. Por otra parte, observamos que existe una mayor cantidad de métricas aplicadas en común entre el experimento de UPM y los de UPV, que con respecto al de UdelaR (15 en el primer caso, 11 en el segundo). A pesar de que el diseño experimental de UdelaR y UPM no son exactamente iguales, existen varias similitudes entre estos. Esto nos hace pensar que además del diseño experimental, que puede impactar en la aplicación e instanciación del modelo y métricas de calidad, es importante considerar otros aspectos que pueden influir. Algunos de ellos podrían ser el contexto en el cual se lleva a cabo cada experimento, los mecanismos de recolección y almacenamiento de datos, y los repositorios de datos utilizados.

Por ejemplo, en los experimentos de técnicas de verificación de UdelaR y UPM varias de las diferencias que encontramos entre las métricas aplicadas se deben a que los repositorios de datos utilizados son diferentes. Sin embargo, tanto en los experimentos de UPV como en el de UPM la forma de almacenamiento de los datos es la misma (planillas de cálculo). Esto influye a la hora de instanciar el modelo de calidad y seleccionar las métricas que se aplicarán en cada caso particular, ocasionando que los posibles problemas de calidad que pueden presentarse sean diferentes. A modo de ejemplo, las 4 métricas de calidad que miden la representación e interpretabilidad del conjunto de datos (M18 a M21), se aplican en los 3 experimentos que utilizan planillas de cálculo para registrar sus datos.

Con estos resultados a la vista, podemos concluir que el proceso de recolección de datos (desde los mecanismos de recolección hasta la forma de almacenamiento) impacta en la calidad global de los datos. Esto también es planteado por otros autores [25], [55], [69]. A modo de ejemplo, Disney [69] plantea en esta misma línea que la calidad de datos global de PSP dependerá de la calidad de datos en la etapa de recolección, independientemente de la automatización de otras etapas.

A partir de la Tabla 20 también observamos que la cantidad de sujetos que participan en el experimento podría ser un factor influyente en la cantidad de mediciones ejecutadas y la cantidad de objetos con problemas de calidad. Al ser mayor la cantidad de sujetos, también es mayor la heterogeneidad y la forma de registro de los datos de cada individuo. Esto puede impactar en la cantidad y variedad de problemas de calidad que se presenten. Sin embargo, la cantidad de datos con errores, más allá de la cantidad de sujetos, creemos que depende principalmente de la cantidad y complejidad de los datos que se registran.

9.2 Problemas de calidad presentes en los datos de los experimentos

En la Tabla 22 se muestra la cantidad de problemas de calidad presentes en común entre los 4 experimentos. El detalle de qué problema de calidad está presente en cada experimento se encuentra en la Tabla 24.

Experimentos	UPV-Base	UPV-Replicación	UPM
UdelaR	3	5	5
UPV-Base	N/A	7	6
UPV-Replicación	N/A	N/A	8

Tabla 22: Cantidad de problemas de calidad presentes en común entre experimentos

No identificamos posibles relaciones entre los problemas de calidad que se presentan en los datos de los experimentos y sus características, como ser la cantidad de sujetos, el tipo de experimento u objetivo. Observamos que más allá de las coincidencias que puedan existir entre los diseños experimentales y métricas aplicadas, los problemas de calidad presentados y la cantidad de objetos y datos afectados son diferentes. Esto indica que mientras el diseño experimental podría influir en la instanciación del modelo y métricas de calidad, el resultado de la aplicación del modelo de calidad de datos dependerá del contexto y los datos particulares a cada ejecución. Por ejemplo, la cantidad de problemas de calidad presentes en común es mayor entre UPV-Replicación y UPM, que entre UPV-Base y UPV-Replicación, cuando los diseños experimentales de estos últimos son análogos.

Por otra parte, no encontramos relación posible entre la cantidad de problemas de calidad identificados y la cantidad de objetos y datos que los contienen. A modo de ejemplo, en el experimento de UPM se identifica la menor cantidad de problemas de calidad, pero la mayor cantidad de objetos afectados. En UdelaR se identifica la mayor cantidad de problemas de calidad y de datos afectados. En este último caso, creemos que la cantidad de datos afectados es bastante mayor porque también lo es la cantidad de datos analizados.

A partir del análisis de los problemas de calidad presentes sobre los datos de los 4 experimentos, se pueden identificar algunas de las posibles causas de su ocurrencia. Este análisis permitirá tomar acciones preventivas para futuras experiencias. Algunas de ellas son:

- Ocurrencia de un suceso excepcional, "fuera de lo común" (por ejemplo, outliers).
- Falta de entendimiento de quien registra los datos, sobre cómo determinar o registrar el dato.
- Desconocimiento, por no saber o poder determinar el valor a ingresar, imposibilidad de conocer los datos reales.
- Distracción o equivocación accidental al ingresar un dato de forma manual (ya sea en una herramienta, planilla, hoja de papel).
- Omisión de ingreso de datos (accidental o por no saber determinarlo).
- Defectos o mal funcionamiento de la herramienta de registro o del repositorio donde se almacenan los datos.
- Falta o mala definición de reglas o restricciones sobre el repositorio de datos o las herramientas utilizadas.

Métricas de Calidad (Modelo de Calidad)				Métricas de Calidad aplicadas a los Experimentos				
Dimensión	**Factor**	**Id**	**Métrica**	**Técnicas Verif. (UdelaR)**	**MDD-Base (UPV)**	**MDD-Replic (UPV)**	**Técnicas Verif. (UPM)**	**Cantidad de aplicaciones**
Exactitud	Exactitud Sintáctica	M1	Valor fuera de rango	Sí	Sí	Sí	Sí	**4**
		M2	Falta de estandarización	Sí	Sí	Sí	Sí	**4**
		M3	Valor embebido	No	Sí	Sí	No	**2**
	Exactitud Semántica	M4	Registro inexistente	Sí	No	No	No	**1**
		M5	Registro con errores	Sí	Sí	Sí	Sí	**4**
		M6	Valor fuera de referencial	Sí	No	No	Sí	**2**
	Precisión	M7	Falta de precisión	No	Sí	Sí	Sí	**3**
Completitud	Densidad	M8	Valor nulo	Sí	Sí	Sí	Sí	**4**
		M9	Información omitida	Sí	Sí	Sí	Sí	**4**
	Cobertura	M10	Registro faltante	No	Sí	Sí	Sí	**3**
Consistencia	Integridad de dominio	M11	Regla de integridad de dominio	Sí	Sí	Sí	Sí	**4**
	Integridad intra-relación	M12	Regla de integridad intra-relación	Sí	Sí	Sí	Sí	**4**
		M13	Valor único	Sí	No	No	No	**1**
	Integridad inter-relación	M14	Regla de integridad inter-relación	No	Sí	Sí	Sí	**3**
		M15	Referencia inválida	Sí	No	No	No	**1**
Unicidad	Duplicación	M16	Registro duplicado	Sí	Sí	Sí	Sí	**4**
	Contradicción	M17	Registro contradictorio	Sí	No	No	No	**1**
Representación	Estructura de datos	M18	Estructura de datos	Sí	Sí	Sí	Sí	**4**
	Formato de datos	M19	Formato de datos	No	Sí	Sí	Sí	**3**
Interpretabilidad	Facilidad de entendimiento	M20	Facilidad de entendimiento	No	Sí	Sí	Sí	**3**
	Metadata	M21	Metadata	Sí	Sí	Sí	Sí	**4**
Total		**21**		**15**	**16**	**16**	**16**	

Tabla 23: Métricas de Calidad aplicadas a cada Experimento

Métricas de Calidad				Problemas de calidad presentes en los datos				
Dimensión	**Factor**	**Id**	**Métrica**	**Técnicas Verif. (UdelaR)**	**MDD-Base (UPV)**	**MDD-Replic (UPV)**	**Técnicas Verif. (UPM)**	**Presencias / Aplicaciones**
Exactitud	Exactitud Sintáctica	M1	Valor fuera de rango	0,979	0,968	0,988	0,912	4 / 4
		M2	Falta de estandarización	1,000	0,941	1,000	0,968	2 / 4
		M3	Valor embebido	N/A	0,692	1,000	N/A	1 / 2
	Exactitud Semántica	M4	Registro inexistente	0,979	N/A	N/A	N/A	1 / 1
		M5	Registro con errores	N/E	1,000	1,000	N/E	0 / 4
		M6	Valor fuera de referencial	0,874	N/A	N/A	1,000	1 / 2
	Precisión	M7	Falta de precisión	N/A	1,000	1,000	1,000	0 / 3
Completitud	Densidad	M8	Valor nulo	0,973	1,000	0,959	0,913	3 / 4
		M9	Información omitida	0,999	1,000	0,800	0,896	3 / 4
	Cobertura	M10	Registro faltante	N/A	0,996	0,933	1,000	2 / 3
Consistencia	Integridad de dominio	M11	Regla de integridad de dominio	0,981	1,000	1,000	1,000	1 / 4
	Integridad intra-relación	M12	Regla de integridad intra-relación	0,978	0,971	0,963	0,955	4 / 4
		M13	Valor único	1,000	N/A	N/A	N/A	0 / 1
	Integridad inter-relación	M14	Regla de integridad inter-relación	N/A	0,857	0,926	1,000	2 / 3
		M15	Referencia inválida	0,950	N/A	N/A	N/A	1 / 1
Unicidad	Duplicación	M16	Registro duplicado	0,947	1,000	1,000	1,000	1 / 4
	Contradicción	M17	Registro contradictorio	0,980	N/A	N/A	N/A	1 / 1
Representación	Estructura de datos	M18	Estructura de datos	Regular	Regular	Regular	Regular	4 / 4
	Formato de datos	M19	Formato de datos	N/A	Bueno	Regular	Regular	2 / 3
Interpretabilidad	Facilidad de entendimiento	M20	Facilidad de entendimiento	N/A	Regular	Regular	Regular	3 / 3
	Metadata	M21	Metadata	Aceptable	Regular	Regular	Regular	3 / 4
Total		**21**		**11**	**9**	**10**	**9**	

Tabla 24: Valor de Calidad obtenido por métrica para cada Experimento

9.3 Acciones correctivas aplicadas

Se encuentran diferencias a la hora de identificar y aplicar acciones correctivas sobre los problemas de calidad identificados. En 2 de los 4 casos de aplicación (experimentos UdelaR y UPV-Replicación) se aplicaron correcciones sobre los datos con errores. Esto se debe principalmente al análisis costo-beneficio. Es importante analizar por un lado el costo asociado (si la corrección puede automatizarse o debe realizarse de forma manual), teniendo en cuenta la cantidad y complejidad de los datos afectados. Por otro lado, considerar el beneficio obtenido, teniendo en cuenta si los datos con errores son significativos para los análisis y su corrección podría impactar en los resultados del experimento.

El experimento de UdelaR es el caso en que se realizan la mayor cantidad de correcciones. Esto se debe principalmente a la cantidad y complejidad de los datos (que resulta mucho mayor en este experimento), y al repositorio de datos utilizado. Mientras que en el experimento de UdelaR los datos se registran en una herramienta y se almacenan en una base de datos relacional, en los otros 3 experimentos los datos se recolectan manualmente y se almacenan en una planilla de cálculo. Utilizar una base de datos relacional para almacenar los datos permite automatizar la mayor parte de los procesos de limpieza, obteniendo un mayor beneficio sobre una gran cantidad de datos.

Un factor que pudo influir en que no se aplicaran correcciones sobre los datos en 2 de los 4 experimentos es que las limpiezas a realizar debieran ser manuales. Otro factor es que los análisis estadísticos de ambos experimentos ya habían sido ejecutados. Debido a que la cantidad y complejidad de los datos recolectados es baja (se almacenen en planillas de cálculo), es posible que el experimentador realice correcciones manuales con un bajo costo de forma previa o durante el análisis de los datos del experimento. Sin embargo, esta forma de proceder ad hoc no asegura la identificación ni corrección de los problemas de calidad presentes en los datos, ni resulta repetible en otros casos. De hecho esto también sucede durante el proceso de desarrollo de software: la corrección de defectos de forma ad hoc puede ocasionar que se introduzcan nuevos defectos o que incluso algunos sean ignorados.

9.4 Análisis de los resultados obtenidos a partir de la aplicación del modelo de calidad de datos

En la Tabla 24 se muestran los resultados obtenidos a partir de la aplicación del modelo de calidad sobre los datos de los 4 experimentos. De las 21 métricas de calidad que conforman el modelo, se aplican entre 15 y 16 de ellas en todos los casos. La Ilustración 15 muestra la cantidad de veces (entre 0 y 4) que cada una de las 21 métricas de calidad definidas son aplicadas sobre los datos de los experimentos. Se observa que todas las métricas son aplicadas al menos un vez, y que hay 10 de ellas que son aplicadas en los 4 casos. De esta forma observamos que todas las métricas de calidad definidas resultan instanciables a los datos de experimentos en ingeniería de software.

Observamos también que las métricas aplicadas se encuentran distribuidas por dimensión y factor de calidad, esto es, que no se aplican siempre las métricas de algún determinado factor o dimensión sino que en general se aplican métricas considerando todos estos conceptos.

Por otra parte, observamos que como resultado de la aplicación de las métricas se identifica la presencia de entre 9 y 11 problemas de calidad sobre los datos de los experimentos. Hay 3 problemas de calidad que se encuentran presentes en los 4 casos (M1, M12 y M18). Todos ellos corresponden a diferentes factores y dimensiones de calidad, destacando una vez más la importancia del enfoque

multi-facético adoptado. En el caso de Valor Fuera de Rango, las 4 aplicaciones corresponden a valores sospechosos y solamente en el experimento de UdelaR resultó posible aplicar acciones correctivas para el 19% de los datos afectados. Para la métrica Regla de integridad intra-relación, en los experimentos de UPV corresponde a valores sospechosos mientras que en los experimentos de UdelaR y UPM corresponden a errores en los datos. Solamente se aplica corrección en el caso de UdelaR. Finalmente, la métrica de Estructura de Datos corresponde en los experimentos de UPV y UPM a oportunidades de mejora, mientras que en el UdelaR corresponde a errores en los datos que son corregidos. En este último caso la diferencia se debe fundamentalmente al repositorio de datos utilizado en cada caso.

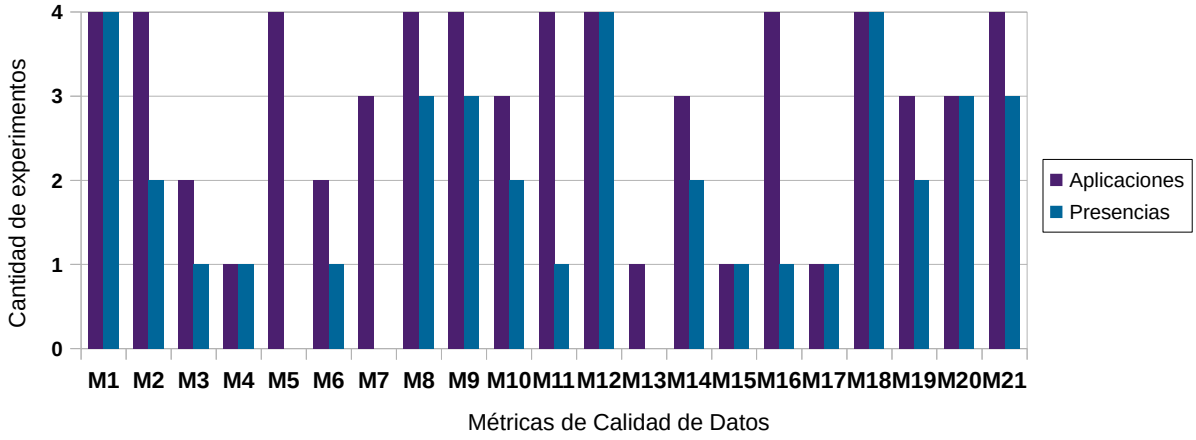

Ilustración 15: Cantidad de métricas aplicadas (Aplicaciones) y problemas de calidad presentes (Presencias) sobre los datos de los 4 experimentos

En resumen, podemos observar a partir de los resultados obtenidos que los experimentos en ingeniería de software presentan diferentes problemas de calidad sobre sus datos, por lo cual no es posible conocer a priori cuáles estarán presentes sobre un caso particular. Es por este motivo que es importante conocer y aplicar el modelo de calidad y la metodología de trabajo de forma disciplinada.

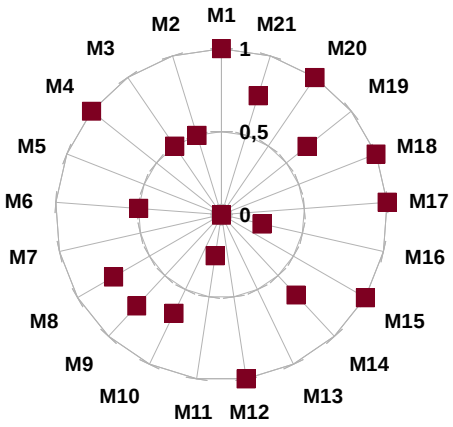

Ilustración 16: Ratio de efectividad (presencias/aplicaciones) por Métrica de Calidad

En la Ilustración 16 se muestra el ratio de efectividad por métrica de calidad, calculado como la cantidad de veces que se detecta la presencia de un problema de calidad sobre la cantidad de veces

que la métrica es aplicada. Para 7 métricas sucede que el valor del ratio es 1.0, esto es, que siempre que la métrica es aplicada se detecta la presencia de un problema de calidad en los datos. Hay 3 métricas con ratio igual a 0 (M5, M7 y M13), indicando que los problemas de calidad asociados a esas métricas no se presentaron sobre ninguno de los casos analizados. En el caso de la métrica M5, la cual fue aplicada en los 4 casos pero en ninguno de ellos se detecta la presencia de una problema de calidad, hay 2 de las 4 ocasiones en las que no fue posible ejecutar la medición por motivos del alto esfuerzo asociado.

Por otra parte, en la Ilustración 17 muestra el ratio de efectividad agrupado por dimensión de calidad, considerando los 4 casos de aplicación. De esta forma podemos observar que se presentan problemas de calidad asociadas a todas las dimensiones planteadas. Las dimensiones con un mayor ratio de efectividad son aquellas asociadas al conjunto de datos (Representación e Interpretabilidad). Esto se debe principalmente a que en 3 de los 4 casos los repositorios de datos utilizados correspondían a planillas de cálculo, por lo que resultó importante considerar las métricas y problemas de calidad asociados a estas dimensiones.

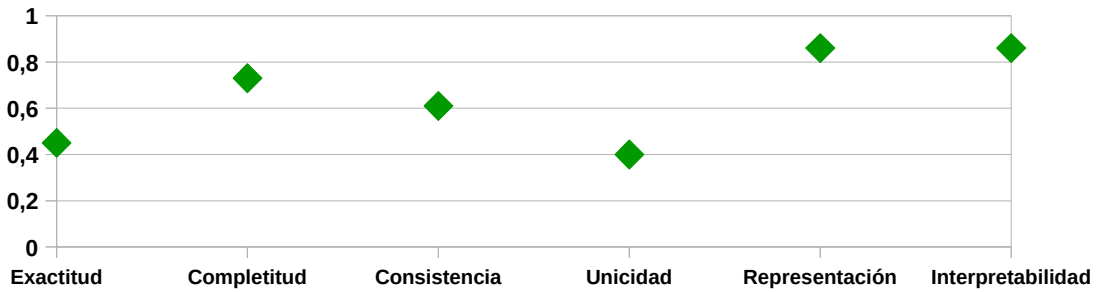

Ilustración 17: Ratio de efectividad (presencias/aplicaciones) por Dimensión de Calidad

9.4.1 Esfuerzo dedicado

En la Ilustración 18 se muestra el esfuerzo (en horas) que invierte cada uno de los roles participantes en la metodología de aplicación del modelo de calidad de datos, por cada experimento. En todos los casos el Analista de Calidad es quien dedica la mayor parte del esfuerzo (entre un 85 y un 93%), ya que participa durante todas las fases de la metodología propuesta y ejecutando la mayor parte de las actividades. La participación del experimentador es en general a demanda del analista. Tanto en las actividades de validación como de generar el conocimiento del experimento el aporte del experimentador es fundamental. Entendemos que es importante que quien cumple el rol de analista de calidad no sea el propio experimentador. De esta forma, quien identifica las métricas y problemas de calidad puede conocer y analizar la realidad del experimento desde un punto de vista más objetivo y con foco en los datos.

Como se muestra en laIlustración 18, en el caso de UdelaR el esfuerzo es significativamente mayor a los otros 3 experimentos. Esto se debió a varios motivos: fue el primer caso de aplicación (parte del esfuerzo puede deberse al propio aprendizaje); participaron 2 analistas de calidad (incrementa el esfuerzo en las actividades de coordinación); la cantidad y complejidad de datos es considerablemente mayor; se desarrollaron y ejecutaron limpiezas automáticas sobre la mayoría de los errores identificados.

Considerando un análisis costo-beneficio, el esfuerzo invertido permite obtener como principal aporte un análisis, evaluación y conocimiento del nivel de calidad que tienen los datos del experimen-

to, de forma de incrementar la confianza en los resultados obtenidos. Por otra parte, cada caso de aplicación permitió la retroalimentación y mejora del modelo de calidad propuesto como base.

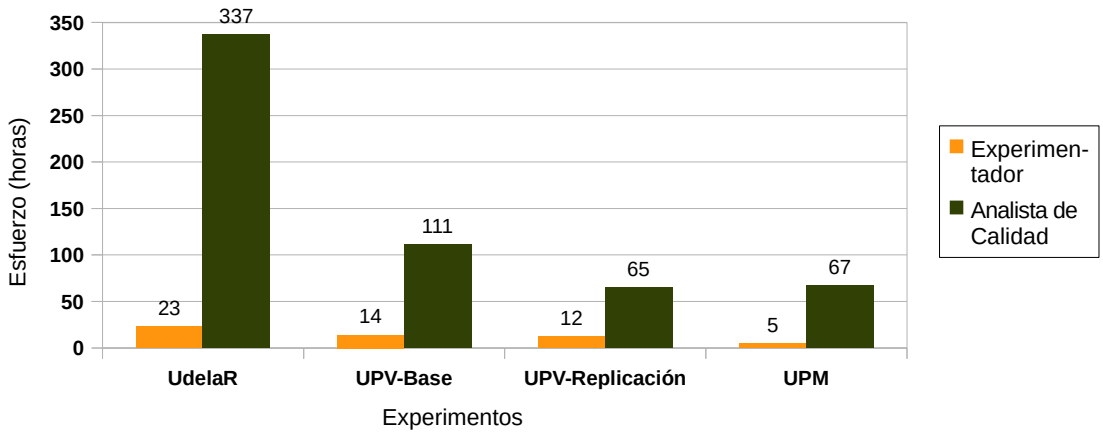

Ilustración 18: Esfuerzo dedicado por Rol por Experimento

Creemos que la cantidad y complejidad de los datos, así como la forma de almacenamiento de los mismos podría ser un indicador para estimar el esfuerzo que se invertirá en el trabajo de calidad de datos. Según nuestra experiencia, y dependiendo de los procesos de corrección que se apliquen y la forma que se lleven a cabo (manual o automáticamente), en la fase de limpieza de datos en general se invierte una cantidad significativa de horas. Esto también es planteado por otros autores: en general entre un 60 y 95% del esfuerzo dedicado al análisis de los datos se invierte en la limpieza de los mismos [37]. En particular, en UdelaR se aplican correcciones sobre casi la totalidad de los problemas identificados, resultando en que el esfuerzo dedicado en este fase (39% del total) corresponde al mayor de las 4 fases. En UPV-Replicación, sin embargo, no se aplican correcciones para todos los problemas de calidad, resultando en un esfuerzo bastante menor (19%).

9.5 Experiencia de los experimentadores

Se realizó una encuesta entre los experimentadores para conocer su opinión y experiencia respecto al uso y aplicación del modelo y la metodología de calidad. El cuestionario de satisfacción fue construido en base al framework propuesto por Moody [93], [94]. Dicho framework ha sido validado previamente y es ampliamente utilizado para evaluar la calidad de los modelos o métodos en términos de tres variables de satisfacción:

- *Usabilidad percibida (PU)*: es el grado en que una persona cree que una representación particular del método será efectiva para alcanzar los objetivos propuestos.
- *Facilidad de uso percibida (PEOU)*: es el grado en que una persona cree que utilizar un método particular será libre de esfuerzo.
- *Intención de uso (ITU)*: es el grado en que un individuo tiene la intención de utilizar un método particular.

Siguiendo el *framework* definimos 8 preguntas para medir la PU, 6 para la PEOU y 2 para la ITU. Además, cada experimentador debe indicar 5 aspectos positivos y 5 aspectos negativos respecto al uso y aplicación del modelo de calidad. El cuestionario resultante así como las respuestas obtenidas por parte de los experimentadores se encuentran en el Anexo D. Cada experimentador responde

a las 16 preguntas indicando un valor de 1 a 5, donde 1 indica "totalmente en desacuerdo" y 5 "totalmente de acuerdo".

La Ilustración 19 muestra el resultado obtenido para cada una de las variables de satisfacción (PU, PEOU, ITU), considerando los experimentos de UdelaR y UPV. Para el experimento de UPM, el cuestionario fue enviado pero no se obtuvo respuesta por parte del experimentador responsable.

En todos los casos los valores obtenidos indican una opinión positiva por parte de los experimentadores respecto al modelo y la metodología. Los valores más altos corresponden a la variable PU, lo cual indica que los responsables de los experimentos entienden que el modelo, las métricas de calidad y la metodología son efectivas en alcanzar el objetivo propuesto. La variable PEOU es la que tiene los menores valores en ambas respuestas. Esto indica que los experimentadores creen que aplicar el modelo y métricas de calidad requiere un esfuerzo no despreciable. Finalmente, la variable ITU indica que tienen intención de volver a aplicar el modelo y métricas de calidad en otras experiencias.

Esta última variable muestra la importancia y valor agregado para los experimentadores tanto de la aplicación del modelo como de la metodología de calidad propuesta. A pesar del esfuerzo invertido durante su aplicación, los experimentadores valoran el beneficio obtenido. Esta retroalimentación positiva también nos demuestra que estamos transitando por el camino adecuado hacia la evaluación y mejora de la calidad de los datos de experimentos en ingeniería de software. Los experimentadores deberían ser los principales interesados en mejorar la confianza en los resultados de sus experimentos. Por este motivo, es fundamental que sean conscientes de la importancia de contar con herramientas y métodos como los propuestos en este trabajo, que les ayuden a lograr este objetivo.

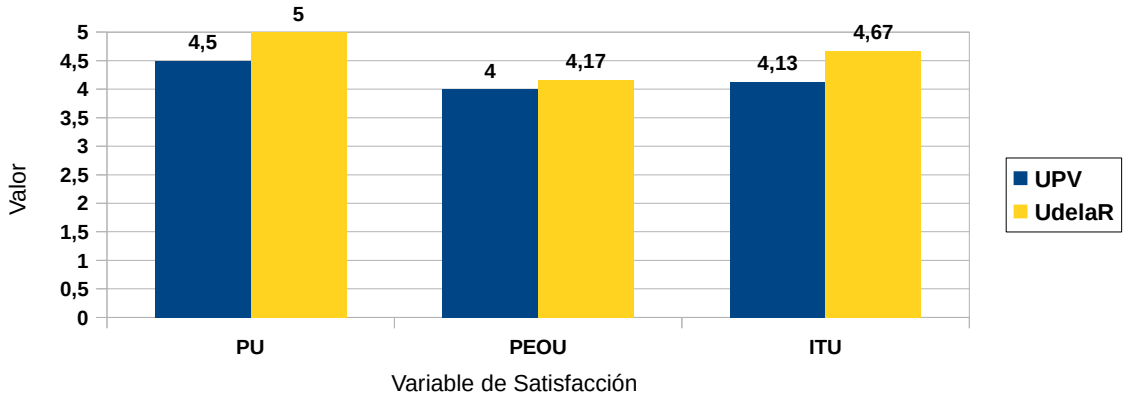

Ilustración 19: Resultado de las variables de satisfacción

A continuación se resumen los principales aspectos positivos y negativos mencionados por los experimentadores.

Los aspectos positivos planteados muestran una vez más el entendimiento de la importancia y valor agregado del modelo y metodología de trabajo para los experimentadores:

- *Métricas preestablecidas ayudan a encontrar problemas en la calidad.*
- *Cada experimentador puede adaptar el modelo a su experimento.*
- *Ayuda a garantizar la calidad de los datos ante terceras personas.*
- *Da ideas de mejora para futuras replicaciones.*
- *Método riguroso en contraposición a uno ad hoc, que acompaña al modelo para una correcta aplicación del mismo y que lleva también a encontrar problemas que podrían ser pasados por alto.*

- *Punto de partida desde la disciplina de la calidad de datos permitiendo esto tener una visión mucho más amplia sobre la calidad de datos de la que normalmente tienen los experimentadores.*
- *Se logran buenos resultados encontrándose problemas de calidad en la práctica, es aplicable y sirve. Esto impacta positivamente en la credibilidad de los datos de los experimentos.*

Los aspectos negativos planteados son importantes para identificar posibles mejoras al modelo y a la metodología de trabajo definida. Esta retroalimentación constituye un aspecto fundamental en el ciclo de mejora continua para el desarrollo del modelo de calidad de datos, presentado en el Error: Reference source not found. A continuación se presentan los aspectos planteados y una breve discusión sobre cada uno de ellos:

- *Solo encuentran problemas de calidad a posteriori de tener los datos y no durante el proceso de experimentación. Sería bueno contar con una aplicación temprana de alguna parte del Modelo donde se atacan problemas de calidad de datos antes de comenzar la ejecución del experimento. Es decir, que el Modelo y el Método ayuden en la prevención de los problemas de calidad de datos.*

Según la metodología propuesta, el modelo de calidad se aplica sobre los datos de los experimentos luego de la ejecución del mismo. Sin embargo, las primeras fases de la metodología (1 y 2) puede realizarse una vez se conozca el diseño del experimento, los datos que se recolectarán y los repositorios donde se almacenarán. Al realizar la instanciación del modelo y métricas de calidad antes de ejecutar el experimento, es posible pensar en ciertos problemas de calidad que podrían presentarse sobre los datos de ese experimento, y de esta forma aplicar acciones preventivas.

Por otra parte, si el experimento a analizar consiste en la replicación de otro que ya fue analizado, entonces existirán acciones de prevención propuestas (como resultado del trabajo de calidad ejecutado sobre el experimento base) que es posible aplicar en la replicación.

De todas formas como trabajo a futuro se podría pensar en una extensión del modelo y metodología de trabajo que incluya actividades más genéricas de prevención de problemas de calidad, y que puedan aplicarse sobre datos de experimentos en ingeniería de software.

- *Puede que haya métricas subjetivas difíciles de clasificar entre error o acierto.*

Por la propia naturaleza subjetiva de la calidad de datos existen también ciertas métricas que se definen de forma subjetiva, lo cual dificulta la clasificación de los errores de calidad.

En nuestro trabajo, las métricas "subjetivas" son las que encuentran problemas de calidad que denominamos "valores sospechosos" u "oportunidades de mejora", justamente porque no podemos estar seguros si esos problemas corresponden o no a errores reales sobre los datos. Los casos de valores sospechosos son analizados de forma separada, y requieren de un esfuerzo adicional para conocer si se tratan de errores en los datos o simplemente datos anómalos.

- *Para gente que no tenga experiencia en el mundo empírico, el aplicar la técnica le puede llevar bastante tiempo de aprendizaje. Es bastante costoso también para el analista de calidad.*

Como se observa en la Ilustración 19 el tiempo invertido por el analista de calidad puede ser alto dependiendo de factores como: su experiencia y conocimiento del modelo y metodología de trabajo, la cantidad y complejidad de los datos, los repositorios de datos utilizados, si se aplican o no acciones correctivos, entre otros. En el caso del experimentador, debido a que su participación es más a demanda y no necesita conocer en detalle la metodología y métricas de calidad, con apoyo del analista puede comprender con un bajo esfuerzo los aspectos más importantes.

Por otra parte, la aplicación del modelo puede resultar en un ahorro de esfuerzo invertido en otras actividades. Por ejemplo, si se detectan problemas de calidad luego de realizar los análisis estadísticos de los experimentos que deben ser corregidos, entonces podría ser necesario ejecutarlos nuevamente.

Como trabajo a futuro se podría pensar en formas de automatización de la metodología de aplicación del modelo de calidad de datos, de forma de disminuir el esfuerzo dedicado por el analista de calidad. Sin embargo esto no parece una tarea sencilla, considerando que la instanciación del modelo de calidad depende de cada caso particular, y la implementación de las mediciones depende del repositorio utilizado.

- *El Método y el Modelo aún están inmaduros y en etapa de construcción, no han sido publicados y por ende la comunidad de experimentadores en ingeniería de software no lo ha discutido en profundidad. El Modelo no está lo suficientemente validado aún.*

Sí, el método y el modelo se encuentran aún en proceso de ajustes y construcción, es un camino que estamos transitando. Nuestro objetivo es que el modelo y método propuestos sean utilizados, validados y complementados por el aporte de la comunidad empírica en ingeniería de software. Los resultados obtenidos hasta el momento muestran que tanto el modelo como la metodología son de utilidad y beneficio para mejorar la confianza en los resultados de los experimentos, y que los experimentadores también valoran este aporte.

9.6 Resumen

En vista de los resultados obtenidos, observamos que los datos recolectados durante la ejecución de experimentos en Ingeniería de Software que involucran sujetos humanos contienen problemas de calidad. Identificamos además que muchos de estos problemas pueden impactar directamente en los resultados de los experimentos.

Para todos los experimentos se encuentra que entre entre un 55 y 75% de las métricas aplicadas dan como resultado la presencia de algún problema de calidad sobre sus datos. Esto muestra que la calidad de datos debe ser atendida y considerada en este dominio de aplicación.

Así como la calidad de datos se define de manera multi-facética (en función de las diferentes facetas o dimensiones de la calidad), los problemas de calidad identificados sobre los datos en este dominio también se distribuyen entre las diferentes dimensiones. Un hecho a destacar es que la exactitud (o correctitud) de los datos, que suele ser una de las dimensiones más consideradas por los diferentes autores (y en muchas oportunidades considerada de forma individual) es una de las que se presenta en menor medida. Aparecen otras dimensiones que no son generalmente abordadas (como la Consistencia o Interpretabilidad), y que merecen la misma atención.

Por otra parte, considerando que el contexto en cual se ejecuta cada experimento es diferente, los problemas de calidad que se presenten en cada caso pueden ser diferentes. A modo de ejemplo, en nuestros casos de aplicación las dimensiones sobre las cuales se identifica una mayor cantidad de problemas de calidad son las que refieren a la estructura y representación de los datos, particularmente porque 3 de los 4 experimentos utilizaron planillas de cálculo. Sin embargo, en contextos de características diferentes los problemas de calidad que sean más relevantes pueden ser diferentes. Este es principal motivo porque el cual es importante considerar la visón multi-facética de la calidad de datos que proponemos en este trabajo.

En líneas generales, podemos decir que la calidad de los datos que son recolectados durante la ejecución de experimentos en Ingeniería de Software debe ser cuestionada y analizada antes de obtener los resultados de los experimentos.

Capítulo 10: Conclusiones y Trabajos a Futuro

En este capítulo se presentan las conclusiones de la tesis, los aportes del trabajo, las limitaciones encontradas y los trabajos planteados a futuro.

10.1 Conclusiones

La experimentación en Ingeniería de Software permite confirmar con hechos de la realidad ciertas teorías, suposiciones y creencias acerca de distintos aspectos del software. Por ejemplo, permite establecer qué técnicas, métodos y herramientas son más efectivas ya sea para verificar o desarrollar un software bajo determinadas situaciones.

Durante la ejecución de experimentos controlados en Ingeniería de Software se generan una gran cantidad de datos que son utilizados para obtener los resultados y las conclusiones del mismo. Si los datos en los cuales se basan los análisis estadísticos no son de buena calidad, los resultados de los experimentos pueden ser incorrectos.

La calidad de los datos es un tema fundamental en cualquier contexto donde se generen y utilicen datos, existiendo hace ya muchos años el área de investigación Calidad de Datos dentro de la comunidad de Sistemas de Información y Bases de Datos.

Sin embargo, este tema no ha sido atendido con la importancia que se merece en el contexto de la Ingeniería de Software Empírica, menos aún para experimentos. Sorprendentemente, en esta comunidad no se presentan ni aplican protocolos o métodos sistemáticos ni explícitos para evaluar y mejorar la calidad de los datos que se utilizan.

El objetivo general de nuestro trabajo es proponer un Modelo de Calidad de Datos que pueda ser aplicado específicamente en el dominio de experimentos en ingeniería de software, y una metodología de trabajo sistemática, disciplinada y estructurada que utilice dicho modelo para evaluar y mejorar la calidad de sus datos. También nos propusimos utilizar el modelo y la metodología en experimentos controlados particulares.

En este trabajo desarrollamos un modelo de calidad que está compuesto de 21 métricas de calidad que proporcionan la base para medir, conocer, evaluar y mejorar la calidad de los datos recolectados durante un experimento. Las métricas definidas constituyen un instrumento fundamental para identificar la presencia de problemas de calidad sobre los datos, y así aplicar las acciones correctivas o preventivas que correspondan con el fin de mejorar su calidad.

Conocer y aplicar las métricas de calidad contribuye en que el experimentador sea consciente de ciertas mejoras que pueden introducirse y que contribuirán en mejorar la calidad global de los datos del experimento y de los resultados obtenidos. El conocimiento y la experiencia que adquiere el experimentador a partir de la aplicación del modelo de calidad podrá ser aplicada en sus futuras experiencias empíricas.

El modelo de calidad está basado en conceptos, técnicas y herramientas definidas en el área de Calidad de Datos. Las dimensiones y factores de calidad considerados así como sus definiciones están basados en los conceptos conocidos y más referenciados sobre los cuales existen un consenso en la literatura del área. A diferencia de la mayoría de los trabajos de calidad de datos que encontramos para el dominio de ingeniería de software empírica, la base conceptual de nuestro trabajo se centra en los conceptos propuestos por un área de investigación desarrollada y aplicada también en otros dominios.

En este sentido, consideramos que la adopción del enfoque multi-facético de la calidad de datos incrementa la probabilidad de que se encuentran una mayor cantidad y variedad de problemas de calidad sobre los datos de los experimentos. Si se limita la evaluación de los datos a determinados aspectos de calidad (tales como *outliers* o valores faltantes), algunos problemas de calidad seguramente serán ignorados.

Una característica importante del trabajo realizado es que el modelo de calidad propuesto fue construido siguiendo una metodología de investigación planteada como parte de este trabajo. Dicho modelo se en-

cuentra inmerso en un ciclo de mejora continua, y es refinado, ajustado y evaluado a partir de cada aplicación sobre los datos de distintos experimentos de ingeniería de software.

De forma de asegurar que el modelo de calidad sea aplicado de forma sistemática, disciplinada y estructurada proponemos una metodología de trabajo que define los pasos y guías para aplicar el modelo de calidad a los datos de experimentos particulares en ingeniería de software. La metodología está compuesta de cuatro fases: 1. Generar conocimiento del experimento, 2. Instanciar el modelo de calidad de datos, 3. Evaluar la calidad de datos y 4. Ejecutar acciones correctivas sobre los datos. Para cada fase se identifican los artefactos de entrada a la misma, las técnicas, herramientas y actividades principales que se utilizan y realizan, así como los artefactos de salida generados. Debido a que la metodología definida es genérica, creemos que podría ser aplicada sobre cualquier experimento con las características mencionadas.

Tanto la metodología de trabajo como el modelo de calidad propuestos fueron aplicados sobre los datos de cuatro experimentos particulares en ingeniería de software. De esta forma se logró conocer, analizar y mejorar la calidad de los datos de los experimentos bajo estudio. Además, se mostró que el modelo y metodología de calidad de datos propuestos son instanciables a casos (experimentos) particulares. A partir de cada aplicación, se obtiene retroalimentación que fue utilizada para ajustar y mejorar el modelo (tal como propone la metodología de investigación), de forma que pueda ser aplicado también en futuras experiencias.

Consideramos que aún en los casos en que no se identifican errores en los datos o no se aplican acciones correctivas existe un beneficio importante en la aplicación del modelo y metodologías propuestos. Aplicar la metodología permite que los experimentadores conozcan y sean conscientes de cuál es el nivel de calidad de los datos que utilizan, y que de esta forma puedan tener confianza en que los resultados que obtienen en base a esos datos serán correctos.

Mediante la aplicación del modelo y metodología de trabajo concluimos que el enfoque multi-facético es importante para analizar la calidad de los datos de los experimentos. Las 21 métricas de calidad que conforman el modelo de calidad propuesto fueron aplicadas en alguno de los casos de aplicación. De esta forma vemos que todas las métricas de calidad definidas resultan instanciables a los datos de experimentos en ingeniería de software. Además, observamos que no se aplican siempre las métricas de algún determinado factor o dimensión, sino que en general se aplican métricas considerando todos estos conceptos.

Como resultado de la aplicación de las métricas se identifica la presencia de entre 9 y 11 problemas de calidad sobre los datos de cada experimento. Todos ellos corresponden a diferentes factores y dimensiones de calidad, destacando una vez más la importancia del enfoque multi-facético adoptado.

Debido a que los experimentos en ingeniería de software presentan diferentes problemas de calidad sobre sus datos, no es posible conocer a priori cuáles estarán presentes sobre un caso particular. Por este motivo, es importante conocer y aplicar el modelo de calidad y la metodología de trabajo de forma disciplinada.

Si bien la aplicación de la metodología y modelo de calidad tienen un esfuerzo asociado, al considerar un análisis costo-beneficio el esfuerzo invertido permite obtener como principal aporte un análisis, evaluación y conocimiento del nivel de calidad que tienen los datos del experimento, de forma de incrementar la confianza en los resultados obtenidos.

Finalmente, se realizó una encuesta entre los experimentadores para conocer su opinión y experiencia respecto al uso y aplicación del modelo y la metodología de calidad. En todos los casos los valores obtenidos indican una opinión positiva por parte de los experimentadores respecto al modelo y la metodología. Los responsables de los experimentos entienden que el modelo, las métricas de cali-

dad y la metodología son efectivas en alcanzar el objetivo propuesto, a pesar de que requiere un esfuerzo no despreciable. Más importante aún, expresan su intención de volver a aplicar el modelo y métricas de calidad en otras experiencias. Esto es fundamental para que el modelo y metodología continúen siendo aplicadas y evolucionando en cada aplicación.

Luego de aplicada la metodología propuesta sobre los datos de cuatro experimentos, concluimos que tanto la metodología como el modelo de calidad definidos cumplen con los objetivos establecidos, contribuyendo en la evaluación y mejora de la calidad de los datos analizados. Mostramos que es posible su aplicación sobre casos concretos, y que la misma resulta en un beneficio importante para el experimento, el experimentador, y para la comunidad en ingeniería de software empírica en general. Los resultados obtenidos nos permiten también posicionar nuestra propuesta de forma tal que pueda ser aplicada sobre los datos de otros experimentos en ingeniería de software que involucran sujetos humanos.

Observamos también que el esfuerzo invertido durante la aplicación no es excesivo, principalmente en relación al importante beneficio obtenido. Los experimentadores también demostraron su amplia aceptación respecto al enfoque propuesto, considerando que es aplicable y ampliamente beneficioso en este dominio.

Por todo esto, consideramos que nuestro trabajo constituye un aporte importante tanto para la comunidad en Ingeniería de Software Empírica como para la de Calidad de Datos, y en particular, muestra un avance en el tema de calidad de datos para experimentos en ingeniería de software.

10.2 Aportes del trabajo

Los aportes de este trabajo de tesis son los siguientes:

- La definición de un modelo de calidad de datos que permite evaluar y mejorar la calidad de los datos resultantes de experimentos en ingeniería de software que utilizan humanos como sujetos. No encontramos en la literatura ningún modelo como el que hemos desarrollado. Esto va en línea con lo mencionado por *Liebchen*: desarrollar un protocolo unificado (modelo en este caso) para la calidad de datos para la ingeniería de software empírica.
- La definición de una metodología sistemática, disciplinada y estructurada que define los pasos y guías para aplicar el modelo de calidad propuesto a los datos de un experimento particular en ingeniería de software. Esto permitirá que el modelo pueda ser utilizado por distintos investigadores en ingeniería de software de distintas partes del mundo. La metodología propuesta permite utilizar fácilmente el modelo si bien requiere que quien lo use tenga conocimientos de calidad de datos.
- La aplicación del modelo y metodología de calidad propuestos sobre los datos de 4 experimentos en ingeniería de software. Mediante estos casos de estudio pudimos apreciar los errores en los datos que el modelo es capaz de encontrar, estudiar la aplicación de la metodología en casos reales y evaluar así su costo beneficio. Esto nos permitió confirmar las bondades de nuestras propuestas.

En la actualidad, la Ingeniería de Software no cuenta con un nivel de desarrollo en materia de experimentación formal comparable con otras disciplinas de la Ingeniería. En particular, la calidad de los datos que son recolectados en experimentos controlados en ingeniería de software no ha sido muy estudiada. Menos aún se han estudiado modelos y metodologías que permitan evaluar la calidad de estos datos.

El modelo, la metodología y el haberlos aplicado en experimentos reales es una contribución a la estandarización de la forma de evaluar la calidad de los datos en experimentos controlados en ingeniería de software. El seguimiento y aplicación de una metodología y enfoque ordenado como el

propuesto en este trabajo, no solo maximiza la cantidad de problemas de calidad que pueden identificarse sobre los datos, sino que también permite que sea repetible en otros estudios de características similares. Por ende, es también una contribución en el desarrollo general de la experimentación en ingeniería de software.

10.3 Limitaciones

Una limitación de nuestra propuesta es con respecto a la generalidad alcanzada por el modelo construido, siendo difícil establecer el grado de la misma. Debido a que un modelo de calidad se define para ser aplicado sobre los datos de un dominio particular, no podemos afirmar si el modelo y la metodología que definimos en este trabajo podrán ser aplicados sobre datos generados en otras áreas de la ingeniería de software, o incluso de la ingeniería de software empírica. Además, cada contexto tiene particularidades que es necesario considerar ya que estas podrán limitar el dominio de aplicación del modelo presentado. Por ejemplo, no podemos asegurar que el modelo y metodología propuestos puedan ser aplicados sobre los datos resultantes de casos de estudio o encuestas en el dominio de la ingeniería de software.

10.4 Trabajos a futuro

Tanto el método como el modelo propuestos en este trabajo se encuentran aún en proceso de ajustes y construcción. El objetivo final es lograr generalizarlos para que puedan ser aplicables sobre los datos de cualquier experimento en ingeniería de software. Para esto, debemos continuar trabajando para que el modelo y método propuestos sean utilizados, validados y complementados por el aporte de la comunidad empírica en ingeniería de software.

También podemos pensar que este modelo y metodología podrían ser aplicados sobre los datos resultantes de la ejecución de procesos en ingeniería de software. De la misma forma que sucede con los experimentos, los procesos durante su ejecución recolectan datos que serán utilizados para obtener conclusiones, análisis y tomar decisiones a partir de los mismos (por ejemplo, mejorar las estimaciones futuras en base a datos históricos). Los datos que se recolectan durante un proceso de desarrollo de software son "similares" a los datos recolectados durante los experimentos en ingeniería de software. Una primera aproximación a este tema fue la aplicación de las métricas de calidad definidas sobre los datos recolectados durante la ejecución del Personal Software Process [33]. Como resultado, se encontró que estos datos también contienen problemas de calidad que deben ser corregidos. No menos importante, mostramos con un caso concreto que es posible la aplicación del modelo de calidad definido sobre los datos resultantes de un proceso de desarrollo de software. Como trabajo a futuro pretendemos utilizar nuestra propuesta en datos provenientes de otros procesos de desarrollo.

Un aporte importante al modelo y metodología definidos podría ser la inclusión de actividades más genéricas de prevención de problemas de calidad, que puedan aplicarse antes de la ejecución de los experimentos. Esto apunta a prevenir la ocurrencia de los problemas de calidad antes de llegar a corregirlos. Para esto, sería necesario identificar qué problemas suelen presentarse en estos experimentos, y de qué forma podrían ser evitados.

Considerando las diferentes fases del proceso experimental, y que el foco de esta trabajo está en los datos generados a partir de la operación de los experimentos, un trabajo a futuro podría considerar la aplicación de la metodología o modelo de calidad de datos desde etapas más tempranas. Por ejemplo, identificando requisitos de calidad de datos durante las diferentes fases del experimento (desde su diseño o planificación). En este caso, se debería analizar cómo mejorar, complementar o ex-

tender el modelo y la metodología planteadas, de forma que su alcance abarque el proceso experimental completo.

Por otra parte, y debido al esfuerzo no despreciable que insume su aplicación, se podrían plantear formas de automatización de la metodología de aplicación del modelo de calidad de datos. Esto no parece una tarea sencilla, considerando que la instanciación del modelo de calidad depende de cada caso particular, y la implementación de las mediciones depende del repositorio utilizado.

Con respecto a la metodología de aplicación del modelo, la estandarización de los reportes de entrada y salida podría incrementar la posibilidad de generalización de la misma, de forma de que puedan ser usados para facilitar el trabajo del experimentador.

Bibliografía

[1] M. Shepperd, "Data quality: Cinderella at the Software Metrics Ball?," in *Proceeding of the 2nd international workshop on Emerging trends in software metrics*, 2011, pp. 1–4.

[2] G. A. Liebchen, "Data Cleaning Techniques for Software Engineering Data Sets (Doctoral Thesis)," Brunel University, 2010.

[3] A. J. E. Bachmann, "Why Should We Care about Data Quality in Software Engineering? (Doctoral Thesis)," University of Zurich, 2010.

[4] J. Radatz, A. Geraci, and F. Katki., "IEEE standard glossary of software engineering terminology," vol. 121990, 1990.

[5] R. Shackelford, A. McGettrick, R. Sloan, H. Topi, G. Davies, R. Kamali, J. Cross, J. Impagliazzo, R. LeBlanc, and B. Lunt, "Computing Curricula 2005 - The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering," 2006.

[6] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[7] N. Juristo and A. M. Moreno, *Basics of Software Engineering Experimentation*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[8] R. Y. Wang, H. B. Kon, and S. E. Madnick, "Data quality requirements analysis and modeling," in *Proceedings of IEEE 9th International Conference on Data Engineering*, 1993, pp. 670–677.

[9] C. Batini and M. Scannapieco, *Data quality: concepts, methodologies and techniques*. Springer, 2006.

[10] D. Strong, Y. Lee, and R. Wang, "Data Quality in Context," *Commun. ACM*, vol. 40, no. 5, pp. 103–110, 1997.

[11] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Commun. ACM*, vol. 45, no. 4, p. 211, Apr. 2002.

[12] R. Wang and D. Strong, "Beyond accuracy: What data quality means to data consumers," *J. Manag. Inf. Syst.*, pp. 5–34, 1996.

[13] M. Scannapieco and T. Catarci, "Data quality under a computer science perspective," in *Archivi & Computer*, 2002, pp. 1–12.

[14] G. Liebchen and M. Shepperd, "Data sets and data quality in software engineering," in *Proceedings of Predictor Models in Software Engineering (PROMISE 2008)*, 2008, pp. 39–44.

[15] G. Liebchen, B. Twala, M. Shepperd, M. Cartwright, and M. Stephens, "Filtering, Robust Filtering, Polishing: Techniques for Addressing Quality in Software Data," in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, 2007, pp. 99–106.

[16] T. C. Redman, *Data Quality for the Information Age*, 1st ed. Norwood, MA, USA: Artech House, Inc., 1997.

[17] Y. Wand and R. Wang, "Anchoring Data Quality dimensions in Ontological Foundations," *Commun. ACM*, vol. 39, no. 11, pp. 86–95, 1996.

[18] T. Menzies, A. Brady, and E. Kocaguneli, "What is 'Enough' Quality for Data Repositories?," *Softw. Qual. Prof.*, vol. 13, no. 2, pp. 42–51, 2011.

[19] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–52, Jul. 2009.

[20] H. Harper and D. Zubrow, "Editors' Introduction: Should You Trust Your Data?," *Softw. Qual. Prof.*, vol. 13, no. 4, pp. 4–8, 2011.

[21] S. Mcgraw and S. E. Measurement, "The Importance of Data Quality," in *SEI Podcast Series*, pp. 1–6.

[22] M. Kasunic, J. McCurley, and D. Zubrow, "Can you trust your data? establishing the need for a measurement and analysis infrastructure diagnostic," 2008.

[23] A. Bachmann and A. Bernstein, "When process data quality affects the number of bugs: Correlations in software engineering datasets," in *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, 2010, pp. 62–71.

[24] K. Chen, S. R. Schach, L. Yu, J. Offutt, and G. Z. Heller, "Open-Source Change Logs," in *Empirical Software Engineering*, 2004, vol. 9, no. 3, pp. 197–210.

[25] M. Bosu and S. MacDonell, "Data quality in empirical software engineering: a targeted review," in *Proceedings of Evaluation and Assessment in Software Engineering (EASE 2013)*, 2013, pp. 171–176.

[26] M. Bosu and S. MacDonell, "A Taxonomy of Data Quality Challenges in Empirical Software Engineering," in *Porceedings of Australian Software Engineering Conference (ASWEC 2013)*, 2013, pp. 97–106.

[27] G. Liebchen and B. Twala, "Assessing the quality and cleaning of a software project dataset: an experience report," in *Proceedings of Evaluation and Assessment in Software Engineering (EASE 2006)*, 2006, pp. 1–7.

[28] A. Bachmann, C. Bird, and F. Rahman, "The missing links: bugs and bug-fix commits," in *Proceedings of the FSE-18*, 2010, pp. 97–106.

[29] W. E. Deming, *Calidad, Productividad y Competitividad: la salida de la crisis*. Madrid: Ediciones Díaz de Santos, 1989.

[30] C. Valverde, D. Vallespir, A. Marotta, and J. I. Panach, "Applying a Data Quality Model to Experiments in Software Engineering," in *QMMQ Workshop, ER 2014*, 2014, pp. 168–177.

[31] W. Humphrey, *PSP: A Self-improvement Process for Software Engineers*, First. Addison-Wesley Professional, 2005.

[32] C. Valverde, A. Marotta, and D. Vallespir, "Análisis de la calidad de datos en experimentos en ingeniería de software," in *XVIII Congreso Argentino de Ciencias de la Computación*, 2012, pp. 794–803.

[33] C. Valverde, F. Grazioli, and D. Vallespir, "Un Estudio de la Calidad de los Datos Recolectados durante el Uso del Personal Software Process," in *JIISIC 2012*, 2012.

[34] P. B. Crosby, *Quality without tears: The art of hassle free management*. New York, NY [u.a.]: McGraw-Hill, 1984.

[35] B. Otto, K. Hüner, and H. Österle, "Identification of Business Oriented Data Quality Metrics," in *ICIQ'09*, 2009.

[36] J. M. Juran, *Quality control handbook*. New York: McGraw-Hill, 1951.

[37] R. D. De Veaux and D. J. Hand, "How to Lie with Bad Data," *Stat. Sci.*, vol. 20, no. 3, pp. 231–238, Aug. 2005.

[38] M. Gertz and G. Saake, "Report on the Dagstuhl Seminar 'Data Quality on the Web,'" *SIGMOD*, vol. 33, no. 1, pp. 127–132, 2004.

[39] S. A. Knight, "The combined conceptual life-cycle model of information quality: part 1, an investigative framework," *Int. J. Inf. Qual.*, vol. 2, no. 3, p. 205, 2011.

[40] A. Marotta, "Calidad de Datos." Instituto de Computación, Facultad de Ingeniería de la UdelaR, Montevideo, 2009.

[41] L. Etcheverry, V. Peralta, and M. Bouzeghoub, "Qbox-Foundation: A Metamodel Plataform for Quality Measurement," in *DKQ 2008 in EGC 2008*, 2008, pp. 1–10.

[42] J. Du and L. Zhou, "Improving financial data quality using ontologies," *Decis. Support Syst.*, vol. 54, no. 1, pp. 76–86, Dec. 2012.

[43] M. Bobrowski, M. Marré, and D. Yankelevich, "A software engineering view of data quality," in *Intl. Software Quality Week Europe (QWE'98)*, 1998, pp. 1–10.

[44] M. Bobrowski, "Measuring data quality," Buenos Aires, 1999.

[45] L. Etcheverry, A. Marotta, and R. Ruggia, "Data Quality Metrics for Genome Wide Association Studies," in *2010 Workshops on Database and Expert Systems Applications*, 2010, pp. 105–109.

[46] I. Caballero, "Calidad y Medición de Sistemas de Información." Universidad de Castilla La Mancha, España, Castilla de la Mancha, 2009.

[47] V. Y. Yoon, P. Aiken, and T. Guimaraes, "Managing Organizational Data Resources," *Inf. Resour. Manag. J.*, vol. 13, no. 3, pp. 5–13, Jan. 2000.

[48] A. Caro, C. Calero, I. Caballero, and M. Piattini, "Defining a Data Quality Model for Web Portals," in *Web Information Systems–WISE 2006*, 2006, pp. 363–374.

[49] J. Gao, S. Lin, and A. Koronios, "Data Quality in Engineering Asset Management Organizations-Current Picture in Australia," in *International Conference on Information Quality (ICIQ 2006)*, 2006.

[50] D. A. Marotta, S. Fagundez, and J. Fleitas, "Calidad de datos en sensores," Universidad de la República, 2010.

[51] E. Martirena and V. Peralta, "Medición de la calidad de datos: Un enfoque parametrizable," Universidad de la República, 2008.

[52] H.-T. Moges, K. Dejaeger, W. Lemahieu, and B. Baesens, "A multidimensional analysis of data quality for credit risk management: New insights and challenges," *Inf. Manag.*, vol. 50, no. 1, pp. 43–58, Jan. 2013.

[53] H. Bui, D. Wright, C. Helm, R. Witty, P. Flynn, and D. Thain, "Towards long term data quality in a large scale biometrics experiment," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing - HPDC '10*, 2010, p. 565.

[54] E. Rahm and H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, 2000.

[55] A. Mockus, "Missing data in software engineering," *Guid. to Adv. Empir. Softw. Eng.*, pp. 185–200, 2008.

[56] V. R. Basili and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 6, pp. 728–738, Nov. 1984.

[57] B. Kitchenham, D. I. K. Sjøberg, O. P. Brereton, D. Budgen, T. Dybå, M. Höst, D. Pfahl, and P. Runeson, "Can we evaluate the quality of software engineering experiments?," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10*, 2010, p. 1.

[58] B. Kitchenham and D. Sjoberg, "Trends in the Quality of Human-Centric Software Engineering Experiments: A Quasi-Experiment," *IEEE Trans. Softw. Eng.*, vol. 39, no. 7, pp. 1002–1017, 2013.

[59] O. Dieste, A. Grim´n, N. Juristo, and H. Saxena, "Quantitative Determination of the Relationship between Internal Validity and Bias in Software Engineering Experiments: Consequences for Systematic Literature Reviews," in *2011 International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 285–294.

[60] G. a. Liebchen and M. Shepperd, "Software Productivity Analysis of a Large Data Set and Issues of Confidentiality and Data Quality," in *11th IEEE International Software Metrics Symposium (METRICS'05)*, 2005, no. Metrics, pp. 46–46.

[61] Y. Seo, K. Yoon, and D. Bae, "An empirical analysis of software effort estimation with outlier elimination," in *Proceedings of Predictor Models in Software Engineering (PROMISE 2008)*, 2008, pp. 25–32.

[62] D. Rodriguez, I. Herraiz, and R. Harrison, "On software engineering repositories and their open problems," in *2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE)*, 2012, pp. 52–56.

[63] K. Strike, K. El Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Trans. Softw. Eng.*, vol. 27, no. 10, pp. 890–908, 2001.

[64] I. Myrtveit, E. Stensrud, and U. H. Olsson, "Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods," *IEEE Trans. Softw. Eng.*, vol. 27, no. 11, pp. 999–1013, 2001.

[65] T. Khoshgoftaar and P. Rebours, "Improving software quality prediction by noise filtering techniques," *J. Comput. Sci. Techonology*, vol. 33, no. 3, pp. 387–396, 2007.

[66] A. Folleco, "Identifying learners robust to low quality data," in *IEEE International Conference on Information Reuse and Integration, 2008. IRI 2008.*, 2008, pp. 190–195.

[67] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, and A. Folleco, "An empirical study of the classification performance of learners on imbalanced and noisy software quality data," *Inf. Sci. (Ny).*, pp. 651–658, 2014.

[68] B. Twala, M. Cartwright, and M. Shepperd, "Ensemble of missing data techniques to improve software prediction accuracy," in *Proceeding of the 28th international conference on Software engineering - ICSE '06*, 2006, p. 909.

[69] A. Disney and P. Johnson, "Investigating Data Quality Problems in the PSP (Experience Paper)," in *SIGSOFT'98*, 1998, pp. 143–152.

[70] P. Johnson and A. Disney, "A critical analysis of PSP data quality: Results from a case study," in *Empirical Software Engineering*, 1999, vol. 349, pp. 317–349.

[71] P. Johnson and A. Disney, "The personal software process: A cautionary case study," *Software, IEEE*, no. December, pp. 85–88, 1998.

[72] A. Wesslén, "A Replicated Empirical Study of the Impact of the Methods in the PSP on Individual Engineers," in *Empirical Software Engineering*, 2000, vol. 123, pp. 93–123.

[73] S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in *Proceeding of the 33rd international conference on Software engineering - ICSE '11*, 2011, p. 481.

[74] A. Bachmann and A. Bernstein, "Software process data quality and characteristics: a historical view on open and closed source projects," in *IWPSE-Evol'09*, 2009, pp. 119–128.

[75] R. Wu, H. Zhang, S. Kim, and S. Cheung, "Relink: recovering links between bugs and changes," in *ESEC/FSE'11*, 2011, pp. 15–25.

[76] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann, "What makes a good bug report?," in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering - SIGSOFT '08/FSE-16*, 2008, p. 308.

[77] P. Schugerl, J. Rilling, and P. Charland, "Mining Bug Repositories--A Quality Assessment," in *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, 2008, pp. 1105–1110.

[78] C. Bird, A. Bachmann, and E. Aune, "Fair and balanced?: bias in bug-fix datasets," in *Proceedings of the ESEC-FSE'09*, 2009.

[79] M. Shepperd and Q. Song, "Data quality: some comments on the NASA software defect datasets," *IEEE Trans. Softw. Eng.*, pp. 1–14, 2013.

[80] D. Gray, D. Bowes, N. Davey, and B. Christianson, "The misuse of the NASA Metrics Data Program data sets for automated software defect prediction," in *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*, 2011, pp. 96–103.

[81] J. Aranda and G. Venolia, "The secret life of bugs: Going past the errors and omissions in software repositories," in *2009 IEEE 31st International Conference on Software Engineering*, 2009, pp. 298–308.

[82] M. H. Cartwright, M. J. Shepperd, and Q. Song, "Dealing with missing software project data," in *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (METRICS 2003)*, 2003, pp. 154–165.

[83] B. Kitchenham, "Procedures for performing systematic reviews," 2004.

[84] R. Sison, "Personal software process (psp) assistant," in *Asia-Pacific Software Engineering Conference, 2005 (APSEC'05)*, 2005, pp. 0–7.

[85] J. Lappalainen, "Tool Support for Personal Software Process," in *Product Focused Software Process Improvement (PROFES 2005)*, 2005, pp. 545–559.

[86] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, "Chapter 3: Measurement," in *Experimentation in Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–43.

[87] L. Briand, K. El Emam, and S. Morasca, "On the Application of Measurement Theory in Software Engineering," *Empir. Softw. Eng.*, vol. 1, no. 1, pp. 1–23, 1996.

[88] C. Kaner and W. Bond, "Software engineering metrics: What do they measure and how do we know?," in *METRICS 2004*, 2004, pp. 1–12.

[89] "PROS - Centro de Investigación en Métodos de Producción de Software (UPV)." [Online]. Available: http://www.pros.upv.es/.

[90] D. W. Embley, S. W. Liddle, and O. Pastor, "Conceptual-Model Programming: A Manifesto," in *Handbook of Conceptual Modeling*, Springer, 2011, pp. 3–16.

[91] O. Pastor and J. C. Molina, *Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[92] "INTEGRANOVA." [Online]. Available: http://www.integranova.com.

[93] D. Moody, "The method evaluation model: a theoretical model for validating information systems design methods," in *ECIS 2003 Proceedings*, 2003.

[94] D. Moody and G. Sindre, "Evaluating the Quality of Process Models: Empirical Analysis of a Quality Framework," in *Conceptual Modeling—ER 2003*, 2003.

[95] "GrISE - Grupo de Investigación en Ingeniería de Software Empírica (UPM)." [Online]. Available: http://www.grise.upm.es/.

[96] "GrIS - Grupo de Ingeniería de Software (UdelaR)." .

[97] N. Juristo and S. Vegas, "Functional testing, structural testing, and code reading: What fault type do they each detect?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2765, pp. 208–232, 2003.

[98] D. Vallespir, C. Apa, and S. De León, "Effectiveness of five verification techniques," in *Proceedings of the XXVIII International Conference of the Chilean Computer Society*, 2009.

[99] D. Vallespir and J. Herbert, "Effectiveness and Cost of Verification Techniques: Preliminary Conclusions on Five Techniques," in *2009 Mexican International Conference on Computer Science*, 2009, pp. 264–271.

[100] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*. Hightstown, NJ, USA: McGraw-Hill, Inc., 1996.

[101] B. Beizer, *Software Testing Techniques (2nd Ed.)*. New York, NY, USA: Van Nostrand Reinhold Co., 1990.

[102] D. Vallespir, C. Apa, and S. De Leon, "Material Módulo de Taller de Verificación." Montevideo, 2008.

[103] C. Valverde and B. Bianchi, "Un caso de estudio en Calidad de Datos para Ingeniería de Software Empírica (Tesis de Grado)," Universidad de la República, Uruguay, 2009.

[104] V. R. Basili and R. W. Selby, "Comparing the Effectiveness of Software Testing Strategies," *IEEE Trans. Softw. Eng.*, vol. 13, no. 12, pp. 1278–1296, 1987.

Agradecimientos

Cuando uno emprende la difícil y costosa aventura de llevar a cabo un trabajo de tesis, necesita necesariamente contar con el apoyo y soporte de muchos. Esos "muchos" son todos aquellos que de una forma u otra estuvieron presentes (o ausentes) cuando tenían que estarlo. Es gracias a esos "muchos" que hoy puedo sentirme orgullosa por el trabajo realizado. Y es a esos "muchos" a quienes quiero decir MUCHAS GRACIAS.

A mis tutores Adriana y Diego, que estuvieron siempre presentes aún en la distancia, apoyando y guiando mi trabajo, dando el soporte profesional, académico y también personal para llevar esta tesis a buen puerto. Gracias por confiar en mi para realizar este trabajo, y darme la autonomía necesaria y suficiente para lograrlo.

A mi hermana Cecilia, que tal vez sin saberlo siempre fue una guía fundamental en mi camino y ejemplo a seguir. Gracias Ceci por apoyarme en mis decisiones, gracias por tus consejos siempre adecuados, gracias por estar.

A mis padres Carmen y Carlos, que siempre me apoyaron y supieron entender mis decisiones. Gracias por siempre querer lo mejor para nosotras, y sobre todo por dejarnos decidir qué es lo mejor. Gracias mamá por soportar mi ansiedad durante las largas horas de tesis.

A mis amigas de siempre, por entenderme y aconsejarme, por preocuparse, por estar siempre presentes y por apoyarme cuando tuve que estar ausente.

Al Grupo de Ingeniería de Software de la Facultad de Ingeniería, al cálido grupo humano que me brindó constante apoyo, soporte y aliento para lograr este trabajo. En especial a mis compañeros de ruta durante la maestría.

A todas las personas y organizaciones que hicieron posibles que parte de esta tesis sea realizada conociendo otras Universidades y grupos humanos. Gracias por la enriquecedora experiencia, por permitirme compartir mi trabajo con otros grupos de investigación y así aportar en mi tesis. Gracias a la Dirección General de Relaciones y Cooperación, a la Universidad de Porto y al Proyecto BABEL, a la Oficina de Acción Internacional de la UPV.

Al BPS que me apoyó en la experiencia realizada en el exterior durante el trabajo de tesis. En especial a Estela y Roque, por entender la importancia de realizar este trabajo y facilitarme las herramientas para poder llevarlo a cabo.

A Natalia y Oscar, que me guiaron durante mi estancia en Madrid y Valencia, por recibirme con los brazos abiertos e integrarme en sus equipos de trabajo.

A mis amigos de Valencia, dentro y fuera de la UPV, que hicieron que mi estancia sea una experiencia sumamente valiosa, que me hizo crecer profesional, académica y personalmente. Gracias a todos los que hicieron más difícil mi regreso.

A mi sobrino Franco, que aunque aún no entiende nada de tesis de maestrías ni mucho menos, fue la principal motivación para llevar a cabo este trabajo en tiempo.

Anexo A: Aplicación de la Metodología y Modelo de Calidad sobre los Datos del Experimento de UdelaR

En este anexo se presenta cómo se aplicó la metodología de trabajo y el modelo de calidad propuesto sobre los datos del Experimento de Efectividad de Técnicas de Verificación de UdelaR.

A.1 Fase 1: Generar conocimiento del experimento

De forma de generar el conocimiento necesario del experimento bajo estudio, se llevaron a cabo reuniones de trabajo entre el analista de calidad de datos y el responsable del experimento. También se estudió el material enviado por el experimentador (informe de proyectos de grado, diagramas, guías) [98], [99], y se realizaron algunas consultas puntuales de forma personal.

Al relevar la información se obtiene conocimiento en el diseño del experimento sobre el cual se analizará la calidad de sus datos. A continuación se presenta la información recolectada de forma resumida.

A.1.1 Diseño experimental

El experimento fue realizado como un trabajo de proyecto de grado, por estudiantes de la carrera Ingeniería en Computación de la Facultad de Ingeniería – Universidad de la República. Fue desarrollado en un contexto académico, en el marco de una asignatura que se dicta en el 4° año de dicha carrera. Participaron un total de 14 estudiantes, utilizando las siguientes técnicas de verificación:

- Inspecciones (estática).
- Particiones en Clases de Equivalencia y Análisis de Valores Límites (dinámica, caja negra).
- Tablas de Decisión (dinámica, caja negra).
- Criterio de Cubrimiento de Condición Múltiple (dinámica, caja blanca).
- Trayectorias Linealmente Independientes (dinámica, caja blanca).

Los verificadores clasificaron los defectos detectados según dos taxonomías: ODC [100] y Beizer [101]. La taxonomía de ODC es una clasificación ortogonal de defectos. En el experimento solo interesó clasificar los defectos según las sub-categorías *DefectType* y *Qualifier*, ambas pertenecientes a la categoría *Closer Section*. Por otra parte, la taxonomía de Beizer es jerárquica. La clasificación de un defecto es un número de 4 dígitos. Las categorías más generales (como 3xxx para defectos de estructura) se dividen en sub-categorías (32xx para defectos de procesamiento, 322x para evaluación de expresión, y así) que representan una clasificación más específica del defecto. No fue requisito excluyente lograr la máxima especificidad al clasificar los defectos.

Los programas a verificar fueron construidos por estudiantes de 4to año de la carrera Ingeniería en Computación, en lenguaje Java, especialmente para uso del experimento.

El principal objetivo del experimento era conocer las relaciones de efectividad entre las técnicas de verificación utilizadas y los distintos tipos de defectos.

La unidad experimental era el software. Se utilizaron 4 programas de distinta naturaleza: contable con base de datos, cálculos matemáticos, generación de un documento consumiendo datos de una base de datos y procesador de texto. El hecho de contar con varios tipos de programa surge de la suposición de que un programa podría concentrar defectos de un determinado tipo y tener muy pocos de otros. Se intenta lograr una muestra más representativa de programas para tener la mayor cantidad de tipos de defectos posible.

Los verificadores hicieron uso de Guías de Trabajo [102], las cuales explican detalladamente la forma de realizar la ejecución del experimento y cómo llevar registro de los datos que se deben recolectar. Cada experimento unitario (o Experiencia de Verificación) consistió en un verificador aplicando una técnica de verifi-

cación a un programa. El diseño del experimento distribuye a los 14 participantes en 40 experiencias de verificación, cada una con un programa y una técnica de verificación.

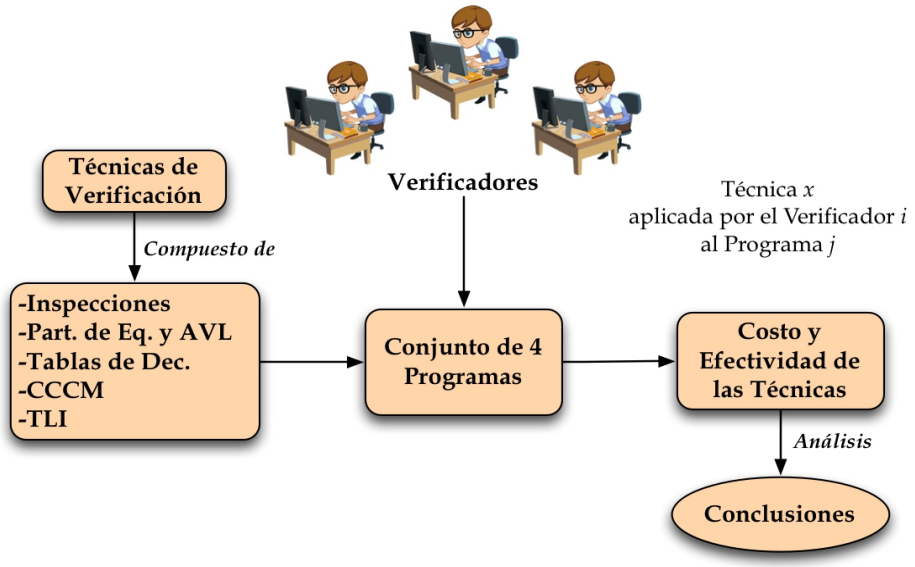

Ilustración 20: Diseño del experimento de Técnicas de Verificación, UdelaR

A.1.2 Datos recolectados y almacenados

Cada verificador aplicó distintas técnicas a distintos programas, y registró los siguientes datos acerca de cada experiencia:

- Fecha y hora de comienzo y finalización.
- Tiempo de diseño y ejecución de casos de prueba.
- Defectos encontrados.
- Para cada uno de los defectos detectados se registran los siguientes datos:
- Archivo en el cual se encuentra el defecto.
- Número de línea de código del defecto.
- Clasificación del defecto en ODC y Beizer.
- Estructura en la cual se encuentra el defecto (IF, FOR, WHILE, MÉTODO, NINGUNA)
- Número de línea en que comienza la estructura (si es que se ingresó estructura en el ítem anterior).
- Tiempo de detección del defecto.
- Descripción del defecto.

La Ilustración 20 muestra el diseño experimental presentado.

A.1.3 Herramienta para registro de defectos

Para la recolección de datos se utilizó una herramienta disponible vía web llamada *TVerificar*, construida a medida para la recolección de datos del experimento. La herramienta es una aplicación web y la arquitectura está basada en un modelo cliente-servidor. Como sistema gestor de Base de Datos se utiliza *HSQLDB*.

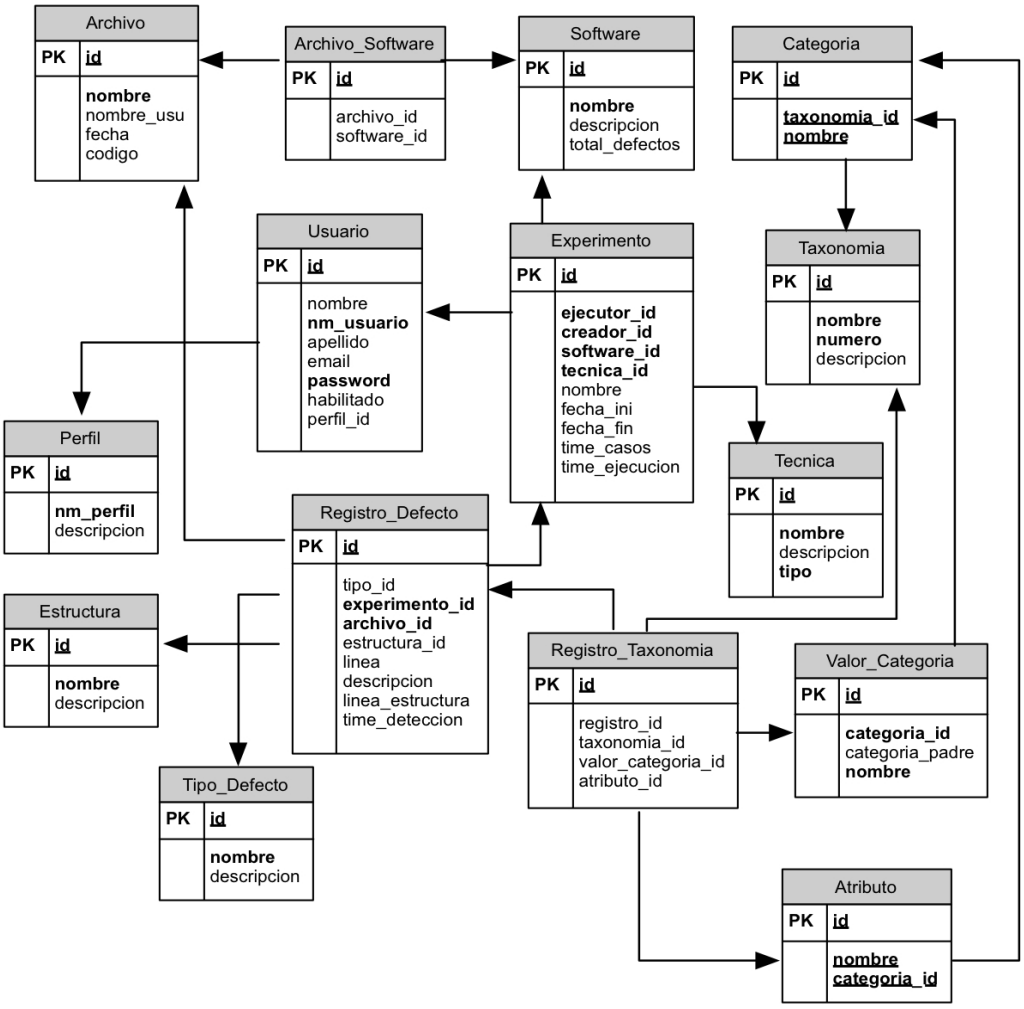

Ilustración 21: Esquema de base de datos de herramienta Grillo

En *TVerificar* se cargaron todos los experimentos unitarios identificados por: programa a verificar, técnica de verificación a aplicar y nombre del verificador. Cada verificador contaba con un usuario en la herramienta mediante el cual accedía a las experiencias que tenía asignadas, y registraba los datos requeridos.

La herramienta *TVerificar* (o Grillo) centraliza el registro de los defectos por parte de todas las experiencias de verificación en una misma base de datos, y permite tener un control y seguimiento de las mismas. Sus principales funcionalidades son gestionar datos de las entidades: Usuarios, Técnicas de Verificación, Experiencias y Defectos.

Se presenta en la Ilustración 21 el esquema de datos utilizado para almacenar la información de los experimentos, las pruebas ejecutadas por los verificadores, así como el registro de los defectos.

La Tabla 25 muestra una breve descripción de cada tabla, indicando si contienen valores precargados (en cuyo caso se especifican los valores), o en caso contrario qué perfil de usuario ingresa datos en las mismas.

A.2 Fase 2: Instanciar el modelo de calidad de datos

Durante esta fase se llevaron a cabo las tres estrategias propuestas por la metodología de trabajo. La Tabla 16 del Capítulo 8 muestra cuáles son las métricas de calidad que se aplican sobre los datos del experimento. Para cada métrica de calidad de datos considerada, se definieron los objetos sobre los cuales se aplicarán y las mediciones correspondientes.

De las 21 métricas de calidad que forman parte del Modelo de Calidad de Datos para Experimentos en Ingeniería de Software, 15 son aplicadas sobre los datos del experimento bajo estudio. Dichas métricas se aplican sobre 51 objetos particulares del dominio (ya sean celdas, tuplas o el conjunto de datos completo).

De forma de registrar los resultados de las mediciones se analizaron diferentes alternativas posibles. La alternativa seleccionada consistió en la creación de una nueva tabla por cada métrica de calidad aplicada, en la misma base de datos del experimento. Cada una de estas tablas contiene el resultado obtenido a partir de la aplicación de dicha métrica sobre todos los objetos definidos. El nombre de las nuevas tablas es igual al nombre de la métrica que se está registrando, más un prefijo *'reg_'*.

La estructura de cada tabla de registro depende de la métrica a registrar, y contiene la información relevante que corresponda así como las referencias (*foreign keys*) a las tablas de la base involucradas.

Se crea además un catálogo de métricas de calidad (*reg_ catalogo_ errores*) que contiene información específica sobre cada métrica aplicada. Todas las tablas de registro hacen referencia a este catálogo, indicando cuál es la métrica sobre la cual se registra la metadata.

Se describe a modo de ejemplo el registro para la métrica *Referencia inválida*. Se pueden encontrar más ejemplos en [103]. En este caso se crea una nueva tabla en la misma base de datos origen de nombre *reg_ referencia_ invalida*. La estructura de la tabla es la siguiente.

- *id*. Identificador autonumerado.
- *registro_ taxonomia_ id*. Identificador del registro taxonomía que contiene una referencia inválida (clave foránea a la tabla *Registro_ Taxonomia*).
- *valor_ categoria_ id*. Identificador del valor de la categoría que contiene una referencia inválida (clave foránea a la tabla *Valor_ Categoria*).
- *error_ id*. Indica cuál es la métrica para la cual se están registrando los resultados (clave foránea a la tabla *reg_ catalogo_ errores*).

En la Ilustración 22 se muestra el esquema de la base de datos origen luego de ejecutado el registro para tres métricas: *Valor Fuera de Rango, Referencia Inválida* y *Registro Duplicado*. Las nuevas tablas de registro creadas se visualizan en color anaranjado, mientras que las tablas de la base de datos origen involucradas en estos casos se ven en color verde.

Nombre Tabla	Descripción Tabla	Valores (para tablas pre-cargadas)	Perfil (para tablas pre-cargadas)
Archivo	Representa un archivo que contiene código fuente (.java).		Administrador
Software	Representa un software (conjunto de clases, métodos, etc.) que será sometido a verificación.		Administrador
Archivo_Software	Asocia un registro de la tabla Archivo con otro registro de la tabla Software. La relación entre las entidades Archivo y Software se representa con la entidad Archivo_Software ya que un Archivo puede estar asociado a 0..N Software, y a su vez un Software puede estar asociado a 1..N Archivo.		Administrador
Usuario	Representa un usuario que hará uso de la herramienta.		Administrador
Perfil	Representa el perfil que contiene el usuario que utiliza la herramienta.	Administrador, Verificador	
Experimento	Representa un experimento a ejecutar y su información asociada.		Los campos Fecha Fin, Tiempo de Ejecución y Tiempo de Casos no son obligatorios y pueden ser cargados tanto por un usuario Administrador como por un Verificador. El resto de los campos son obligatorios y solo pueden ser completados por un usuario Administrador.
Técnica	Representa una Técnica de Verificación. Cada registro de Experimento tiene una única Técnica asociada.	Inspecciones, CCCM, Trayectorias Independientes, Tablas de Decisión,Partición de Equivalencia y A. de Valores Límite	
Registro_Defecto	Representa un defecto hallado en el código por un usuario Verificador.		Verificador
Estructura	Representa los tipos de estructuras que pueden existir	IF, FOR, WHILE, SWITCH, DO	

	en el código fuente. A cada registro de la tabla Registro_Defecto se le asocia un único registro de la tabla Estructura.	WHILE, METODO, CLASE	
Tipo_Defecto	Representa si el defecto está contenido o no en una estructura. A cada registro de la tabla Registro_Defecto se le asocia un único registro de la tabla Tipo_Defecto.	Estructura, Sin estructura	
Taxonomia	Representa las taxonomías existentes.	ODC, Beizer	
Categoria	Representa los "niveles" de cada taxonomía.	Para ODC: *Section,* Actividad, *Trigger,* Impacto, *Defect Type, Qualifier, Age, Source* Para Beizer: BeizerNivel1, BeizerNivel2, BeizerNivel3, BeizerNivel4	
Atributo	Tabla vacía (sin uso)		
Registro_Taxonomia	Representa un la clasificación de un registro de defecto según las taxonomías de Beizer e ODC.		Verificador
Valor_Categoria	Representa las categorías de cada nivel de la taxonomía.	Ver base	

Tabla 25: Descripción de las tablas de la base de datos

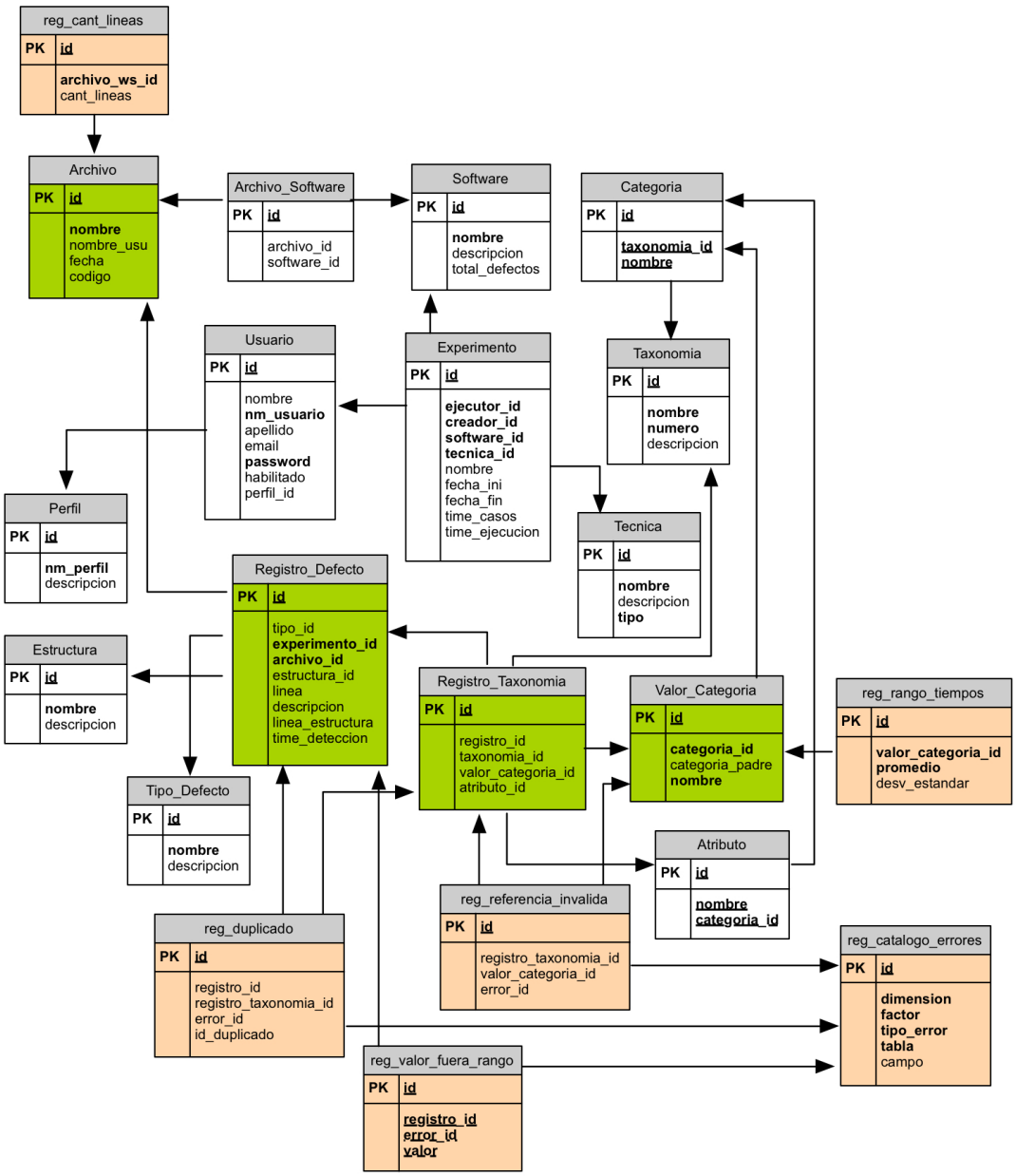

Ilustración 22: Esquema de base de datos de herramienta Grillo con metadatos de calidad

A.2.1 Definición de las métricas de calidad instanciadas

A continuación se describe cómo se aplica cada métrica de calidad a los datos del experimento bajo estudio.

Valor fuera de rango

Resulta de interés que los valores de los tiempos (de diseño y ejecución casos de prueba, y detección de defectos) se encuentren dentro de un rango determinado.

El rango definido tiene su mínimo en 0 y a priori no se podría definir un valor máximo con certeza. Por este motivo se determina estadísticamente un intervalo considerando el valor medio y la desviación estándar de los tiempos registrados. Se considerará libre de error a aquellos valores dentro del rango [0, media + 2*desviación estándar], mientras que los tiempos fuera de dicho intervalo se considerarán candidatos a contener errores y serán analizados de manera aislada.

Falta de estandarización

Interesa que todos los tiempos se encuentren registrados en la unidad de medida minutos, de forma tal de que puedan realizarse operaciones (tal como hallar promedios) entre estos. Se considera que dicha estandarización queda incluida en la métrica Valor fuera de rango, ya que se puede pensar en tiempos muy pequeños como expresados en horas, o tiempos muy grandes como expresados en segundos.

Interesa además que los valores de los tiempos y líneas sean enteros. El significado de un tiempo con fracciones es difícil de interpretar, y no interesa lograr una precisión mayor a minutos. Por otra parte, carece de sentido que un número de línea no sea un valor entero. A pesar de que existe un control en la herramienta que no permite el ingreso de valores con decimales, podrían ingresarse datos directamente en la base que ocasionen la presencia de este problema de calidad.

La estandarización de las fechas de inicio y fin del experimento no resultan de particular interés en este caso.

Registro inexistente

Para esta métrica se analizan dos casos.

- Registro de defectos. Son defectos que fueron registrados por los verificadores en la herramienta, pero que no corresponden a un defecto real en el código.
- Archivos. Son archivos de código que fueron registrados por los administradores en la herramienta, pero no se utilizan en el experimento.

Registro con errores

Interesa que la información de los defectos sea ingresada tal cual sucedió en la realidad. Cualquiera de los datos que ingresan los verificadores al registrar un defecto podrían ser incorrectos y no reflejar la realidad.

Valor fuera de referencial

Resulta de interés que los valores ingresados se encuentren dentro de los referenciales definidos como válidos. Esto sucede para los datos de *Técnica, Software, Tipo de Defecto, Estructura, Taxonomía y Categoría.*

En caso de identificar la existencia de algún valor fuera del referencial definido, es necesario comprobar si hay registros que utilizan o referencian a alguno de estos valores. Estos también corresponderán a errores en los datos y por lo tanto deben ser tratados.

Valor nulo

Se identifican cuáles son los campos que admiten nulos según el esquema de la base de datos, pero deberían contener algún valor distinto de vacío. Estos son los casos que interesa medir. Se asume que el control sobre los campos declarados como no nulos se realiza correctamente por el SGBD.

Los campos sobre los cuales se verificará la existencia de valores nulos son los siguientes.

- Sobre la herramienta: perfil del usuario.
- Sobre el experimento: nombre del experimento, tiempo de ejecución y diseño de casos de prueba (este último solo para técnicas de verificación dinámica).
- Sobre los defectos: tipo, línea de código y tiempo de detección del defecto.

- Sobre la clasificación de defectos: taxonomía asociada al defecto, identificador y categoría de la taxonomía.

Información omitida

Interesa que todos los defectos detectados estén clasificados según las taxonomías de Beizer y ODC. Para la taxonomía Beizer, el defecto debe ser clasificado en al menos el Nivel 1 de dicha taxonomía. No es necesario que se logre la máxima precisión dentro de esta taxonomía, pero sí que se cuente con una clasificación para cada defecto. Para la taxonomía de ODC, el defecto debe ser clasificado según las categorías *Defect Type* y *Qualifier*. Pueden clasificarse además según otras categorías dentro de esta misma taxonomía, pero estos datos no serán considerados para los análisis.

Regla de integridad de dominio

Para el caso de las líneas de código, el dominio de valores tendrá su mínimo en 1 y un máximo igual a la cantidad de líneas que posee el archivo. Es necesario considerar que los defectos que no se encuentran asociados a ninguna línea en particular tienen en este campo los valores 0, 999, 9999, y así sucesivamente. Por lo tanto, estos valores deberán tener un tratamiento aparte ya que se situarán fuera del dominio válido pero no corresponden a datos erróneos. Un valor en las líneas de código fuera del dominio especificado no permitiría identificar, por ejemplo, cuáles son los registros de defectos que corresponden al mismo defecto de la realidad.

Por otra parte, se debe cumplir que los tiempos de ejecución deben ser mayores a 0, tanto para técnicas de verificación estáticas como dinámicas.

Regla de integridad intra-relación

Se definen las siguientes reglas de consistencia que deben cumplirse sobre los datos registrados para los experimentos y los defectos detectados.

- En las técnicas de verificación estáticas no se requiere diseñar casos de prueba, por lo que el tiempo de diseño debe ser 0 o nulo.
- En las técnicas de verificación dinámicas siempre se requiere diseñar casos de prueba, por lo que el tiempo de diseño debe ser mayor a 0.
- En las técnicas de verificación estáticas el tiempo de detección del defecto debe ser igual a 0. Esto se debe a que durante estas técnicas se trabaja sobre el código, detectando directamente los defectos.
- Un defecto puede ser del tipo "Estructura" o "Sin estructura", dependiendo de si el mismo se detectó o no en una estructura (IF, FOR, WHILE, etc) dentro del código. Si el defecto se identifica dentro de una estructura, entonces se debe ingresar la línea de código en la cual se encuentra la estructura.

Valor único

Se identifican cuáles son los campos para los cuales no fue definida la restricción de unicidad, pero que deberían contener un valor único. Estos son los casos que interesa medir. Se asume que el control sobre los campos declarados como únicos se realiza correctamente por el SGBD.

Los campos sobre los cuales se verificará la existencia de valores que no son únicos son los siguientes.

- Nombre del experimento.
- Nombre de la categoría.
- Nombre del valor de la categoría y su padre. Estos atributos se consideran en conjunto ya que pueden existir nombres de categorías iguales para distintos niveles de la taxonomía.

Referencia inválida

Se analizan tres casos para los cuales la definición de foreign key fue omitida: Taxonomía inválida, Registro de defecto inválido y Categoría padre inválido. Estas inconsistencias ocasionan la existencia de referencias hacia registros que no existen en la base del experimento.

Registro duplicado

A pesar de que los controles del SGBD evitan la existencia de registros duplicados con la misma clave primaria, se realizan los chequeos necesarios para verificar que no existan registros repetidos según el criterio de duplicación definido en cada caso.

Se consideran tres casos.

- Registros de defectos. Se considera que dos o más registros de defectos se encuentran duplicados si fueron identificados en la misma línea de código del mismo archivo fuente, y durante el mismo experimento (o sea por el mismo verificador). Debido a que podrían existir dos defectos distintos que cumplan con estas características (si existe más de un defecto en la misma línea del mismo archivo), los registros resultantes deben ser analizados para corroborar la duplicación.
- Registros de Taxonomia para ODC y para Beizer (clasificación de defectos). Se considera que dos o más registros de taxonomía se encuentran duplicados si se asocian a un mismo registro de defecto y una misma categoría.

Registro contradictorio

Se consideran tres casos.

- Archivo de Software. Existen contradicciones cuando dos o más registros de archivos de software hacen referencia al mismo archivo pero con distinto software. El mismo archivo no puede pertenecer a más de un software distinto.
- Registro de Taxonomia para ODC y para Beizer. Se considera que dos o más registros de taxonomía son contradictorios si para una misma categoría contienen valores distintos. Un defecto debe estar clasificado de una única manera según una misma taxonomía.
  En particular para la taxonomía de Beizer, se considera además el caso en el que la clasificación de un defecto se encuentra "cortada". Esto significa que las categorías en las cuales fue clasificado un defecto para los distintos niveles de Beizer, no se corresponden con la jerarquía planteada (a modo de ejemplo, un defecto se encuentra clasificado en 1xxx, 21xx y 311x).

Estructura de datos

Se identifican cuáles son las restricciones que existen sobre los datos (tales como *unique, not null, foreign key*), y las reglas que deberían cumplirse sobre estos. Luego se evalúa si están definidas, si son controladas por el SGBD, y si se encuentran documentadas.

Por otra parte, se analiza si la estructura de base de datos que se define es adecuada para almacenar los datos de la realidad bajo estudio.

Metadata

Se identifica si se define y documenta el esquema de base de datos, la metadata e información de trazabilidad correspondiente, y si están de acuerdo a la realidad que representan. También es importante que esté documentada la información necesaria sobre la herramienta en la cual se registran los datos, y pautas o guía sobre el ingreso de los mismos.

A.3 Fase 3: Evaluar la calidad de los datos

En la Tabla 16 del Capítulo 8 se muestra el resultado obtenido luego de la aplicación de cada métrica de calidad sobre los datos del experimento. A partir de la medida obtenida (valor de calidad), es posible identificar cuáles son los problemas de calidad que están presentes en los datos.

Se ejecutan 51 mediciones sobre cada uno de los objetos del dominio definidos en la fase anterior, de las cuales:

- El 74% (38 mediciones) se realizan de forma automática, mediante la ejecución de consultas SQL y algoritmos programados.
- El 18% (9 mediciones) se realizan de forma manual por no ser posible o necesaria su automatización. En todos los casos corresponden a revisiones o verificaciones manuales sobre los datos, mediante la comparación con otra fuente de datos o realizando consultas al experimentador. Como parte de estas mediciones, se incluyen 2 referentes a la Representación e Interpretabilidad.
- Las mediciones del restante 8% (4 mediciones) son aquellas que no es posible su ejecución.
  Para el caso de Valor fuera rango (mediciones A1.1 y A1.2) se consideran las distintas combinaciones de técnica-software, obteniendo en ambos casos que existen como máximo 3 experimentos por cada combinación posible. Esta información no es suficiente para calcular promedios y desviaciones estándar que resulten significativos para el análisis, por lo tanto no es posible realizar estas mediciones.
  La medición A3.1 de Registro inexistente no fue ejecutada por el esfuerzo que requiere. Sería necesario chequear manualmente los 1009 registros de defectos, y evaluar si corresponden o no a un defecto de la realidad. Para esto se debe recurrir al código fuente e identificar a partir de la información existente si efectivamente existe o no un defecto.
  En el caso de Registro con errores (medición A4.1) sería necesario hacer una revisión manual de los datos ingresados para cada defecto, y evaluar si corresponden o no a valores válidos. Otra forma posible de medición manual sería recurrir a los sujetos del experimento, consultando por la correctitud de los datos ingresados. Debido a su alto costo, esta medición no fue realizada.

Como resultado se encuentra que la medida obtenida es menor a 1,00 o 'Regular' para 20 de las 51 mediciones ejecutadas, indicando la presencia de un problema de calidad en los datos. Estas mediciones corresponden a 11 métricas de calidad diferentes. Se analizan los resultados obtenidos y se identifican los datos que contienen algún problema de calidad, clasificándose como sigue.

A.3.1 Errores en los datos

Como resultado de la ejecución de 19 de las 20 mediciones se identifican datos erróneos. Se detalla cada caso a continuación.

Registro inexistente: en archivos de software (medición A4.2)

Se verifica manualmente que el archivo DataPregunta.java del software Matemático no se encuentra entre los fuentes utilizados en el experimento. La causa de este problema es una equivocación del administrador al registrar un archivo que no se utiliza en el experimento.

Valor fuera de referencial: en el nombre de la categoría (mediciones A6.6 y A6.7)

Se verifica mediante consultas con el responsable del experimento que existen 6 valores de Categoría que no se encuentran dentro del referencial definido como válido, estos son: *Section, Actividad, Trigger, Impacto, Age y Source*. El Administrador de la herramienta es quien ingresa estos valores, ocasionando la existencia de datos que se encuentran fuera del referencial considerado como válido.

A su vez, se buscan los registros de defectos que se clasificaron según alguna de las 6 categorías que se encuentran fuera del referencial. Como resultado se obtienen 3721 registros de taxonomía.

La Tabla 26 muestra la cantidad de registros con valores fuera de referencial por cada categoría. Existen 685 registros que por presentar el problema de calidad Referencia inválida (medición A15.2) no son considerados en la siguiente tabla.

Categoría	Cantidad de registros con valores fuera de referencial
Section	1009
Age	1006
Source	1006
Actividad	5
Trigger	5
Impacto	5

Tabla 26: Cantidad de registros con valores fuera de referencial por categoría

Valor nulo: en el tiempo de ejecución de casos de prueba, identificador de taxonomía, valor de categoría y tiempo de diseño de casos de prueba (mediciones A8.7, A8.8, A8.9, A8.10)

Hay 3 registros con valores nulos en el tiempo de ejecución de casos de prueba, y 3 con valores nulos en el tiempo de diseño de casos de prueba (para técnicas dinámicas). Seguramente esto se deba a una omisión por parte del Verificador al ingresar o calcular este tiempo.

Se identifican por otra parte 1324 registros que contienen valores nulos en el identificador de la taxonomía, y 2 registros que contienen valores nulos en la categoría. Puede haber ocurrido algún error al almacenar la información correspondiente a la clasificación de un defecto que causa la existencia de estos valores nulos.

Información omitida: en la clasificación para ODC (medición A9.1)

Se obtienen 3 registros que no fueron clasificados según las categorías Defect Type y Qualifier para la taxonomía ODC. Esto puede deberse a una omisión o falta de conocimiento por parte del Verificador, o un error de la herramienta al almacenar dicha información.

Regla de integridad de dominio: en líneas de código y estructura, y tiempo de ejecución (mediciones A11.1, A11.2, A11.2)

Se identifican 8 registros que presentan un valor en la línea de código que no pertenece al dominio definido, y 3 para el caso de la línea de estructura. Estos 3 registros coinciden con alguno de los 8 detectados en la medición A11.1, además de que el valor en la línea y la línea de la estructura también coinciden.

En el caso de las líneas de código el error puede deberse a que se haya solicitado un pedido de corrección del código fuente, por encontrar un defecto bloqueante o que cubra otros defectos. A pesar de que se requiere registrar la línea del archivo original, es posible que se haya registrado la línea del archivo corregido.

Finalmente, se detectan 2 casos en los cuales el valor del tiempo de ejecución es 0.

Regla de integridad intra-relación: en el tiempo de detección (medición A12.3)

Hay 90 casos en los cuales el tiempo de detección de defectos para la técnica de verificación estática no es 0. Podría ocurrir que el verificador considere otras actividades como parte del tiempo de detección de un defecto (por ejemplo, el tiempo en clasificarlo), lo cual ocasiona que sea mayor a 0.

Referencia inválida: en clasificación de defectos y categorías (mediciones A15.2 y A15.3)

Hay 472 registros con referencias inválidas en la clasificación de defectos, que corresponden a 33 registros de defectos (cada defecto se encuentra asociado a varios registros de taxonomía).

Por otra parte, hay 19 registros que contienen referencias inválidas a categorías.

La fuente de este error se debe a un error en el diseño del esquema de la base de datos, ya que se omite la definición de foreign keys sobre ciertos atributos.

Registro duplicado: en clasificación de defectos para ODC (medición A16.2)

Hay 1534 registros de taxonomía que se encuentran duplicados, que corresponden a 171 registros de defectos. Existen 773 casos ya considerados en el resultado de la medición A6.6, y 204 en la A15.2.

Como se observa en la Tabla 27, la cantidad de registros taxonomía duplicados para un mismo defecto y una misma categoría, puede variar desde un mínimo de 2 hasta un máximo de 44 (se muestran algunos registros de defectos a modo de ejemplo). Seguramente este problema se deba a un error de la herramienta que ocasione se almacenen registros repetidos en la base.

Identificador *Registro_Defecto*	**Cantidad de registros duplicados**
29	2
48	6
79	8
473	16
165	20
40	25
49	29
901	44

Tabla 27: Cantidad de registros duplicados por tupla

Registro contradictorio: en software de archivos, clasificación de defectos para ODC y Beizer (mediciones A17.1, A17.2 y A17.3)

Se obtienen 2 registros de archivos contradictorios (uno de ellos no existe en la realidad al ser también resultante de la medición A4.2). La causa de este problema se debe a un error en el diseño de la base, al establecer el atributo Archivo.nombre como único. Esta restricción ocasiona la existencia de archivos que contienen el mismo nombre a pesar de que no corresponden al mismo archivo de la realidad.

A partir de la medición A17.2 se obtienen 122 registros de taxonomía contradictorios para ODC, que corresponden a 23 registros de defectos. Existen 26 ya considerados en la medición A6.6, y 100 incluidos en la A15.2. Como se observa en la Tabla 28, la cantidad de registros taxonomía contradictorios varía desde 2 hasta un máximo de 31.

Identificador Registro_Defecto	**Cantidad de registros contradictorios asociados**
132	2
29	4
37	8
0	24

40	27
49	31

Tabla 28: Cantidad de registros contradictorios por tupla (ODC)

Finalmente, a partir de la medición A17.3 se obtienen 43 registros de taxonomía contradictorios para Beizer, que corresponden a 22 defectos distintos. La cantidad de registros taxonomía contradictorios en este caso varía desde un mínimo de 2 hasta un máximo de 6.

Seguramente estos problemas de calidad se deba a un error de la herramienta que ocasione se almacenen registros contradictorios en la base.

Estructura de datos (medición A18.1)

Los datos se registran en una base de datos relacional, utilizando un manejador de base de datos (HSQLBD). Se definen algunas restricciones sobre el repositorio de datos.

A partir de la evaluación de la calidad de los datos, se identifican errores en el diseño de la base de datos que se traducen en problemas de calidad sobre los datos. Los errores identificados son:

- En la realidad planteada, cada defecto hallado en el código es categorizado según dos taxonomías: ODC y Beizer. En el diseño actual de la base, existe una tabla Registro_Defecto que representa un defecto en el código, y otra tabla *Registro_ Taxonomía* que representa una clasificación dentro de alguna taxonomía. La relación entre *Registro_ Taxonomía* y *Registro_ Defecto* es N a 1. Esta relación permite que un defecto esté clasificado más de una vez en una misma taxonomía, o que solamente esté clasificado en una de ellas.
  El diseño debería asegurar que una tupla de la tabla *Registro_ Defecto* esté asociada a exactamente tres tuplas de la tabla *Registro_ Taxonomía*: uno para Beizer, uno para ODC *Qualifier* y otro para ODC *Defect Type*.
- La representación de las diferentes categorías dentro de cada taxonomía se realiza mediante las tablas *Categoria* y *ValorCategoria*. Para la clasificación de un defecto dentro de la taxonomía jerárquica de Beizer, interesa conocer solamente la categoría más específica a la que se logra llegar. Según el esquema actual, sin embargo, puede darse el caso de que un defecto en el código esté asociado a más de una categoría dentro de una misma taxonomía.
- Según el esquema actual de la base de datos, es posible que un mismo archivo esté asociado a más de un software. Esto ocasiona la existencia de contradicciones en los datos, debido a que en la realidad planteada, dos software distintos no pueden tener archivos en común.

Estos errores en el diseño causas los problemas de calidad de Registro Duplicado y Registro Contradictorio.

- Se omite la definición de la restricción not null y unique sobre ciertos atributos, así como la definición de determinadas foreign keys.

Los problemas asociados a estos errores de diseño se describen en las métricas Valor nulo, Valor único y Referencia inválida respectivamente.

A.3.2 Valores sospechosos

Se detecta la presencia de 1 valor sospechoso para el que no es posible asegurar la existencia de un error en los datos.

Valor fuera de rango: en el tiempo de detección de defectos (medición A1.3)

Se define un rango por cada tipo de defecto (según ODC *Defect Type*) para identificar los tiempos de detección que se encuentran fuera de dicho rango. Se calcula el promedio y desviación estándar, sin considerar los defectos detectados con la técnica “Inspecciones” (su tiempo de detección es 0).

Como resultado de la medición se obtienen 21 registros que contienen valores fuera del rango definido. Sin embargo, debido a que la definición de este rango es arbitrario, no se puede asegurar la existencia de errores en los datos hasta compararlos con valores reales.

En la Tabla 29 se muestran los registros de defectos cuyos tiempos de detección se alejan más notoriamente del rango definido.

Id *Registro_Defecto*	**Tiempo de detección**	**ODC** *Defect Type*	**Máximo**
360	35	Function/Class/Object	17
132	60	Assign/Initialization	25
304	64	Assign/Initialization	25
348	312	Checking	56
317	376	Checking	56

Tabla 29: Tuplas con los valores más lejanos al rango definido

A.3.3 Oportunidades de mejora

No se identifican oportunidades de mejora asociados a los problemas de calidad encontrados.

A.4 Fase 4: Ejecutar acciones correctivas sobre los datos

Se analiza en conjunto con el responsable del experimento si para los problemas de calidad identificados es posible aplicar acciones correctivas.

En este caso se ejecutan limpiezas para la mayoría de los errores identificados sobre los datos. Para ello, se planifican las siguientes etapas que deben ejecutarse secuencialmente.

A.4.1 Etapa 0: Corrección de errores en el diseño de la base de datos

En esta etapa se analiza el problema de calidad Estructura de datos (medición A18.1). Esto debe hacerse previo al análisis de cualquier otro problema, ya que involucra la estructura de la base en donde se almacenarán los datos.

Debido a la identificación de errores a nivel del diseño de la base de datos se define un nuevo esquema que los corrige. Por este motivo, resulta necesario llevar a cabo la migración de los datos desde la base de datos origen del experimento hacia el nuevo esquema definido. No se describen aquí las transformaciones ni correcciones realizadas a nivel del esquema o restricciones de la base de datos, las mismas se encuentran detalladas en [103].

En las siguientes etapas se detallan las correcciones que fueron llevadas a cabo de forma de limpiar los problemas de calidad detectados. La mayoría de los errores en los datos se corrigen durante la migración de las distintas tablas. Para ello se comienza desde las tablas que no contienen referencias a ninguna otra, siguiendo con aquellas que referencian a tuplas ya cargadas, y así sucesivamente.

Tanto la migración de los datos como la limpieza de la mayoría de los problemas de calidad se realizan de manera automática. Se construye una aplicación que ejecuta estas actividades. La información técnica, manual de usuario así como guías para adaptar la herramienta a otros experimentos se encuentran en [103]. A partir de un esquema de base de datos origen y uno destino, la aplicación realiza la medición de calidad, registro de metadatos, limpieza y migración de datos correspondiente. Para las limpiezas semi-automáticas, se brinda al usuario información sobre el objeto particular que contiene un problema de calidad, y se consulta sobre la acción a realizar. Opcionalmente se puede se-

leccionar un script que indica las correcciones manuales a realizar sobre la base de datos destino (luego de ejecutada la migración). Como salida, produce un archivo de *log* que registra las operaciones realizadas durante la ejecución del programa.

A.4.2 Etapa 1: Corrección de errores automática y semi-automática

Esta etapa incluye la limpieza de los problemas de calidad que pueden ser corregidos de forma automática o semi-automática.

Valor fuera de referencial en el nombre de la categoría (mediciones A6.6 y 6.7)

Se deben eliminar los 6 registros de Categoría que se encuentran fuera del referencial definido, así como los 3721 registros de taxonomía que referencian a alguna de dichas categorías.

Para esto, sólo se migran las categorías y clasificaciones de defectos que correspondan a *Defect Type* o *Qualifier* para la taxonomía de ODC, descartando los demás datos.

Valor nulo en el identificador de taxonomía (medición A8.8)

No resulta necesario aplicar limpiezas ya que los 1324 registros de taxonomía que contienen este problema de calidad son eliminados por la limpieza del problema Valor fuera de referencial.

Regla de integridad intra-relación sobre el tiempo de detección para técnicas estáticas (medición A12.3)

Se consulta al usuario del programa de limpieza y migración si para cada uno de los 90 registros que presentan este problema de calidad, la acción a realizar respecto al tiempo de detección es: ingresar el valor 0, ó migrar el valor almacenado en la base origen.

Referencia inválida en el identificador de defecto (mediciones A15.2 y A15.3)

Para limpiar este problema de calidad se deben eliminar de la base los 472 registros de taxonomía que contienen una referencia inválida en el identificador de defecto, y los 19 registros de valor de categoría que contienen una referencia inválida en la categoría padre.

Para esto, las tuplas que contienen una referencia inválida no son migradas a la base de datos destino.

Registro duplicado para la taxonomía ODC (medición A16.2)

Se deben eliminar los registros de taxonomía que se encuentran duplicados. Para esto, las 1534 categorizaciones que están duplicadas (mismo registro defecto y misma categoría) no son migradas a la nueva base destino, se migra una única categorización por cada defecto.

Registro contradictorio de archivo de software (medición A17.1)

Se agregan 2 nuevas tuplas de archivo, una por cada archivo de software contradictorio. Las nuevas tuplas contendrán los mismos datos que el archivo contradictorio correspondiente, pero con un distinto identificador.

Registro contradictorio para las taxonomías ODC y Beizer (mediciones A17.2 y A17.3)

En estos casos se debe definir un único registro de taxonomía (de entre los contradictorios identificados) para cada registro de defecto, los demás registros son descartados (un total de 122 para ODC y 43 para Beizer).

Para esto, se realiza la consulta al usuario del programa de limpieza y migración, quien debe seleccionar una única clasificación para cada defecto involucrado (23 para el caso de ODC y 22 para el caso de Beizer) de entre el conjunto de clasificaciones contradictorias posibles.

A.4.3 Etapa 2: Corrección de errores manual

En esta etapa se incluyen todas las limpiezas que se llevan a cabo de manera manual, debido a que corresponden a casos particulares que es necesario corregir pero no es posible automatizar. Estas limpiezas se realizan luego de finalizada la ejecución del programa de limpieza y migración, mediante la ejecución de un script directamente sobre la base de datos destino.

Valor fuera de rango en el tiempo de detección de un defecto (medición A1.3)

Debido a que este caso corresponde a un valor sospechoso, el primer paso para la limpieza consiste en conocer si realmente existe un error en los datos. Para ello se realizan consultas a los sujetos del experimento (mediante el envío de e-mails), para verificar si creen o recuerdan que existe un error en estos datos. Por otra parte, se comparan los valores que se encuentran fuera de rango con las planillas utilizadas por los sujetos, donde se registraron los datos del experimento y los defectos detectados (en paralelo al registro en la herramienta Grillo).

En la Tabla 30 se muestra el resultado obtenido para los 21 registros que contienen este problema de calidad. Como se observa, hay 4 registros que pueden ser corregidos ya que se detecta mediante la comparación de los valores en la base y en la planilla la existencia de un error en los datos. En 3 de los 4 casos se debe a que se ingresa el valor de la línea en el campo correspondiente al tiempo de detección (también se actualiza en la base el valor de la línea de código ya que se encontraba intercambiado con el valor del tiempo). En el caso restante, existe una diferencia de un dígito entre el valor registrado y la planilla (probablemente se deba a un error de tipeo). Para todos estos casos se actualiza el valor del tiempo de detección con el de la planilla. Vale destacar que 3 de los 4 registros limpiados corresponden a los valores que más se alejan del rango definido (ver Tabla 29).

En los restantes 17 casos, no es posible aplicar correcciones por no poder asegurar la existencia de un error en los datos, o por no conocer el valor real.

Cantidad de registros	Respuesta sujeto	Planillas de sujetos	Acción
4	Tiempo correcto	Igual valor en tiempo que en base	No limpiar
5	Tiempo correcto	No existe	No limpiar
3	No establece	Igual valor en tiempo que en base	No limpiar
2	No recuerda	No existe	No limpiar
3	Tiempo incorrecto	No existe	No limpiar
4	No establece	Diferente valor que en base	Limpieza manual

Tabla 30: Análisis de casos con valores de tiempos fuera de rango

Registro inexistente de archivos (medición A4.2)

La limpieza consiste en eliminar el registro que no corresponde a ningún archivo de la realidad. Se verifica que no existen registros de defectos que referencien a dicho archivo.

Esta limpieza se realiza de forma manual al igual que su detección.

Valor nulo en el tiempo de ejecución de casos de prueba (medición A8.7)

Se consulta con los responsables del experimento, y se verifica que 2 de los 3 casos corresponden a experimentos inválidos (ya sea porque no culminaron o porque fueron reemplazados por otro). Para el caso restante se detecta que existe un valor faltante, el cual se obtiene a partir de la planilla de registro del sujeto correspondiente.

Valor nulo en la categoría de la taxonomía (medición A8.9)

No resulta necesario aplicar limpiezas ya que los 2 registros de taxonomía que contienen este problema de calidad son eliminados por la limpieza del problema Referencia inválida.

Valor nulo en el tiempo de diseño de casos de prueba para técnicas dinámicas (medición A8.10)

Aplican las mismas correcciones que para la medición A8.7.

Información omitida en la clasificación de defectos para ODC (medición A9.1)

Esta información no fue ingresada en la planilla de registro de sujetos. Tampoco se realizan consultas a los sujetos ya que se considera un dato demasiado específico como para recordarlo. Por lo tanto, no es posible realizar corrección en ninguno de los 3 casos por no poder conocer cuál es el valor real.

Regla de integridad de dominio en líneas de código y estructura (mediciones A11.1 y A11.2)

Además de las consultas realizadas a los sujetos, se compara la cantidad de líneas de los archivos fuentes originales con aquellos modificados luego de algún pedido de corrección (por contener algún defecto bloqueante). También se comparan los nombres de archivos que contienen líneas fuera de rango con aquellos registrados en las planillas. De esta forma es posible identificar si existe un error en el valor de la línea debido a que se refiere a un archivo diferente al considerado en la medición.

Como muestra la Tabla 31, se realiza corrección en los 4 casos en los que el nombre del archivo es distinto al ingresado en la herramienta, y la cantidad de líneas del nuevo archivo hace que el valor de la línea se encuentre dentro del rango definido.

En los otros 4 casos, se mantiene el mismo valor de la base origen (no se realiza corrección), ya que no es posible conocer cuál es el valor real.

Cantidad de registros	Respuesta sujeto – pedido corrección archivo	Planillas de sujetos	Acción
4	No recuerda	Igual valor en línea y archivo que en base	No limpiar
4	No establece	Diferente valor en nombre del archivo que en base, igual valor en línea	Limpieza manual

Tabla 31: Análisis de casos con valores de línea fuera del dominio

A partir de la corrección aplicada para la medición A11.1 también se corrige el valor de la línea de estructura en 2 de los 3 registros resultantes de la medición A11.2, mientras que en el restante caso no es posible hacer corrección.

Regla de integridad de dominio sobre el tiempo de ejecución de casos (medición A11.3)

En 1 de los 2 casos es posible obtener el valor del tiempo de ejecución a partir del registro horario del sujeto, y realizar la corrección correspondiente. En el otro caso, el sujeto establece haber olvidado verificar la clase *Main.java*. Por lo tanto, el valor 0 en este campo resultaría correcto, y no corresponde aplicar corrección.

Registro contradictorio de archivo de software (medición A17.1)

En este caso es necesario actualizar la referencia (*foreign key*) de los registros de defectos hacia las 2 nuevas tuplas de archivos registradas, que fueron detectados como contradictorios.

Por otra parte, se proponen actividades de prevención generales que consideran a todos los problemas de calidad que fueron identificados, y que podrían ser incorporadas para futuras replicaciones del experimento de forma de contribuir en la mejora de su calidad. Estas se encuentran detallas en [103].

La Tabla 32 muestra el valor de calidad obtenido para las tablas sobre las cuales se identifican registros con algún problema de calidad. Vale destacar que un registro erróneo puede contener más de un problema diferente. Se aprecia cómo mejora el valor de calidad de todas las tablas luego de aplicar las limpiezas definidas.

Tabla	**Valor de calidad por tabla (antes limpieza)**	**Valor de calidad por tabla (después limpieza)**
Archivo	0,98	1,00
Archivo_Software	0,92	1,00
Experimento	0,89	0,98
Valor_Categoria	0,90	1,00
Registro_Defecto	0,88	0,98
Categoria	0,50	1,00
Registro_Taxonomia	0,48	1,00

Tabla 32: Valor de calidad por tabla antes y después de la limpieza

Anexo B: Aplicación de la Metodología y Modelo de Calidad sobre los Datos del Experimento de UPM

En este anexo se presenta cómo se aplicó la metodología de trabajo y el modelo de calidad propuesto sobre los datos del Experimento de Efectividad de Técnicas de Verificación y Relación con Tipos de Faltas de UPM.

B.1 Fase 1: Generar conocimiento del experimento

De forma de generar el conocimiento necesario del experimento bajo estudio, se llevó a cabo una reunión de trabajo entre el analista de calidad de datos y el responsable del experimento. También se tiene en cuenta el material enviado por el experimentador [97], y se realizan algunas consultas puntuales vía mail.

Al relevar la información se obtiene conocimiento en el diseño del experimento sobre el cual se analizará la calidad de sus datos. A continuación se presenta la información recolectada de forma resumida.

B.1.1 Diseño experimental

El objetivo del experimento consistió en comparar la efectividad relativa de técnicas de pruebas dinámicas (funcionales y estructurales) y estáticas (revisión de código), y relacionarlas con los tipos de faltas que detectan. La hipótesis general del estudio establecía que una técnica *ti* es más efectiva para detectar faltas de tipo *fj*, donde *ix*j corresponden a las diferentes sub-hipótesis.

Para este propósito, el estudio estaba compuesto de dos experimentos. El análisis de calidad se realizó sobre los datos del segundo experimento ejecutado, que se describe en esta sección.

El experimento fue ejecutado en el año 2004-2005, durante cuatro sesiones. Participaron un total de 32 sujetos, divididos en 6 grupos de trabajo. Los sujetos eran estudiantes que estaban familiarizados con las técnicas aplicadas durante el experimento, ya que en cuarto año de la carrera toman una asignatura relacionada. De todas formas, su conocimiento práctico era limitado. Por este motivo, durante el experimento los sujetos completaban un cuestionario de auto-evaluación con respecto a su experiencia y conocimiento previo en lenguajes de programación.

Durante la primera sesión, se explicaba a los estudiantes de todos los grupos sobre el experimento y la documentación correspondiente que debían estudiar. Los sujetos conocían en todo momento que estaban participando de un experimento, y que los resultados obtenidos serían utilizados como forma de calificación.

En las tres sesiones siguientes, los integrantes de cada uno de los 6 grupos ejecutaban una de las técnicas asignadas (estructural, funcional o revisión) sobre el mismo programa. De esta manera, todos los sujetos ejecutaban cada una de las tres técnicas sobre cada uno de los tres programas. La asignación de los grupos a los días, y de los programas y técnicas a los grupos se realizó de forma aleatoria.

El experimento incluía la documentación de las instrucciones a seguir para cada una las técnicas. Estas instrucciones eran entregadas a los sujetos para que las estudien, y detallaban el paso a paso para aplicar cada técnica, así como los formularios donde se registraban los datos correspondientes. Los estudiantes debían entregar los ejercicios completados (de forma manual) en los formularios provistos, aplicando las tres técnicas propuestas.

Los objetivos específicos del segundo experimento eran investigar:

- La influencia de la visibilidad de la falta. De forma de conocer si una falta no fue detectada porque no se generó un caso de prueba que cause la falla, o porque el sujeto no la identificaba cuando ocurría, se incluyeron nuevos casos de prueba generados por los responsables que aseguraban la detec-

ción de los defectos. Los sujetos ejecutaron los casos de prueba diseñados por ellos mismos, y además los casos de prueba provistos por los responsables.

- La influencia de la técnica y el tipo de falta. Se quiere conocer en qué medida el uso de una u otra técnica de pruebas influía en la detección de una falta.
- La influencia de la posición de la falta. Se quiere investigar si ciertas faltas eran más fáciles de detectar durante una revisión debido a su posición en el código, influyendo en la efectividad resultante.

Se plantearon en particular las siguientes hipótesis nulas. Las primeras cinco corresponden a ambos experimentos, mientras que las últimas dos aplicaban sólo para el segundo.

- La técnica de detección de faltas no impacta en la cantidad de faltas detectadas.
- El tipo de falta no impacta en la cantidad de faltas detectadas.
- El uso de diferentes técnicas de detección de faltas para diferentes tipos de faltas no impacta en la cantidad de faltas detectadas.
- El uso de diferentes técnicas de detección de faltas en diferentes programas no impacta en la cantidad de faltas detectadas.
- Diferentes tipos de faltas en los diferentes programas no impacta en la cantidad de faltas detectadas.
- La visibilidad de las fallas generadas por las faltas no impacta en la efectividad.
- La posición de las faltas no impacta en la efectividad.

La variable de respuesta se definió como la efectividad de la técnica, medida en términos de la cantidad de sujetos que detectaban una falta dada, por cada falta presente en un programa. Dado que se buscaba probar la dependencia entre la efectividad y el tipo de falta, el experimento recolectaba datos de efectividad para cada técnica y falta.

Los factores del estudio eran los siguientes.

- *Tipo de falta*. La cantidad de faltas debía ser la misma para cada programa, y con la misma distribución con respecto al tipo de faltas. Se inyectaron un total de 7 faltas en cada programa. Se utilizaron los tipos de faltas inicialización, control, cosmética y cálculo de la clasificación de Basili [104] (interface y datos no fueron usados en este experimento). Se diferenciaban las faltas de "omisión" (algo que falta) de las faltas de "comisión" (algo que es incorrecto).
- *Técnica de prueba*. Se aplicaron las siguientes técnicas de pruebas:
  - Clases de equivalencia y análisis de valores límite (funcional)
  - Cobertura de sentencias y cobertura de decisión (estructural)
  - Abstracción paso a paso (revisión de código)

  En el segundo experimento existió una diferencia respecto a la aplicación de las técnicas funcional y estructural. Los sujetos primero aplicaban las técnicas para generar sus propios casos de prueba. Estos casos eran utilizados en el análisis del experimento para determinar qué falta detecta cada técnica. A continuación, los sujetos aplicaban los casos de prueba que son provistos por los responsables del experimento, y que detectaban todas las faltas del programa. Esto permitía conocer si la visibilidad de las faltas influye en su detección.
- *Tipo de programa*. Se utilizaron tres programas diferentes (cmdline, nametbl, ntree), que correspondían a dos tipos de software (funcional, datos).
- *Versión del programa*. Este es un nuevo factor que se incluyó en el segundo experimento. Se implementaron dos versiones de cada programa, de forma de poder replicar todas las faltas inyectadas sobre cada programa. Las versiones del mismo programa diferían en las faltas que contenían, pero siempre conteniendo la misma cantidad y del mismo tipo.

Por otra parte, se definieron las siguientes variables que son utilizadas como parámetros.

- *Tamaño del programa.* Cada programa contenía aproximadamente 200 líneas de código (excluyendo líneas en blanco y comentarios).
- *Tipo de sujetos.* Los sujetos eran estudiantes de quinto año de la carrera en Ciencias de la Computación, de la Universidad Politécnica de Madrid, que tenían poca experiencia.
- *Lenguaje de programación.* Se utiliza *C* como lenguaje de programación para ambos experimentos.
- *Tiempo máximo utilizado.* Por razones logísticas, el tiempo máximo para la revisión estaba limitado a 2 horas, mientras que no existía un límite impuesto para las otras dos técnicas.
- *Faltas.* Cada programa contenía el mismo número de faltas del mismo tipo.

B.1.2 Datos recolectados y almacenados

Los datos eran ingresados por los sujetos en los formularios (en papel) provistos por los responsables del experimento. Debían completar un formulario por cada técnica aplicada. Estos datos eran transcriptos por el responsable del experimento en una planilla de cálculo y se ingresaban en un fichero SPSS.

En la Tabla 33 se detallan los datos que fueron ingresados por los sujetos por cada una de las técnicas aplicadas. Además se muestra el mapeo de los datos ingresados por los sujetos en los formularios, y sus correspondientes datos registrados por el responsable del experimento en la planilla de cálculo.

B.2 Fase 2: Instanciar el modelo de calidad de datos

Durante esta fase se llevaron a cabo las mismas actividades que las descriptas para el experimento base MDD-UPV.

La Tabla 17 del Capítulo 8 muestra cuáles son las métricas de calidad que se aplicaron sobre los datos del experimento.

De las 21 métricas de calidad que forman parte del Modelo de Calidad de Datos para Experimentos en Ingeniería de Software, 16 fueron aplicadas sobre los datos del experimento bajo estudio. Dichas métricas se aplican sobre 71 objetos particulares del dominio. Las restantes 5 métricas no resultaron aplicables debido a las características de los datos y del contexto bajo estudio. Por ejemplo, la métrica "Valor Embebido" no fue considerada ya que no se ingresaron datos en formato texto; tampoco se consideró la métrica "Registro contradictorio" al no identificar casos en los cuales pudieran ocurrir contradicciones entre los datos.

La forma de registrar los resultados de las mediciones de calidad, dado que el repositorio de datos utilizado son planillas de cálculo, coincide con el caso del experimento base MDD-UPV.

Técnica	**Datos ingresados en formulario (sujetos)**	**Datos ingresados en planilla de cálculo (experimentador)**
Revisión, Estructural, Funcional	Experiencia relativa con lenguaje de programación C	*Rel. Exp.*
Revisión, Estructural, Funcional	Experiencia absoluta (años)	*Abs. Exp.*
Revisión	Tiempo requerido para construir abstracciones	*Technique application time*
Estructural, Funcional	Tiempo requerido para elaborar casos de prueba	
Revisión	Cantidad de niveles de abstracciones generadas	*No. of Abstraction*

Funcional	Cantidad de clases de equivalencia	*No. of classes*
Estructural, Funcional	Cantidad de casos de prueba generados	*No. of test cases*
Revisión	Hora de comienzo de búsqueda de inconsistencias	*Test case execution time*
	Hora de finalización del experimento	
Estructural, Funcional	Tiempo requerido para ejecutar los casos de prueba	
Estructural, Funcional	Hora de comienza de búsqueda de fallos	*Failure/Fault detection time*
	Tiempo requerido para identificar fallos	
Revisión, Estructural, Funcional	Estimación de porcentaje de fallos encontrados	*% estimated defects*
Revisión, Estructural, Funcional	Confianza sobre la correcta aplicación de la técnica	*Confidence*
Revisión, Estructural, Funcional	Descripción de la falta/falla que fue detectada	*F1 a F7 (1 si fue detectada, 0 en caso contrario)*

Tabla 33: Datos ingresados por sujetos

B.2.1 Definición de las métricas de calidad instanciadas

A continuación se describe cómo se aplica cada métrica de calidad a los datos del experimento bajo estudio.

Valor fuera de rango

Resulta de interés que los valores de los tiempos así como de las cantidades que son ingresadas por los sujetos (abstracciones, clases y casos de prueba) se encuentren dentro de un rango determinado, tal como se muestra en la Tabla 34. En todos los casos, los rangos se establecen a partir de la experiencia (juicio de experto) del experimentador.

A sugerencia del responsable del experimento, se calcula también el tiempo total como la suma de todos los tiempos (aplicación, ejecución y detección) por cada técnica de prueba. Este tiempo no es calculado en las planillas de cálculo durante el experimento, sino que se utiliza de forma de verificar si el tiempo que dedicaron en total para cada técnica se sitúa o no dentro del rango esperado.

Para el caso de la cantidad de abstracciones se podría definir un rango más preciso si se identificara en cada programa la cantidad de niveles de anidación de las estructuras. Sin embargo, esto no puede realizarse debido a que no se cuenta con el código que fue utilizado en el experimento.

Campo	Técnica de prueba	Valor mínimo	Valor máximo
Tiempo de aplicación de la técnica	Funcional y Estructural	0	120
Tiempo de aplicación de la técnica	Revisión	0	180
Tiempo de ejecución de casos de prueba	Funcional y Estructural	0	40
Tiempo de ejecución	Revisión	0	50
Tiempo de detección de faltas	Funcional y Estructural	0	30
Tiempo total	Funcional y Estructural	0	190
Tiempo total	Revisión	0	230
Cantidad de abstracciones	Revisión	2	6
Cantidad de clases	Funcional	5	20
Cantidad de casos de prueba	Funcional y Estructural	10	30

Tabla 34: Rangos definidos

Falta de estandarización

Interesa que los valores de los tiempos sean todos enteros, ya que son ingresados en minutos.

Las cantidades ingresadas (abstracciones, clases y casos de prueba) también deben ser valores enteros, carece de sentido que se ingresen valores con decimales.

Por otra parte, el formato definido para indicar cuando se detecta una falta es el dominio de valores {0,1}.

Registro con errores

Se busca si existen desviaciones en los tiempos y faltas detectadas, comparando los valores registrados con los valores reales. Estos valores son registrados por los sujetos y transcriptos por los experimentadores, por lo que podría ocurrir un error al copiar esos datos en la planilla correspondiente.

Valor fuera de referencial

Resulta de interés que los valores ingresados se encuentren dentro de los referenciales definidos como válidos. Esto sucede para los datos de Programa, Técnica y Versión.

En caso de identificar la existencia de algún valor fuera del referencial definido, también es necesario comprobar si hay registros que utilizan o referencian a alguno de estos valores.

Falta de precisión

Interesa que el porcentaje de faltas y fallas detectadas se registren como un entero con dos números decimales, de forma de lograr una mayor precisión en los cálculos que se realizan con estos valores.

Valor nulo

Se busca conocer si existen valores nulos en la experiencia de los sujetos (relativa y absoluta).

Por otra parte, interesa que no existan valores nulos en los tiempos ni cantidades ingresadas. Notar que los datos ingresados dependen de cada técnica (ver Tabla 33). Además, se debe haber ingresado la estimación de defectos así como la confianza del sujeto para todas las técnicas.

Por último, interesa que los porcentajes de faltas y fallas detectadas hayan sido calculados.

Información omitida

En este caso, si la cantidad de faltas o fallas visibles es 7, entonces no deberían existir valores vacíos en ninguna de las faltas o fallas (F1 a F7). Por otra parte, si la cantidad de faltas o fallas visibles es 6, entonces solo puede existir un valor vacío en alguna de las faltas o fallas (F1 a F7).

Registro faltante

Para cada sujeto deben existir tres registros de ejecución y tres registros de faltas, uno por cada técnica aplicada sobre cada programa. Además, deben existir dos registros de fallas (para las técnicas funcional y estructural).

Regla de integridad de dominio

Para el caso de los porcentajes y estimación de defectos, el dominio válido es de 0 a 100.

Por otra parte, tanto en los datos de la experiencia relativa como de la confianza del sujeto se solicita ingresar un valor entre 0 y 5, mientras que para la experiencia absoluta debe ser un valor mayor o igual a 0 (corresponde a la cantidad de años).

Regla de integridad intra-relación

Se definen las siguientes reglas sobre los datos:

- El porcentaje de faltas/fallas es calculado como la cantidad de faltas/fallas detectas sobre la cantidad de faltas/fallas totales.
- La cantidad total de faltas/fallas es igual a la sumatoria de 1's en los campos F1 a F7.
- Para encontrar fallas visibles se aplican las técnicas Funcional o Estructural, pero no Revisión.
- Si se aplica la técnica Funcional o Estructural sobre la versión 1 del programa cmdline, entonces no se encuentra la falta 1 (F1).
- Si se aplica la técnica Funcional o Estructural, no se ingresa ningún valor en la cantidad de abstracciones.
- Si se aplica la técnica Revisión o Estructural, no se ingresa ningún valor en la cantidad de clases.
- Si se aplica la técnica Revisión, no se ingresa ningún valor en el tiempo de detección de fallas ni en la cantidad de casos de prueba.
- Para un mismo sujeto, la experiencia relativa y absoluta es la misma en las tres técnicas aplicadas.

Reglas de integridad inter-relación

La regla definida establece que, para un mismo sujeto, la combinación (Programa-Técnica-Versión) debe ser la misma para todos sus registros.

Registro duplicado

Cada sujeto debe probar cada programa y aplicar cada técnica por única vez.

Estructura de datos, Formato de datos, Facilidad de entendimiento, Metadata

En estos casos las métricas se instancian de la misma forma que para el caso del experimento base MDD-UPV.

B.2.2 Fase 3: Evaluar la calidad de los datos

En la Tabla 17 del Capítulo 8 se muestra el resultado obtenido luego de la aplicación de cada métrica de calidad sobre los datos del experimento. A partir de la medida obtenida (valor de calidad), es posible identificar cuáles son los problemas de calidad que están presentes en los datos.

Se ejecutan 71 mediciones sobre cada uno de los objetos definidos en la fase anterior, de las cuales:

- El 82% (58 mediciones) se realizan de forma automática, mediante el uso de fórmulas de cálculo.
- El 14% (10 mediciones) se realizan de forma manual por no ser posible o necesaria su automatización. En todos los casos corresponden a revisiones o verificaciones manuales sobre las planillas que contienen los datos. Como parte de estas mediciones, se incluyen las 4 referentes a la Representación e Interpretabilidad de los datos.
- Las mediciones correspondientes al restante 4% (3 mediciones) son aquellas que no es posible su ejecución, debido a que corresponden a mediciones manuales con alto costo (esfuerzo) de implementación. Esto sucede para el caso de Registro con errores (mediciones D5.1, D5.2, D5.3). En todos los casos sería necesario revisar las hojas registradas por los sujetos de forma de comprobar si existen diferencias entre los valores registrados en las hojas y los ingresados en la planilla de cálculo.

Como resultado se encuentra que la medida obtenida es menor a 1,00 o 'Regular' para 37 de las 71 mediciones ejecutadas, indicando la presencia de un problema de calidad en los datos. Estas

mediciones corresponden a 9 métricas de calidad diferentes. Se analizan los resultados obtenidos y se identifican los datos que contienen algún problema de calidad, clasificándose como sigue.

B.2.3 Errores en los datos

Como resultado de la ejecución de 24 de las 37 mediciones se identifican datos erróneos. Se detalla cada caso a continuación.

Falta de estandarización: en cantidad de abstracciones y casos de prueba, experiencia, estimación y confianza (mediciones D2.4 a D2.5, y D2.9 a D2.12)

Se identifican 2 casos en que la cantidad de abstracciones contiene el texto "2 o 3", cuando debería contener un valor entero. Esto demuestra que no comprendieron cómo aplicar la técnica, ya que la cantidad de abstracciones es un número entero que es posible obtener a partir del código.

Por otra parte, se identifica 1 caso en la cantidad de casos de prueba que tiene el valor "-". Luego de consultar con el experimentador se indica que esto representa un valor nulo. Esto se trata en la medición D19.1.

Finalmente, se identifican otros casos en los cuales se ingresan valores con decimales. Esto sucede para la experiencia relativa (19 casos), experiencia absoluta (6 casos), defectos estimados (1 caso) y confianza (4 casos). Luego de consultar con el experimentador, se indicó que los decimales corresponden a casos en los cuales el sujeto ingresa una respuesta en el formulario que no es clara; por ejemplo, si el sujeto marca su respuesta entre el valor "2" y el "3", entonces el experimentador registrará un "2.5".

Valor nulo: en los datos de experiencia, tiempos, cantidades, estimaciones y porcentajes (mediciones D8.1 a D8.11)

Se identifican valores nulos para 11 de las 12 mediciones ejecutadas. El resultado obtenido en cada caso se muestra en la Tabla 35. En todas los casos se verifica la existencia ya sea de celdas vacías o celdas con '-' (según lo indicado por el experimentador).

Campo	Cantidad de valores nulos	Observaciones
Experiencia relativa	1	Se identifica que estos datos son ingresados una vez por cada técnica, cuando deberían ingresarse por única vez por sujeto. Esto se considera en las mediciones D12.10 y D12.11.
Experiencia absoluta	19	
Tiempo de aplicación	6	Notar que los sujetos sobre los cuales se identifica la existencia de valores nulos en estos datos coinciden para varias mediciones.
Tiempo de ejecución	8	
Tiempo de detección	7	
Cant. abstracciones	5	
Cant. clase	1	
Cant. casos de prueba	5	
Estimación de defectos	14	
Confianza	11	
Porcentaje de faltas	5	Los porcentajes son vacíos debido a que no existen los datos base para poder calcularlos.

Porcentaje de fallas	0	

Tabla 35: Resultado de aplicar la métrica Valor Nulo

Información omitida: en faltas y fallas visibles (mediciones D9.1 y D9.2)

En el caso de faltas, se detectan 2 registros que contienen valores nulos en los datos F1 a F7.

Por otra parte, existen 10 registros con 6 fallas visibles y ningún nulo en F1 a F7. De esta forma no es posible saber cuál es la falla que no detectaron. Además, en 3 casos los valores de F1 a F7 son todos 0's, e incluso el valor en porcentaje (0%). Esto genera dudas sobre si estos valores deberían ser también vacíos. El experimentador desconoce el motivo por el cual se omite esta información o existen datos con el valor '0'.

Reglas de integridad intra-relación: incumplimiento de reglas de consistencia (mediciones D12.6, D12.7, D12.8, D12.10, D12.11)

Como resultado de la medición D12.6 se obtienen 10 registros, que coinciden en todos los casos con los resultantes de la medición D9.2. En todos los casos que se cumple la combinación Programa-Técnica-Versión planteada, el valor para F1 es 0, cuando debería tener un valor vacío (debido a que esta falta no es posible detectarla en ese caso). El experimentador desconoce el motivo de este problema de calidad.

Se identifica un mismo registro como resultado de las mediciones D12.7 y D12.8. En este caso se ingresa el valor '75' tanto en la cantidad de abstracciones como en la cantidad de clases. Debido a que para este sujeto el porcentaje de defectos estimados corresponde a un valor nulo, podría haber sucedido que el valor '75' haya sido ingresado en las celdas que no corresponde.

A partir de las mediciones D12.10 y D12.11 se obtienen como resultado 3 y 10 registros con inconsistencias en la experiencia relativa y absoluta respectivamente. Este problema de calidad se debe a que estos datos son solicitados por cada aplicación de cada técnica, cuando debería solicitarse una única vez por experimento. No es posible conocer cuál de los valores ingresados corresponde al valor real.

B.2.4 Valores sospechosos

Se detecta la presencia de 9 valores sospechosos que corresponden a datos para los cuales no es posible asegurar la existencia de un error y fueron analizados junto con el responsable del experimento. Entre los problemas identificados se incluyen:

Valor fuera de rango: en tiempos y cantidades (mediciones D1.1 a D1.9)

En 9 de las 10 mediciones ejecutadas se obtiene como resultado al menos un valor fuera del rango definido. Los resultados se muestran en la Tabla 36.

Para los tiempos se encuentran valores fuera de los rangos definidos en todos los casos. Es por este motivo que se calcula el tiempo total y se define un nuevo rango, de forma de verificar si el tiempo que dedicaron en total para cada técnica se sitúa o no dentro del rango esperado. Se observa de esta forma que de los 31 casos que se situaban fuera de rango para cada tiempo de forma individual, solo 3 se sitúan fuera del rango considerando el tiempo total para técnicas dinámicas, y ninguna para estáticas.

Campo	Cantidad de valores fuera de rango	Rango definido	Rango de valores por fuera del definido
Tiempo aplicación (F, E)	4	[0,120]	[125,150]

Tiempo aplicación (R)	1	[0,180]	[200]
Tiempo ejecución (F, E)	16	[0,40]	[45,60]
Tiempo ejecución (R)	2	[0,50]	[65,90]
Tiempo detección (F, E)	8	[0,30]	[31,47]
Tiempo total (F, E)	3	[0,190]	[200,205]
Tiempo total (R)	0	[0,230]	N/A
Cant. abstracciones (R)	6	[2,6]	[1], [8,42]
Cant. clases (F)	7	[5,20]	[22,32]
Cant. casos de prueba (F, E)	20	[10,30]	[4,9], [33,42]

Tabla 36: Resultado de aplicar la métrica Valor Fuera de Rango.

Según el responsable del experimento, el contar con una gran cantidad de valores fuera de los rangos definidos como válidos, es un indicativo de la probable falta de entendimiento por parte de los sujetos. Posiblemente esté mostrando que la técnica no haya sido correctamente entendida y/o aplicada, y por lo tanto los datos que se ingresan no están de acuerdo a lo esperado.

Otras posibles causas podrían ser una equivocación por parte de los sujetos al completar los datos en las hojas de papel (sobre todo en los valores que llaman más la atención, por situarse de forma más lejana del rango definido). O también que los datos hayan sido transcriptos de forma incorrecta desde las hojas de papel que completan los sujetos a la planilla de cálculo. Sin embargo esto se considera menos probable.

Hay 3 casos que coinciden con los resultados de las mediciones B2.5 y B2.6, ya que al ser valores registrados con un formato no válido se consideran también como fuera del rango definido.

B.2.5 Oportunidades de mejora

Se identifican propuestas de mejora a considerar sobre las 4 mediciones que refieren a las métricas que se aplican sobre el conjunto de datos completo.

Estructura de datos (medición D18.1)

Los datos se registran en una planilla de cálculo y un fichero SPSS. Aplica el mismo diagnóstico y recomendaciones que para el experimento base MDD-UPV.

Formato de datos (medición D19.1)

En general, las diferentes hojas de la planilla de cálculo conservan el mismo formato para registrar los mismos datos. Sin embargo, la nomenclatura utilizada para indicar valores nulos no es siempre la misma. Mientras que en algunos casos las celdas permanecen vacías, en otros se ingresa un '-' para representar la falta del dato.

A partir de las mediciones se identifica también que hay valores decimales ingresados con '.' y no con ',' lo cual impacta en las operaciones que se realizan con estos datos. Por ejemplo, si se desea calcular un promedio de valores, las celdas que contienen decimales con '.' no son considerados y por lo tanto el resultado obtenido será incorrecto.

Facilidad de entendimiento (medición D20.1)

Fue necesario consultar al experimentador para poder interpretar el significado de ciertos datos registrados en las planillas. Por ejemplo, se consultó sobre la relación que existe entre los datos registrados en cada una de las hojas (*"Observable faults", "Failure visibility", "Observed faults"*).

También se consultó sobre la relación existente entre los datos registrados en la planilla de cálculo y los datos exportados a partir del SPSS. No fue posible comprender esta relación, por lo cual los datos del fichero SPSS no fueron utilizados para el análisis de calidad.

Por otra parte, hay datos con sombras de diferentes colores pero se desconoce su significado.

Metadata (medición D21.1)

Aplica el mismo diagnóstico y recomendaciones que para el experimento base MDD-UPV.

B.3 Fase 4: Ejecutar acciones correctivas sobre los datos

Se analiza en conjunto con el responsable del experimento si para los problemas de calidad identificados es posible aplicar acciones de corrección.

Para los casos de errores en los datos no se identifican posibles limpiezas a aplicar. Sería necesario conocer el valor real que deberían tomar, ya que los errores suceden sobre datos ingresados por los sujetos.

En los casos en que se omitió el ingreso información o hay valores nulos, se podrían verificar manualmente los formularios registrados por los sujetos, ya que la omisión puede haber sido por parte del experimentador al transcribir los datos en las planillas de cálculo. Por razones de costo (esfuerzo) asociado, no se llevó a cabo esta acción.

Muchos de los errores identificados suceden sobre datos registrados en la hoja "*Subject Data*". Esta hoja contiene los datos de tiempos y cantidades registrados por los sujetos. Se consultó con el experimentador por este hecho, indicando que esos datos no son utilizados para realizar análisis estadísticos debido a que no se consideran "datos fiables". Los resultados obtenidos a partir de las mediciones comprueban este hecho.

Para el caso de los valores sospechosos, debido a que no existen fuentes de datos sobre las cuales se pueda comparar el valor registrado con el real, no se identifican acciones de corrección posibles.

Se proponen actividades de prevención para todos los casos en los que se identifica la presencia de un problema de calidad, que podrían ser incorporadas para futuras replicaciones del experimento para contribuir en la mejora de su calidad.

Anexo C: Modelo de Calidad de Datos y Aplicaciones

En este anexo se detalla:

- La definición del Modelo de Calidad de Datos (general) para Experimentos en Ingeniería de Software.
- La instanciación del modelo y métricas de calidad para cada uno de los cuatro casos de aplicación: experimento UdelaR, experimentos UPV (base y replicación) y experimento UPM.
- El resultado obtenido a partir de la ejecución de las mediciones en los cuatros casos de aplicación, incluyendo un análisis de los resultados, diagnóstico y acciones.

Por motivos de espacio, este anexo está disponible aquí.

Anexo D: Cuestionario de Satisfacción y Respuestas de Experimentadores

En este anexo se presenta el cuestionario que fue realizado a los experimentadores responsables de los experimentos sobre los cuales se evaluó la calidad de sus datos. También se muestran las respuestas obtenidas.

El objetivo del mismo era conocer la opinión y experiencia respecto el uso y aplicación del modelo y la metodología de calidad. El cuestionario de satisfacción fue construido en base al framework propuesto por Moody [93], [94]. Dicho framework ha sido validado previamente y es ampliamente utilizado para evaluar la calidad de los modelos o métodos en términos de tres variables de satisfacción: usabilidad percibida (PU), facilidad de uso percibida (PEOU) e intención de uso (ITU).

Siguiendo el framework definimos 8 preguntas para medir la PU, 6 para la PEOU y 2 para la ITU. Además, cada experimentador debe indicar 5 aspectos positivos y 5 aspectos negativos respecto al uso y aplicación del modelo de calidad.

D.1 Cuestionario de satisfacción

El cuestionario de satisfacción enviado a los experimentadores es el siguiente.

Evaluación del Uso y Aplicación del Modelo de Calidad de Datos en Experimentos en Ingeniería de Software

Este cuestionario está dirigido a experimentadores en ingeniería de software.

Le ofrece la oportunidad de expresar su opinión sobre el uso y aplicación del Modelo de Calidad de Datos en Experimentos en Ingeniería de Software.

Por favor complete los siguientes datos.

Nombre del experimentador:

Experimento ejecutado:

Fecha en que fue aplicado el modelo de calidad de datos:

Parte 1 – Cuestionario de Satisfacción

Por favor, lea cada sentencia y puntúela en base a su opinión. Los posibles valores de la puntuación son:

- 1= Totalmente en desacuerdo
- 2= Bastante en desacuerdo
- 3= Neutral
- 4= Bastante de acuerdo
- 5= Totalmente de acuerdo

Sentencias	1	2	3	4	5
1. Las métricas de calidad propuestas resultaron sencillas y fáciles de aplicar sobre los datos de mi experimento	O	O	O	O	O

2. Creo que haber aplicado las métricas de calidad incrementa la probabilidad de encontrar problemas de calidad sobre los datos, que podrían ser más difíciles de identificar de otra forma	O	O	O	O	O
3. El modelo de calidad es fácilmente entendible, pudo ser instanciado y aplicado a los datos de mi experimento	O	O	O	O	O
4. En general, encuentro que el modelo de calidad podría ser fácilmente instanciado y aplicado por el analista de calidad sobre los datos de un experimento	O	O	O	O	O
5. Contar con métricas de calidad predefinidas facilita al experimentador conocer los problemas de calidad que tienen los datos	O	O	O	O	O
6. Creo que sería posible que el analista de calidad aplique las métricas de calidad de forma sencilla sobre los datos de otro experimento	O	O	O	O	O
7. En general, encontré la definición y el resultado de la aplicación de las métricas de calidad útiles	O	O	O	O	O
8. En general, el resultado de la aplicación de métricas de calidad facilita el trabajo de evaluación y mejora de la calidad de los datos de un experimento	O	O	O	O	O
9. En mi opinión, el esfuerzo invertido en trabajar con el analista de calidad para que se aplique el modelo y las métricas de calidad sobre los datos de mi experimento se ve redituado por el beneficio obtenido	O	O	O	O	O
10. Definitivamente, quisiera que se aplique el modelo de calidad sobre los datos de mis futuros experimentos	O	O	O	O	O
11. El modelo de calidad instanciado a los datos de mi experimento me pareció claro y sencillo de comprender	O	O	O	O	O
12. En general, creo que contar con métricas de calidad predefinidas proporciona una solución efectiva para evaluar y mejorar la calidad de los datos de los experimentos	O	O	O	O	O
13. Considero que como resultado de la aplicación de las métricas de calidad se pueden encontrar una mayor cantidad de problemas de calidad de forma eficiente	O	O	O	O	O
14. Estoy seguro de que quisiera que se apliquen las métricas de calidad sobre los datos de mis próximos experimentos	O	O	O	O	O
15. En general, creo que contar con un modelo de calidad de datos es una mejora con respecto a utilizar un enfoque *ad hoc*	O	O	O	O	O
16. Recomendaría a otros experimentadores utilizar el modelo y las métricas de calidad para evaluar la calidad de los datos de sus experimentos	O	O	O	O	O

Parte 2 – Opinión

Por favor, mencione 5 aspectos positivos y 5 aspectos negativos sobre las métricas y el modelo de calidad que fueron aplicados para evaluar la calidad de los datos de su experimento.

ASPECTOS POSITIVOS:

1.
2.
3.
4.
5.

ASPECTOS NEGATIVOS:

1.
2.
3.
4.
5.

D.2 Respuestas de los experimentadores

El cuestionario fue enviado a 3 experimentadores (en el caso de los experimentos de UPV el experimentador responsable es el mismo). Se obtuvieron respuestas por parte de 2 de ellos.

D.2.1 Respuesta Dr. Ignacio Panach

A continuación se muestra la respuesta del responsable de los experimentos de UPV (base y replicación).

Nombre del experimentador: Ignacio Panach

Experimento ejecutado: MDD vs. desarrollo tradicional (experimentos base y replicación)

Fecha en que fue aplicado el modelo de calidad de datos: Noviembre y Diciembre 2013

Sentencias	1	2	3	4	5
1. Las métricas de calidad propuestas resultaron sencillas y fáciles de aplicar sobre los datos de mi experimento	O	O	O	O	O
2. Creo que haber aplicado las métricas de calidad incrementa la probabilidad de encontrar problemas de calidad sobre los datos, que podrían ser más difíciles de identificar de otra forma	O	O	O	O	O
3. El modelo de calidad es fácilmente entendible, pudo ser instanciado y aplicado a los datos de mi experimento	O	O	O	O	O
4. En general, encuentro que el modelo de calidad podría ser fácilmente instanciado y aplicado por el analista de calidad sobre los datos de un experimento	O	O	O	O	O
5. Contar con métricas de calidad predefinidas facilita al experimentador conocer los problemas de calidad que tienen los datos	O	O	O	O	O
6. Creo que sería posible que el analista de calidad aplique las métricas de calidad de forma sencilla sobre los datos de otro experimento	O	O	O	O	O
7. En general, encontré la definición y el resultado de la aplicación de las métricas de calidad útiles	O	O	O	O	O
8. En general, el resultado de la aplicación de métricas de calidad facilita el trabajo de evaluación y mejora de la calidad de los datos de un experimento	O	O	O	O	O
9. En mi opinión, el esfuerzo invertido en trabajar con el analista de calidad para que se aplique el modelo y las métricas de calidad sobre los datos de mi experimento se ve redituado por el beneficio obtenido	O	O	O	O	O
10. Definitivamente, quisiera que se aplique el modelo de calidad sobre los datos de mis	O	O	O	O	O

futuros experimentos					
11. El modelo de calidad instanciado a los datos de mi experimento me pareció claro y sencillo de comprender	O	O	O	O	O
12. En general, creo que contar con métricas de calidad predefinidas proporciona una solución efectiva para evaluar y mejorar la calidad de los datos de los experimentos	O	O	O	O	O
13. Considero que como resultado de la aplicación de las métricas de calidad se pueden encontrar una mayor cantidad de problemas de calidad de forma eficiente	O	O	O	O	O
14. Estoy seguro de que quisiera que se apliquen las métricas de calidad sobre los datos de mis próximos experimentos	O	O	O	O	O
15. En general, creo que contar con un modelo de calidad de datos es una mejora con respecto a utilizar un enfoque *ad hoc*	O	O	O	O	O
16. Recomendaría a otros experimentadores utilizar el modelo y las métricas de calidad para evaluar la calidad de los datos de sus experimentos	O	O	O	O	O

ASPECTOS POSITIVOS:

1. Ayudan a encontrar defectos
2. Se pueden automatizar en la mayoría de los casos
3. Cada experimentador puede adaptarlo a su experimento
4. Ayuda a garantizar la calidad de los datos ante terceras personas
5. Da ideas de mejora para futuras replicaciones

ASPECTOS NEGATIVOS:

1. Solo encuentran defectos a posteriori de tener los datos y no durante el proceso de experimentación.
2. Los márgenes para marcar un error son a veces demasiado estrictos, marcando un falso error.
3. Puede que haya métricas subjetivas difíciles de clasificar entre error o acierto.
4. Para gente que no tenga experiencia en el mundo empírico, el aplicar la técnica le puede llevar bastante tiempo de aprendizaje.
5. El hecho de cada experimentador defina sus propias reglas de calidad puede desembocar una mala aplicación de la propuesta.

D.2.2 Respuesta Dr. Diego Vallespir

A continuación se muestra la respuesta del responsable del experimento de UdelaR.

Nombre del experimentador: Diego Vallespir

Experimento ejecutado: Comparación de 5 técnicas de verificación de software (año 2008)

Fecha en que fue aplicado el modelo de calidad de datos: Durante el año 2009

Sentencias	1	2	3	4	5
1. Las métricas de calidad propuestas resultaron sencillas y fáciles de aplicar sobre los datos de mi experimento	O	O	O	O	O
2. Creo que haber aplicado las métricas de calidad incrementa la probabilidad de encontrar problemas de calidad sobre los datos, que podrían ser más difíciles de identificar de otra forma	O	O	O	O	O
3. El modelo de calidad es fácilmente entendible, pudo ser instanciado y aplicado a los datos de mi experimento	O	O	O	O	O
4. En general, encuentro que el modelo de calidad podría ser fácilmente instanciado y aplicado por el analista de calidad sobre los datos de un experimento	O	O	O	O	O
5. Contar con métricas de calidad predefinidas facilita al experimentador conocer los problemas de calidad que tienen los datos	O	O	O	O	O
6. Creo que sería posible que el analista de calidad aplique las métricas de calidad de forma sencilla sobre los datos de otro experimento	O	O	O	O	O
7. En general, encontré la definición y el resultado de la aplicación de las métricas de calidad útiles	O	O	O	O	O
8. En general, el resultado de la aplicación de métricas de calidad facilita el trabajo de evaluación y mejora de la calidad de los datos de un experimento	O	O	O	O	O
9. En mi opinión, el esfuerzo invertido en trabajar con el analista de calidad para que se aplique el modelo y las métricas de calidad sobre los datos de mi experimento se ve redituado por el beneficio obtenido	O	O	O	O	O
10. Definitivamente, quisiera que se aplique el modelo de calidad sobre los datos de mis futuros experimentos	O	O	O	O	O
11. El modelo de calidad instanciado a los datos de mi experimento me pareció claro y sencillo de comprender	O	O	O	O	O
12. En general, creo que contar con métricas de calidad predefinidas proporciona una solución efectiva para evaluar y mejorar la calidad de los datos de los experimentos	O	O	O	O	O
13. Considero que como resultado de la aplicación de las métricas de calidad se pueden encontrar una mayor cantidad de problemas de calidad de forma eficiente	O	O	O	O	O
14. Estoy seguro de que quisiera que se apliquen las métricas de calidad sobre los datos de mis próximos experimentos	O	O	O	O	O
15. En general, creo que contar con un modelo de calidad de datos es una mejora con respecto a utilizar un enfoque *ad hoc*	O	O	O	O	O
16. Recomendaría a otros experimentadores utilizar el modelo y las métricas de calidad para evaluar la calidad de los datos de sus experimentos	O	O	O	O	O

ASPECTOS POSITIVOS:

1. Método riguroso en contraposición a uno ad hoc
2. Métricas preestablecidas que logran encontrar problemas en la calidad que podrían pasarse por alto
3. Método que acompaña al modelo para una correcta aplicación del mismo y que lleva también a encontrar problemas que podrían ser pasados por alto.
4. Punto de partida desde la disciplina de la calidad de datos permitiendo esto tener una visión mucho más amplia sobre la calidad de datos de la que normalmente tienen los experimentadores.
5. Se logran buenos resultados encontrándose problemas en la práctica (es decir, al utilizar el modelo vemos que no es solo algo teórico sobre la construcción del Modelo y el Método sino que realmente se encuentran problemas de calidad). O sea, es aplicable y sirve. Esto impacta positivamente en la credibilidad de los datos de los experimentos. Cuestión para nada menor!

ASPECTOS NEGATIVOS:

1. Es bastante costoso para el analista de calidad su aplicación.
2. El Método y el Modelo aún están inmaduros y en etapa de construcción.
3. Sería bueno contar con una aplicación temprana de alguna parte del Modelo (aún esto no está pronto) donde se atacan problemas de calidad de datos antes de comenzar la ejecución del experimento. Es decir, que el Modelo y el Método ayuden en la prevención de los problemas de calidad de datos.
4. El Modelo no está lo suficientemente validado aún.
5. El Método y el Modelo no han sido publicados y por ende la comunidad de experimentadores en ingeniería de software no lo ha discutido en profundidad.