Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Tesis de Maestría
en Ingeniería en Computación

Medición de la calidad de datos: Un enfoque parametrizable

Elena Martirena

Directora: Ing. Verónika Peralta, PhD

Montevideo, Uruguay
6 de Agosto de 2008

Medición de la calidad de datos:
Un enfoque parametrizable
Elena Martirena

ISSN 1510-7264
Tesis de Maestría en Ingeniería en Computación
Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, 6 de Agosto de 2008

Agradecimientos

Quiero expresar mi gratitud a todas las personas que hicieron posible este trabajo de tesis. En primer lugar mi más sincero agradecimiento a mi directora de proyecto Verónika Peralta por su fe y dedicación al proyecto, por otorgarme su apoyo y brindarme sus conocimientos y guía durante todo el proyecto.

También me gustaría agradecer a los profesores: Adriana Marotta, Raúl Ruggia, Omar Viera y Hugo Naya por haber aceptado ser integrantes del tribunal, por su dedicación y los aportes recibidos.

A otros profesores del INCO que me ayudaron en los inicios y concepción del proyecto y por brindarme apoyo y material cuando fue necesario: Lorena Etcheverry, Marita Urquhart y Regina Motz.

Por haber creído en el proyecto y haberme brindado las herramientas y la oportunidad de desarrollarlo en una aplicación real, a mi gerente Fernando. A mis compañeros de trabajo y amigos Hernán, Gustavo, Ximena y Gabriel por apoyarme en tareas de investigación y brindarme su apoyo cuando fue necesario. Al resto de mis compañeros de oficina por su constante interés y compañía en momentos difíciles, en especial: Natalia, Ylia, Lídice, Fernando B., Vicky, Soledad, Analía, Laura, Andrea y Mónica R. quien desde la distancia continuó brindándome su aliciente y ejemplo para seguir adelante.

A mis queridas amigas por su amor, apoyo y comprensión: Isabel, Diamela, Alicia, Verónica; en especial a mi querida amiga Mónica W. por sus buenos aportes y continuo aliento para continuar con el proyecto. Y a mis queridos excompañeros de Historia Laboral que marcaron un camino en los inicios de mi vida profesional lo que influyó en mi decisión para realizar la maestría.

A toda mi familia que siempre ha estado a mi lado y me ha acompañado en este proyecto, en especial a mis padres, a mis hermanos, a mi tía Silca y a mis sobrinos. ¡Los quiero mucho a todos!

Y por último y de forma muy especial a mi hijo Guillermo por su amor y tolerancia a las horas dedicadas al proyecto y quien con sus alegres juegos junto con sus amigos del barrio Pancho, Facundo, Martín e Iván, hicieron este trabajo de tesis mucho más entretenido. Gracias Guille por ser un sol en mi vida y motivarme todos los días para seguir adelante.

Resumen

En este momento, las bases de datos constituyen uno de los principales activos de las empresas. Los problemas de calidad de datos inducen a errores o falta de precisión en el análisis de los mismos, lo cual puede derivar en un alto costo para la empresa. En tal sentido, en esta tesis nos enfocamos en el estudio de mecanismos de medición de la calidad de los datos. Presentamos un estado del arte sobre medición de algunas dimensiones de calidad y experimentamos en una aplicación real de un área de negocio financiera, con el dominio de aplicación CRM, en un esquema de replicación de bases de datos. Para medir la calidad ponemos en práctica una metodología en la que las métricas de calidad se obtienen refinando las metas de calidad de la organización. Como resultado obtuvimos una biblioteca de métodos de medición de la 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 diferentes aplicaciones. Nuestro enfoque puede ser utilizado en las empresas con diferentes objetivos: estadísticas, particionamiento de las tablas de acuerdo a su calidad, mejoras en la explotación de la información, tareas de data-cleaning, entre otros.

Abstract

At this moment, databases are part of the companies' main assets. Data quality problems induce to errors or lack of precision in their analysis which can derive in a high cost for a company. Regarding this issue, in this thesis we focus in the analysis of data quality measurement mechanisms. We present a state of the art on the measurement of some quality dimensions and we experiment in a real application, a CRM in the area of financial business, in a database replication scheme. In order to measure data quality we put into practice a methodology where the quality metrics are obtained by refining the organization's quality goals. As results we obtained a library of quality measurement methods and a database with the measures obtained. We propose parametric and extensible methods which can be used in different applications. Our approach can be used in other companies, with different objectives: keeping statistics, fragmenting data into segments of similar quality, improving information management, data-cleaning, among others.

Índice

1. Introducción

Este capítulo describe la problemática de medición de la calidad de los datos. Se describen los problemas técnicos a tratar y se presentan brevemente los aportes de esta tesis para resolver dichos problemas.

1.1 Contexto y Motivación

En este momento, las bases de datos constituyen uno de los principales activos de las empresas. Estas bases de datos, suelen tener problemas de calidad, cuyas consecuencias son usualmente experimentadas en nuestra vida diaria, y a pesar de ello, no hacemos las conexiones necesarias con sus verdaderas causas. Por ejemplo, cuando una carta llega a destino retrasada, o nunca llega, por lo general se culpa al servicio postal, sin embargo una mirada más cercana a la información, usualmente revela que las causas de estos problemas se deben a errores en las bases de datos, típicamente a domicilios mal almacenados. De forma similar, la entrega duplicada de correo denota un error de registro duplicado de clientes. Inexactitud y duplicación de datos, son dos ejemplos de los problemas de calidad de las bases de datos [Batini+2006].

La necesidad de mejorar la calidad de las bases de datos ha sido identificada y ha ido en incremento en los últimos años. La expansión del consumo de información a través de la Word Wide Web, el uso de información, tanto a nivel académico como para la industria, han determinado la existencia de varios actores con diferentes perfiles interesados en mejorar la calidad de las bases de datos. Es por esto que muchos proyectos de investigación (por ejemplo los proyectos DWQ [Jarke+1997], Quadris [Akoka+2007] y DWH [Hammer+1995]), iniciativas industriales (por ejemplo de Hewlett-Packard [Clément+2007], Oracle [ORACLE2008] y SAP [Goerk2004]) e iniciativas gubernamentales tanto en Europa [Green2007] como en Estados Unidos [Office2006] han puesto su foco en estudiar los problemas de calidad de las bases de datos.

Existen diferentes enfoques para medir la calidad de los datos. Algunos autores se centran en la definición de dimensiones de calidad en diferentes contextos de aplicación [Wang+1996] [Batini+2006] [Pipino+2002]. Otros autores estudian cómo medirla en forma práctica, por ejemplo, en [Etcheverry+2007] se mide la exactitud de datos en un sistema de Data Warehousing. En un contexto de Sistemas de Integración de Datos, aparece también la necesidad de evaluar la calidad de datos resultantes de sofisticados procesos de integración que combinan datos fuentes de calidad muy variada. Algunos autores se centran en la medición de la calidad de los datos que responden a consultas de usuarios y la satisfacción de las exigencias de dichos usuarios en términos de calidad [Naumann+1999] [Peralta2006].

En esta tesis nos enfocaremos en la medición de la calidad en una aplicación empresarial de la vida real. La aplicación seleccionada es un sistema de información corporativo que registra información de clientes de una Institución Financiera. Esta base de datos data de mucho tiempo, y muchos problemas de calidad han sido causados por los diferentes procesos que ha sufrido la base de datos en el correr del tiempo. En este momento, tanto usuarios como técnicos se encuentran ante la necesidad de mejorar la calidad de la base de datos, para satisfacer las necesidades de los clientes externos, y bajar el nivel de incidencias reportadas por los mismos. Por ejemplo, la información

publicada en la web, debe ser fresca, exacta y completa para que el usuario que se encuentra en su casa y accede a la misma mediante la funcionalidad de "banca electrónica" pueda obtener a través de sus consultas la información que satisfaga sus necesidades. Para poder medir la calidad de datos de nuestra aplicación de contexto vamos tener en cuenta los puntos de vista de los diferentes actores: a) desde el punto de vista técnico; b) desde el punto de vista del negocio y c) desde el punto de vista de personas relevantes de la Institución.

a. Desde el punto de vista técnico

La arquitectura de los sistemas de información, así como las técnicas de modelado y los motores de bases de datos, han ido evolucionando a través del tiempo. La aplicación de contexto seleccionada, data de mucho tiempo, por lo que ha sido migrada múltiples veces, con cambios tecnológicos en la arquitectura, modelo de datos y software de base.

b. Desde el punto de vista del negocio

La forma de registración de los datos, así como las normas de validación han ido cambiando por reglamentaciones y por cambios en las políticas y metas de las empresas. Adicionalmente, según el sector que registra la información en la empresa, existen diferentes objetivos y visiones: área comercial, contable, de análisis de riesgo, entre otras. Asimismo, la expansión de la empresa - por fusión o campañas comerciales - ha derivado en integración de múltiples bases de datos, con diferentes modelos y criterios de validación.

En este trabajo queremos contemplar los requisitos de calidad desde los diferentes puntos de vista y analizar la calidad de los datos que han pasado por las diferentes etapas mencionadas.

c. Desde el punto de vista de personas relevantes de la Institución

Hay muchas personas que ocupan diferentes cargos en la Institución y que por sus tareas pueden llegar a identificar diferentes y múltiples necesidades de calidad de la base de datos. Dichas personas - gerentes de áreas, encargados de sectores – trabajan y procesan la información para planificar, tomar decisiones y hacer informes que pueden ser de uso interno o externo. Para poder cumplir con la demanda de información sea por clientes internos o externos necesitan mejorar la calidad de los datos.

Nos enfocaremos en identificar y estudiar las principales dimensiones de calidad a partir del análisis de los problemas y definición de objetivos de calidad de nuestro contexto de aplicación. Se necesita medir la calidad de los datos en dicha aplicación, identificando dimensiones de calidad relevantes para el caso, definiendo las métricas e implementando métodos de medición que permitan cuantificar la calidad.

1.2 Problemática y objetivos

Una base de datos con problemas de calidad puede acarrear sustanciales impactos económicos y sociales. Tanto a nivel gubernamental como a nivel privado, cada vez más las empresas se están planteando el problema de calidad, ya que en algunos casos puede llevar a pérdidas millonarias. Por ejemplo, un informe emitido en el año 2002 por el *Data Warehousing Institute* sobre calidad de datos, indica que los problemas de calidad le han costado a los negocios en Estados Unidos más de 600 billones de dólares al año [Batini+2006].

Los problemas de calidad de datos que enfrentan en este momento las empresas se deben a diferentes motivos – volumen, diversidad de criterios para su registración, sistemas legados, evolución de la tecnología – por lo que la medición de la calidad depende del contexto de aplicación elegida: lógica del negocio, arquitecturas de los sistemas, entornos de trabajo, entre otros. Por tanto consideramos que resultaría de mucha utilidad contar con técnicas de medición que sean parametrizables a diferentes realidades.

Por lo tanto, el objetivo general de esta tesis es la construcción de una plataforma parametrizable que asista en la medición de la calidad, y que pueda ser aplicable a diferentes empresas para medir la calidad de sus bases de datos. Se quiere construir una biblioteca de funciones de medición (extensible) permitiendo parametrizar dichas funciones a cada contexto concreto.

Para lograr este objetivo, en el desarrollo de la tesis vamos a ir obteniendo resultados intermedios, que sustentarán nuestro objetivo final:

- estado del arte sobre medición de la calidad;
- análisis de objetivos de calidad del contexto de aplicación;
- definición de métricas y métodos para el contexto de aplicación;
- diseño de la plataforma de medición;
- aplicación de la plataforma al caso de estudio.

1.3 Solución propuesta

Para la construcción de la plataforma parametrizable proponemos:

- La construcción de un catálogo de calidad;
- La aplicación de la metodología basada en el paradigma Goal-Question-Metric (GQM) siguiendo un enfoque top-down;
- Los mecanismos necesarios para instanciar los métodos paramétricos definidos.

El enfoque es hacerlo a través de un **caso real**, por lo que comenzamos nuestro trabajo analizando el contexto de aplicación seleccionado, identificando los problemas y definiendo los objetivos de calidad de dicha aplicación. Estos objetivos son asociados a preguntas de calidad, las cuales nos planteamos, tratando de imaginar las preguntas que se harían los diferentes actores que son conocedores de los problemas de calidad de la base de datos seleccionada. Luego descomponemos las preguntas asociando dimensiones y factores, obteniendo como resultado un conjunto de factores instanciados para la aplicación de contexto. Posteriormente vamos refinando dichos factores hasta obtener métricas que permiten cuantificar la calidad de los datos de la aplicación.

Esas métricas surgen de las propuestas en la literatura, pero son instanciadas para adaptarse a las particularidades de la aplicación. Igualmente, se definen e instancian los métodos de medición que permiten medir la calidad de acuerdo a dichas métricas.

Finalmente desarrollamos los métodos para realizar las mediciones sobre la aplicación de la vida real, obteniendo una biblioteca de métodos parametrizables y extensibles. Diseñamos los metadatos para almacenar las mediciones realizadas, de forma de poder explotar los resultados. Dicho metamodelo contempla el registro de información histórica, permitiendo almacenar resultados de múltiples iteraciones de mediciones con diferentes escenarios. Una de las principales virtudes de esta biblioteca es que los

escenarios pueden variar en fechas de medición, dominios de aplicación, áreas de negocio y contexto de aplicación. Otra de las virtudes de esta biblioteca es que es extensible, pudiendo incorporar nuevos métodos de medición, incluso agregando nuevas dimensiones no consideradas inicialmente.

En nuestro trabajo, nos concentramos en el desarrollo de los métodos, aprovechando la plataforma e interfaz gráfica ´Qbox-Foundation´ presentada en [Etcheverry+2008] ya desarrollada. Los métodos desarrollados en el marco de esta tesis, serán invocados desde la interfaz gráfica ´Qbox-Foundation´ y utilizarán las bibliotecas de abstracciones de calidad definidas en el mismo.

Finalmente realizamos las mediciones, obteniendo una base de datos con medidas de calidad de la aplicación de contexto, la cual será puesta a disposición de la empresa para su análisis y apoyo de posibles actividades futuras de mejora continua de la calidad de datos.

1.4 Contribuciones principales

Las principales contribuciones de esta tesis son:

1- *Estado del arte sobre medición de la calidad:* El estudio incluye el relevamiento de las siguientes dimensiones de calidad: *frescura, exactitud, completitud, trazabilidad, integridad y unicidad.* Esto permite analizar, ordenar y jerarquizar la información sobre diferentes criterios de calidad propuestos en la literatura;

2- *Análisis de necesidades de calidad en una aplicación real:* El resultado de este análisis es la identificación de los problemas de calidad del contexto de aplicación y su asociación a factores de calidad descritos en la literatura. Además se identifican las métricas de calidad, que permiten cuantificar las necesidades de calidad acotadas al contexto de aplicación;

3- *Propuesta de un mecanismo parametrizable de definición de métodos de medición:* Nuestro aporte consiste en la definición de métodos de medición parametrizables, que permiten su instanciación para adaptarse a diferentes contextos de aplicación. Asimismo, se definen rutinas de uso genérico que pueden ser invocados desde diferentes métodos, facilitando su implementación. También se diseñan los metadatos que permiten registrar los resultados de la ejecución de los métodos;

4- *Prototipo de una plataforma parametrizable que permita la medición de la calidad de datos:* Este prototipo cuenta con una biblioteca extensible de dimensiones, factores, métricas y funciones de medición, la cual se instancia, en un primer momento, con los conceptos desarrollados para nuestro caso de estudio. Esta biblioteca podrá extenderse, incorporando nuevos conceptos de calidad, según lo requiera un nuevo contexto de aplicación.

5- *Explotación de las medidas obtenidas:* La base de datos con resultados puede ser analizada y explotada por la empresa. El conocer la calidad de los datos, es un aporte en sí mismo, pero además puede ser una entrada para tareas de mejora del diseño del sistema o del tratamiento de los datos.

1.5 Organización del informe

El resto del informe está organizado de la siguiente manera:

En el **capítulo 2** se presenta el estado del arte de las dimensiones seleccionadas para esta tesis. Provee definiciones encontradas en la bibliografía estudiada de las dimensiones seleccionadas: *frescura, completitud, trazabilidad, integridad, unicidad* y *exactitud.*

En el **capítulo 3** se incluye una descripción del contexto de aplicación seleccionado. Se presenta la arquitectura, los procesos de actualización de datos y todo aquello que consideramos necesario para el entendimiento del caso.

El **capítulo 4** describe el enfoque adoptado para realizar la medición de calidad incluyendo el metamodelo de calidad. Análogamente se presenta la lista de objetivos de calidad identificados para el contexto de aplicación y un resumen de los factores y métricas de calidad seleccionados.

El **capítulo 5** detalla la forma de implementar cada una de las métricas seleccionadas: identifica los métodos necesarios para llevar a cabo la medición, detalla cómo se realizará la implementación y la instanciación de los mismos aplicada al contexto de aplicación.

El **capítulo 6** detalla la implementación y las decisiones de diseños tomadas para la integración con la herramienta 'Qbox-Foundation'. Se incluye además los resultados de las mediciones y un breve análisis de los resultados obtenidos. Por último se presentan dificultades encontradas en el momento de la implementación y decisiones tomadas al respecto.

El **capítulo 7** presenta las conclusiones, aportes realizados y trabajos que se pueden implementar en el futuro, relacionados con los productos desarrollados en esta tesis.

Por cuestiones de confidencialidad, los ejemplos o información presentados en este trabajo no coinciden con información real, sino que fueron preparados especialmente para este documento.

2. Estado del arte sobre medición de la calidad

Even though quality cannot be defined, you know what it is
[Robert Pirsig, metafísica de la calidad].

En este capítulo se presenta el resultado de la actividad de relevamiento e investigación sobre técnicas de medición de la calidad de base de datos. Enfocaremos nuestro trabajo en el análisis del estado del arte de las dimensiones de calidad seleccionadas para esta tesis: *frescura, exactitud, completitud, trazabilidad, integridad y unicidad.* En el capítulo 4 mostraremos cómo realizamos dicha selección.

2.1 Visión multidimensional de la calidad

Son muchos los trabajos que definen, modelizan, evalúan o proponen mejoras para la calidad de datos. En general, hay consenso en que la calidad es un concepto multidimensional. En efecto, la calidad suele estudiarse vía múltiples dimensiones que caracterizan diferentes facetas de los datos, por ejemplo, *exactitud, accesibilidad, integridad, precisión, confiabilidad, completitud, consistencia, flexibilidad, trazabilidad, seguridad en el acceso, facilidad de manipulación, relevancia,* entre otros [Pipino+2002] [Wang+1996] [Motro+1998].

No obstante, los diferentes trabajos difieren en las dimensiones de calidad que deberían estudiarse. Wang y Strong afirman que para mejorar la calidad de los datos, es necesario entender qué significa la calidad para los usuarios [Wang+1996] y presentan un ranking de dimensiones de calidad que son más relevantes para éstos: *relevancia, exactitud, interpretabilidad, accesibilidad,* entre otros. Otros autores proponen listas de dimensiones principales para algunos tipos de sistemas o dominios de aplicación, por ejemplo, *completitud, unicidad, consistencia, frescura, exactitud* para Sistemas de Integración de Datos [Akoka+2007], *completitud, credibilidad, exactitud, consistencia e interpretabilidad* para sistemas de Data Warehousing [Jarke+1999], *exactitud, completitud, frescura y consistencia* para Sistemas Web [Gertz+2004]. Otros trabajos se focalizan en el estudio en profundidad de algunas dimensiones, por ejemplo *completitud* [Naumann+2003], *frescura* [Gançarski+2003] [Peralta+2004] o *exactitud* [Etcheverry+2007].

Tampoco hay consenso en cuanto a definiciones y forma de nombrar las dimensiones de calidad. Por ejemplo, en [Motro+1998] se define como *soundness* lo que otros autores definen como *correctitud, precisión, exactitud* o *validez.*

Además para una misma dimensión podemos encontrar en la literatura diferentes definiciones. Tomemos *exactitud* como ejemplo, ya que es una de las dimensiones más estudiada. En [Motro+1998] la describe como "*la información que está disponible contiene los valores correctos*"; mientras que en [Wang+1996] da una definición más completa, describiéndola como que *"los datos son correctos, confiables y certificados como libres de error"*.

En definitiva, a pesar de la cantidad de iniciativas, tanto a nivel gubernamental como privado, para estudiar la calidad de datos, no existe una definición rigurosa y estándar de las dimensiones y factores de calidad. Además, si bien las dimensiones suelen

estudiarse de forma independiente, es ampliamente aceptado que muchas dimensiones están relacionadas y se influencian mutuamente. Algunos trabajos estudian las relaciones entre algunas dimensiones de calidad en sistemas específicos, por ejemplo [Bright+2002] [Ballou+1985]. En otras palabras, la cantidad de dimensiones de calidad existentes y su interrelación, provoca que el estudio de la calidad de datos sea un problema complejo de varias variables.

En esta tesis profundizamos en el estudio de algunas dimensiones de calidad que son relevantes para la aplicación de contexto seleccionada. En las siguientes sub-secciones presentaremos una descripción de los conceptos y terminología utilizada y de las dimensiones, factores y métricas seleccionadas. Para poder referenciarnos a ellas de forma simple y unívocamente, les dimos un nombre en español el cual será utilizado de aquí en adelante: *frescura (freshness), exactitud (accuracy), completitud (completeness), trazabilidad (trazability), integridad (integrity) y unicidad (uniqueness).*

2.2 Conceptos y terminología

En esta sub-sección se describen los principales conceptos y términos de calidad que son utilizados en esta tesis.

2.2.1 Conceptos Básicos

A continuación se incluyen definiciones de los conceptos básicos de calidad.

- **Dimensiones de calidad:** Las dimensiones de calidad capturan facetas de alto nivel de la calidad de los datos (*frescura, exactitud, completitud,* etc.*)* o de los procesos que manipulan esos datos (*tiempo de respuesta, confiabilidad, seguridad,* etc.*).*

- **Factores de calidad:** Un factor representa un aspecto en particular de una dimensión de calidad, por ejemplo, *exactitud* involucra *correctitud semántica, correctitud sintáctica y precisión* [Peralta2006]. Un factor puede ser más adecuado que otro para algún tipo de problema o aplicación.

- **Métricas de calidad:** Una métrica es un instrumento usado para medir cierto factor de calidad, por ejemplo el *porcentaje de datos que no tienen errores de sintaxis* es una métrica del factor *correctitud sintáctica*. Se pueden definir varias métricas para cada factor de calidad.

- **Métodos de medición de la calidad:** Un método es un proceso que implementa una métrica de calidad. Para una métrica pueden haber varios métodos que la calculen, por ejemplo siguiendo diferentes heurísticas. Se definen dos tipos de métodos: i) *métodos de cálculo*: calculan la calidad de un objeto realizando una medición directa, por ejemplo, contando el número de valores nulos de un registro; ii) *métodos de agregación*: calculan la calidad de un objeto compuesto realizando agregaciones de los valores de calidad de las partes de un objeto, por ejemplo: calcular la *edad de una tabla*, promediando la *edad de los registros*.

En resumen, los conceptos de calidad se pueden representar con la siguiente jerarquía, lo cual se muestra en la Figura 1:

- Las dimensiones representan facetas de calidad de alto nivel;
- Cada dimensión puede refinarse en un conjunto de factores que representan aspectos particulares de calidad;
- Cada factor puede medirse con varias métricas;
- Cada métrica puede implementarse con varios métodos de medición.

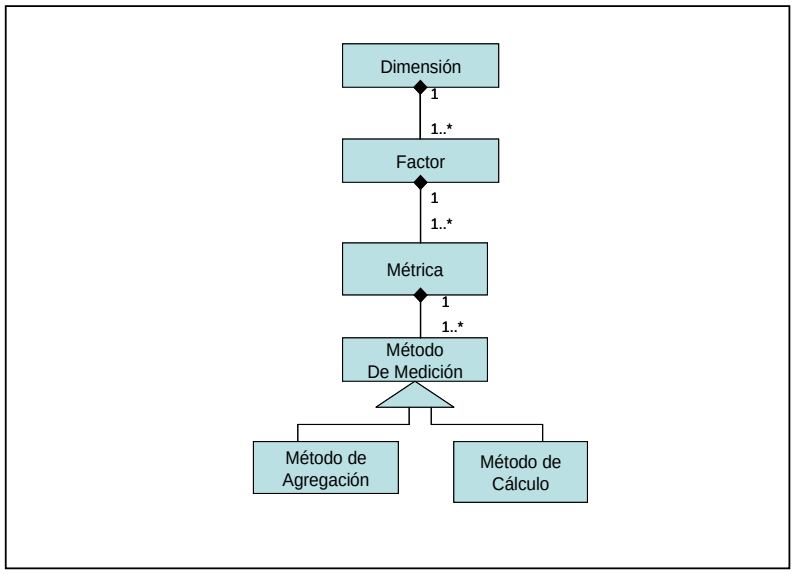

Figura 1- Jerarquías de conceptos de calidad

En la siguiente sub-sección presentamos algunas consideraciones a tener en cuenta en la definición de métricas y métodos de calidad.

2.2.2 Unidades y granularidad de la medición

Al definir una métrica, además de especificar su significado, se debe indicar:

- La *granularidad de la medición*: Indica el nivel de detalle para el que se toman las medidas, por ejemplo: la cantidad de valores nulos *en una tabla*, el tiempo transcurrido desde la última actualización *de un registro*;

- La *unidad de medición:* indica la escala en la que serán expresadas las medidas, por ejemplo, el tiempo de respuesta se puede medir en *días, horas o minutos*; la cantidad de valores nulos se puede medir en *unidades* o en *porcentajes*.

Respecto a la granularidad describimos dos niveles:

- *Granularidad celda*: se realizan mediciones para cada celda de una tabla;

- *Granularidad conjunto*: se realizan mediciones para un conjunto de celdas de una tabla.

Los conjuntos de celdas pueden ser definidos de muchas maneras, determinando granularidades específicas. Para esta tesis consideramos las siguientes granularidades: tabla, área (porción de una tabla), tupla, atributo o celda. Por ejemplo si estamos manejando granularidad tabla, las medidas que obtenemos están asociadas a la tabla

entera (no a elementos contenidos en la tabla) y si manejamos granularidad celda, las medidas que se obtienen corresponden a celdas concretas (un valor de un atributo de una tabla).

Las medidas de granularidad celda son las que nos proveen más detalle sobre la calidad de los datos y la localización precisa de los problemas de calidad. Sin embargo puede ser muy costoso obtener la medida a nivel de celda – en performance, almacenamiento, etc. - aunque algunos emprendimientos industriales indican su importancia a pesar del costo [Laboisse2005]. Para sobrellevar ese costo muchos trabajos proponen métricas de granularidad tabla o atributo (por ejemplo [Naumann+1999]), considerando que la calidad es uniforme en una tabla o incluso en una fuente de datos.

Sin embargo, las fuentes de información son raramente de calidad uniforme, por lo que el principal problema de la granularidad tabla es que no se puede identificar donde están concentrados los problemas, los cuales pueden estar en un conjunto de atributos o tuplas. Una fuente de información puede proveer información de excelente calidad en un tema y de muy poca calidad en otro tema [Motro+1998]. Una forma de mejorar los métodos de medición es definiendo particiones de la base de datos en *áreas* que sean homogéneas con respecto a la calidad de los datos. Las *áreas* son vistas que involucran una selección y/o proyección sobre una tabla, es decir indican conjuntos de tuplas y/o atributos. En [Motro+1998] se proponen algunas técnicas para calcular la homogeneidad de la información, y para particionar una tabla de acuerdo a su homogeneidad. Estas particiones y las métricas definidas sobre sus componentes, proveerán una especificación más refinada de calidad.

Los mecanismos de obtención de medidas son diferentes de acuerdo a la granularidad. Para la granularidad celda las medidas pueden obtenerse ejecutando métodos de cálculo (por ejemplo comparando una calle con un diccionario de calles) o pueden ser resultado de una apreciación (por ejemplo un experto de dominio puede indicar la confianza de un dato). Para la granularidad conjunto (ya sea tabla, área, atributo o tupla) las medidas pueden obtenerse ejecutando métodos de cálculo (por ejemplo determinando la disponibilidad de una tabla accediéndola regularmente), puede ser resultado de una apreciación (por ejemplo la reputación de una tabla puede indicarlo un experto de dominio de acuerdo a su fuente) o pueden obtenerse resumiendo valores de calidad de menor granularidad (por ejemplo calculando la antigüedad de una tabla como promedio de las antigüedades de sus celdas).

Las unidades de medición son diferentes según: i) se calcule o estime un valor de calidad; ii) se resuma un valor a partir de un conjunto de valores de calidad.

En el primer caso las *unidades de medición* pueden clasificarse en las siguientes categorías:

- *Booleano*: Indica si un dato tiene buena calidad o no (0=falso; 1= verdadero), por ejemplo determinando si la dirección de un cliente es correcta;

- *Grados*: Indica la calidad como un valor en determinado intervalo, por ejemplo el porcentaje de confianza de que una palabra haya sido bien reconocida por un OCR;

- *Cantidades*: indica la calidad como un conteo de fenómenos, por ejemplo la cantidad de caracteres mal escritos en una palabra. Un caso especial de esta categoría son las *unidades de tiempo*. Por ejemplo, la antigüedad de un dato, puede medirse en *días, horas o minutos;*

- *Desviaciones de valores*: indica la distancia entre un *valor v* del sistema de información y un *valor v'* de los aceptados como válidos, por ejemplo la distancia entre una calle y el término más próximo en una guía de calles.

A su vez dichos valores pueden ser absolutos (en alguna escala ad-hoc) o normalizados (trasladados al intervalo [0,1]).

En el segundo caso, las medidas de calidad se obtienen aplicando funciones de agregación. Le llamamos *agregación* de las medidas al pasaje de una granularidad a otra mayor, es decir con menor nivel de detalle. Por ejemplo, realizamos una agregación cuando, a partir de medidas a nivel de tuplas de una tabla, calculamos medidas globales para dicha tabla. Las funciones de agregación típicas que van a ser referenciadas en esta tesis son:

- *Ratio*: Esta técnica calcula el ratio o porcentaje de medidas verdaderas (unidades booleanas) o medidas cuyo valor supera un umbral (unidades grados, cantidades o desviaciones). Por ejemplo se puede calcular el ratio de valores que son más frescos que 3 días (antigüedad de cada celda es menor que 3 días). Para medidas booleanas, se cuenta la cantidad de valores verdaderos, por ejemplo, el ratio de valores no nulos.

- *Promedio*: Esta técnica es la más usada; calcula el promedio de un conjunto de medidas, las cuales pueden tener cualquier tipo de unidades (booleanas, grados, cantidades o desviaciones de valores). Notar que si se aplica a medidas booleanas, entonces coincide con el *ratio*;

- *Promedio ponderado*: esta técnica calcula el promedio ponderado de un conjunto de medidas, asignando diferentes "pesos" a los datos dependiendo de su importancia. Se utiliza para indicar que algunas celdas son más importantes que otras, en el cálculo de la medida agregada, por ejemplo la calidad del primer apellido es más importante que la del segundo apellido, para calcular la calidad de la tupla correspondiente a un cliente.

En las siguientes sub-secciones se describe el estado del arte sobre algunas dimensiones de calidad, concretamente *frescura, exactitud, completitud, trazabilidad, integridad y unicidad*. Para cada dimensión detallamos los factores y métricas que han sido propuestos en la literatura y los complementamos con métricas que definimos en el marco de esta tesis. Presentamos el nombre original que fue dado a las dimensiones y factores indicando las referencias de quienes los definieron; pero en muchos casos (por ejemplo las métricas) los conceptos no tienen nombre, por lo tanto los nombramos en el marco de esta tesis. Los métodos de medición serán presentados en el capítulo 5.

2.3 Frescura

La *Frescura (freshness)* intuitivamente introduce la idea de cuán viejos son los datos. Responde a las preguntas: *¿Los datos son lo suficientemente frescos de acuerdo a las expectativas de los usuarios? ¿Cuándo fue generada la información? ¿Tiene cierta fuente los datos más recientes?* [Peralta2004].

La *frescura* es de mucha importancia para los diferentes actores que consumen los datos. Por ejemplo en una aplicación de *banca electrónica,* es muy importante el tiempo que transcurre desde que se realiza la actualización de los saldos de los clientes en el sistema central, hasta que se propaga la actualización para que pueda ser visto por los clientes que están consultando sus cuentas por Internet. El sistema tiene que garantizar la *actualidad* de los datos. Por otro lado, los datos que estamos consultando pueden haber sido actualizados recientemente, pero también pueden ser datos muy antiguos, por ejemplo el teléfono de contacto de un cliente. Es por esto que también es importante conocer la *edad* de los datos.

2.3.1 Factores de frescura

Existen dos factores de calidad relacionados a la *frescura* de los datos [Peralta2004]:

- *Factor actualidad (currency*) [Segev+1990]: Representa cuán actualizados están los datos respecto de las fuentes. La información es extraída de las fuentes, procesada (posiblemente almacenada temporalmente) y luego mostrada a los usuarios, pero los datos en la fuente pueden haber sido modificados en ese intervalo de tiempo y los usuarios pueden recibir información obsoleta. Por lo tanto, el factor *actualidad*, captura ese gap o desfasaje entre la extracción de datos desde la fuente hasta que son mostrados a los usuarios [Peralta2004]. Este factor puede ser aplicado al ejemplo de banca electrónica mencionado anteriormente, ya que el saldo de un cliente puede cambiar en la fuente de datos en el intervalo de tiempo que el cliente lo está consultando por Internet. La *actualidad* indica qué tan desactualizado es el saldo con respecto a la base de datos del banco;

- *Factor edad (timeliness)* [Wang+1996]*:* Representa qué tan viejos son los datos desde su creación o modificación en las fuentes de datos. La *edad* representa el tiempo transcurrido entre la creación o modificación del dato hasta que es consultado, no importando la fecha de extracción. Puede ser muy importante, por ejemplo en explotación de resultados de encuestas. Las preferencias de la gente pueden haber cambiado, por lo tanto la *edad* indica cuán viejas son esas preferencias, desde que fue realizada la encuesta, hasta que se publica su resultado.

En la Figura 2 se muestra cuáles son los intervalos de tiempo que cada factor quiere medir:

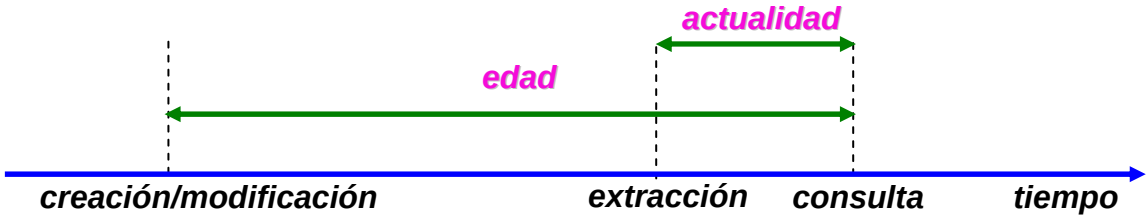

Figura 2- Factores Edad y Actualidad

Es importante señalar que ambos factores se refieren a eventos producidos en la fuente de datos, y no a eventos de la vida real.

2.3.2 Métricas de frescura

A continuación se describen las métricas propuestas para medir la dimensión *frescura*, clasificadas por factor, las cuales hemos extraído de [Peralta2006]:

- *Métricas para el factor actualidad*
  - *Actualidad (currency)* [Gançarski+2003]*:* En sistemas de replicación la *actualidad* mide el retardo de propagación de un dato entre el nodo madre y una réplica, o sea el tiempo transcurrido desde que el dato fue modificado en el nodo madre hasta que se modifica en la réplica. Si los cambios son propagados automáticamente, la *actualidad* es cero*;*

  - *Obsolescencia (obsolescence)* [Gançarski+2003]: En sistemas de replicación la *obsolescencia* mide la cantidad de transacciones que han sido realizadas en el nodo madre pero que no han sido propagados a los nodos suscriptores. Puede ser medido a partir de logs, bases de datos de trazas o de técnicas de detección de cambios, por ejemplo triggers de la base de datos;

  - *Ratio de frescura (freshness-ratio)* [Cho+2000]: Mide el porcentaje de elementos de las réplicas extraídos (tuplas o atributos) que están actualizados respecto al nodo madre. Se mide realizando comparación entre los valores de ambas bases.

- *Métricas para el factor edad*
  - *Edad (age)* [Hammer+1995]: Mide la diferencia de tiempo entre el momento de la consulta y la creación o modificación de los datos. Puede ser medido a partir de logs o bases de datos de trazas. También puede ser medida a partir de timestamps.

En la Tabla 1 se muestra un resumen de los factores y métricas para *frescura.*

Factor de calidad	Métrica	Descripción
Actualidad	Actualidad	Mide el retardo de propagación de un dato entre el nodo madre y una réplica.
	Obsolescencia	Mide la cantidad de transacciones que han sido realizadas en el nodo madre pero que no han sido propagados a los nodos suscriptores.
	Ratio de Frescura	Mide el porcentaje de elementos de las réplicas extraídos que están actualizados respecto al nodo madre.
Edad	Edad	Mide la diferencia de tiempo entre el momento de la consulta y la creación o modificación de los datos.

Tabla 1- Factores y métricas para frescura

2.4 Exactitud

La *exactitud (accuracy)* es una de las dimensiones más estudiadas en la literatura. Intuitivamente la exactitud indica qué tan precisos, válidos y libres de errores están los datos [Peralta2006]. Responde a las preguntas: *¿Estos datos se corresponden con la vida real? ¿Está la información libre de errores? ¿Estos datos son lo suficientemente precisos para nuestras necesidades?*

Es muy importante en diferentes dominios de aplicación, por ejemplo: i) en CRM la información de contacto de un cliente para hacer campañas comerciales o el domicilio de un cliente, para enviarle correspondencia y que no sea rechazada; ii) en sistemas financieros la precisión de los montos, por ejemplo, la cantidad de decimales que se define para el arbitraje de una moneda puede variar significativamente el resultado de la operación de cambio.

2.4.1 Factores para exactitud

La *exactitud* se relaciona con la *correctitud – semántica y sintáctica* - y la *precisión* con la que están representados los datos en el sistema de información. Se definen los siguientes factores:

- *Correctitud Semántica (semantic correctness):* Indica qué tan bien se representan los estados del mundo real en el sistema de información [Shanks+1999]. Calcula la diferencia (o la distancia semántica) que hay entre la información representada en el sistema y la información de la vida real [Kon+1995]. Hay varios problemas de correctitud semántica [Kon+1995]: i) datos que no se corresponden con ningún estado del mundo real (mismembers); ii) datos que corresponden a un estado equivocado del mundo real; iii) datos con errores en algunos atributos. Por ejemplo, la información registrada en la base de datos de una persona, puede ser que referencie a una persona inexistente, que referencie a la persona equivocada o que haya errores en alguno de los atributos (nombre, edad, entre otros);

- *Correctitud Sintáctica (syntactic correctness):* Indica si los datos del sistema de información están libres de errores sintácticos o de formato [Naumann+1999]. Los datos son correctos desde un punto de vista sintáctico si satisfacen las reglas y restricciones impuestas por los usuarios [Peralta2006]. Estas restricciones no siempre están definidas en la base de datos. Por ejemplo, el usuario puede solicitar que el número de teléfono se registre siempre con la característica de telediscado. Hay varios problemas de correctitud sintáctica: i) errores de valores, que pueden ser valores fuera de rango, errores ortográficos y de tipo, (ej. la calle “Julio Herrera” registrada como “Julio Herera”; ii) errores de estandarización, por ej. diferentes tipos de unidades o escala; iii) valores embebidos, en un mismo atributo se registran valores que corresponden a múltiples atributos (ej. en el atributo localidad se registra “Paris-France”;

- *Precisión (precision):* Indica que tan detallados están los datos en el sistema de información [Redman1996]. Calcula la diferencia que hay entre el nivel de detalle de la información registrada en el sistema de información y el nivel de detalle esperado. Por ejemplo, la cantidad de decimales que se registra en una tasa de interés: “10,1408” versus 10” o la fecha de un evento: “2008-01-14 10:55:13” versus “2008-01-14”.

2.4.2 Métricas para exactitud

A continuación se describen las métricas propuestas para medir la dimensión *exactitud*, clasificadas por factor:

- Métricas para el factor *correctitud semántica*
  - *Ratio de correctitud semántica (semantic correctness ratio):* Mide el porcentaje de datos semánticamente correctos en el sistema de información [Motro+1998]. Se calcula dividiendo la cantidad de datos del sistema que coinciden con la vida real versus la cantidad total de datos del sistema. Recopilar los datos de la vida real puede ser muy costoso para las empresas, ya que se requiere de un gran trabajo manual, tales como campañas de recolección de datos de forma telefónica o personal, lo cual requieren de una asignación muy alta de recursos humanos y materiales, para poder llevar a cabo la tarea. Una alternativa es comparar contra un referencial considerado válido u otra base de datos. Por ejemplo: verificar teléfonos de los clientes contra una guía telefónica;

  - *Desviación de correctitud semántica (semantic correctness deviation):* Mide la distancia semántica entre un dato del sistema *(system datum)* y su correspondiente dato en la vida real [Kon+1995]. El cálculo de esa distancia depende del tipo del datos y de la aplicación, por ejemplo para tipos de datos numéricos se puede calcular como la diferencia de ambas cifras mientras que para strings se puede calcular la distancia de edición (contar la cantidad de caracteres a ser cambiados para transformar un string en otro);

- Métricas para el factor *correctitud sintáctica*
  - *Ratio de correctitud sintáctica (syntactic correctness ratio):* Mide el porcentaje de datos sintácticamente correctos del sistema [Pipino+2002]. Este porcentaje se calcula como el número de datos del sistema que satisfacen las reglas sintácticas, dividido la cantidad de datos del sistema. Lo más usual es chequear reglas de sintaxis tales como i) fuera de rango, ii) formato no estándar y iii) valores embebidos;

  - *Desviación de correctitud sintáctica (syntactic correctness deviation):* Mide la distancia sintáctica entre un dato del sistema y algún dato vecino que sea sintácticamente correcto [Fugini+2002], ej. calles versus un padrón de calles;

- Métricas para el factor *precisión*
  - *Escala (scale)*: Mide la precisión asociada a una escala de medición. Para valores numéricos, la precisión es comúnmente asociada a la escala de medición (o ratio de error del instrumento de medición), ej. si el largo de una tabla es 87 ± 1 cm., entonces la imprecisión es de ± 1 cm. [Peralta2006].

  - *Granularidad (granularity):* Mide el número de atributos usados para representar un concepto simple [Redman1996]. Por ejemplo el nombre de una persona puede ser representado por el primer apellido (por

ejemplo Rodríguez) o puede ser representado por un conjunto de atributos tales como el primer y segundo nombre, primer y segundo apellido (por ejemplo María José Rodríguez Men). Una métrica simple consiste en contar los valores no nulos que representan cada concepto.

En la Tabla 2 se muestra un resumen de los factores y métricas para *exactitud.*

Factor de calidad	Métrica	Descripción
Correctitud semántica	Ratio de correctitud semántica	Mide el porcentaje de datos semánticamente correctos en el sistema de información.
	Desviación de correctitud semántica	Mide la distancia semántica entre un dato del sistema y su correspondiente dato en la vida real.
Correctitud Sintáctica	Ratio de correctitud sintáctica	Mide el porcentaje de datos sintácticamente correctos del sistema.
	Desviación de correctitud sintáctica	Mide la distancia sintáctica entre un dato del sistema y algún dato vecino que sea sintácticamente correcto.
Precisión	Escala	Mide la precisión asociada a una escala de medición.
	Granularidad	Mide el número de atributos no nulos que representan un concepto simple.

Tabla 2- Factores y métricas para exactitud

2.5 Completitud

La *completitud (completness)* indica si todos los datos relevantes para un dominio de aplicación han sido registrados en un sistema de información [Gertz+2004]. Intuitivamente, la completitud indica si un sistema de información contiene toda la información de interés. Responde a las preguntas: *¿El sistema de información representa todos los objetos de nuestra realidad? ¿Tenemos todos los datos que describen a nuestros objetos? ¿Tenemos muchos valores nulos?*

La falta de completitud en los datos puede acarrear errores graves para las empresas, pérdidas de oportunidades y de negocios o costos muy grandes. Por ejemplo para sistemas de remates on line la falta de *completitud* puede provocar la pérdida de un negocio, si la oferta no está completa. Asimismo es de suma importancia en diferentes tipos de aplicación, en particular para sistemas que recuperan información de fuentes externas en línea, por ejemplo, en banca electrónica, la *completitud* me indicaría si el saldo que el cliente está consultando tiene los movimientos realizados por cajero automático, por caja y por la misma banca electrónica.

2.5.1 Factores de completitud

Los factores definidos para *completitud* son los siguientes:

- *Cobertura (coverage)* [Naumann+2003]: Describe si todas las entidades de la realidad son identificadas en el sistema de información. Lo que interesa medir es la porción de los datos de la realidad contenidos en el sistema de información. En un ambiente de replicación puede interesar corroborar que todos los

individuos que están en la base de datos del nodo madre, estén en las bases de datos suscriptoras;

- *Densidad (densitiy)* [Naumann+2003]: Mide la completitud de los datos a nivel del registro, o sea cuantos valores de un registro están almacenados y cuántos faltan (valores nulos). Los valores nulos tienen diferentes interpretaciones: i) existe pero no se conoce (por ejemplo un cliente tiene correo electrónico pero no lo conozco), por lo que en este caso los datos son incompletos; ii) no existe (el cliente no tiene correo electrónico), por lo que los datos no serían incompletos; iii) no se sabe si existe (no sé si el cliente tiene correo electrónico), por lo que en este caso no podemos determinar la completitud de los datos.

2.5.2 Métricas para Completitud

Se presentan las métricas agrupadas por factor:

- *Métricas para el factor Cobertura*
  - *Ratio de cobertura (coverage ratio)* [Naumann+2003]: Esta métrica, definida en un ambiente de replicación, mide el porcentaje de individuos presentes en la base de datos del nodo madre, que están replicados en una base de datos suscriptora.
- *Métricas para el factor Densidad*
  - *Ratio de densidad (density ratio)* [Naumann+2003]: Mide el porcentaje de los datos obligatorios que están definidos en el sistema de información (son no nulos y no contienen valores dummy, por ej. ceros o espacios). Una variante a nivel de esta métrica consiste en ponderar los atributos de acuerdo a su importancia, por ejemplo, es más importante que la persona tenga registrado los nombres a que tenga registrado el correo electrónico.

En la Tabla 3 presentamos un resumen de los factores y las métricas para la dimensión *completitud*.

Factor de calidad	Métrica	Descripción
Cobertura	Ratio de cobertura	Mide el porcentaje de individuos presentes en la base de datos del nodo madre, que están replicados en una base de datos suscriptora.
Densidad	**Ratio de densidad**	**Mide el porcentaje de los datos obligatorios que están definidos en el sistema de información (son no nulos y no contienen valores dummy, por ej. ceros o espacios).**

Tabla 3- Factores y métricas para Completitud

2.6 Trazabilidad

La *trazabilidad (traceability)* se define como la propiedad de la información de ser bien documentada, verificable y fácilmente atribuible a una fuente [Wang+1996]. Responde a las preguntas: *¿Qué usuario realizó las modificaciones de un cliente? ¿De qué fuente provino el dato de un cliente?*

Se ha identificado un interés creciente – tanto a nivel empresarial e industrial - en poder identificar el origen de los datos. A nivel empresarial la *trazabilidad* es cada vez más

importante, y es aplicable a varios dominios de aplicación. Por ejemplo, en sistemas financieros, puede ser de mucha utilidad identificar qué cambios se han realizado en los datos y quien realizó cada cambio. En sistemas de integración de datos puede ser muy interesante poder identificar de qué fuente provino la información ya que de ahí se pueden derivar otras propiedades, tales como la confiabilidad de la información en base a la confiabilidad de su fuente. A nivel industrial, en la última versión del motor de bases de datos de Microsoft - Sql Server 2008 – se han incorporado nuevas herramientas de auditoría que facilitarán la *trazabilidad* de las bases de datos [Pan2008].

2.6.1 Factor de verificabilidad

Si bien muchos trabajos resaltan la importancia de la trazabilidad no encontramos definiciones de factores ni métricas para medirla. Por ese motivo vamos a definir un factor y una métrica que nos permita realizar un mínimo de mediciones en los siguientes capítulos:

- *Factor verificabilidad (verifiability):* El factor verificabilidad representa cuán localizables son los datos de actualización de la información. Por ejemplo en sistemas de auditoría, la *verificabilidad* indica el grado en el que se puede identificar la fuente de actualización de los datos y los cambios realizados.

2.6.2 Métricas para Trazabilidad

A continuación se describen las métricas propuestas para medir la dimensión *trazabilidad*, para el factor *verificabilidad:*

- *Ratio de verificabilidad (verifiability ratio):* Mide el porcentaje de elementos de un sistema de información que puede ser atribuible a una fuente de datos y para los cuales se conocen los usuarios que realizaron cada evento de su historia de modificaciones. En la Tabla 4 se presenta un resumen del factor y la métrica seleccionada para la dimensión *trazabilidad*.

Factor de calidad	Métrica	Descripción
Verificabilidad	Ratio de Verificabilidad	Mide el porcentaje de elementos de un sistema de información que puede ser atribuible a una fuente de datos y para los cuales se conocen los usuarios que realizaron cada evento de su historia de modificaciones.

Tabla 4-Factores y métricas para Trazabilidad

2.7 Integridad

La *Integridad (integrity)* o *consistencia (consistency)* indica si se satisface un conjunto de restricciones específicas que establecen la relación entre diferentes elementos de los datos [Redman1996]. Las restricciones más comunes son el chequeo de nulos, unicidad de clave primaria, dependencias funcionales y reglas de dominio.

Las reglas de integridad pueden estar definidas en la base de datos o pueden estar no implementadas, pero son necesarias para la aplicación. Para el primer caso podemos decir que es un problema que está resuelto a nivel de la base de datos; en el segundo caso por lo general se controlan a nivel de los procedimientos de ingreso y actualización de la información. Debido a que estos procedimientos pueden fallar tanto en su

definición como implementación, es que nos interesa medir la integridad de las bases de datos. Por ejemplo podemos definir como regla que si un empleado tiene menos de 3 años de antigüedad en una empresa, entonces el salario no puede ser mayor a 20.000 pesos mensuales. La *integridad* indica si esta regla se cumple para todos los empleados registrados.

2.7.1 Factores para integridad

Existen los siguientes factores:

- *Integridad de dominio (domain integrity)* [Batini+2006]: Controla reglas sobre el contenido de un atributo de una tabla. Por ejemplo para un registro de empleados de una empresa definimos una regla de integridad de atributo simple, que la edad debe ser entre 18 y 120 años;

- *Integridad intra-relación (relation integrity)* [Batini+2006]: Controla reglas sobre la relación entre varios atributos de una misma tabla. Por ejemplo para un registro de personas, que contiene estado civil y datos del cónyuge, si el estado civil es 'casado', entonces el dato 'nombre del cónyuge' debe ser no vacío;

- *Integridad referencial (referential integrity)* [Batini+2006]: Controla la satisfacción de reglas entre atributos de diferentes tablas. La regla más común es la *foreign key constraint* la cual controla que el valor de una celda esté incluido en la clave primaria de la tabla referenciada. También incluye reglas que establecen relación entre diferentes entidades del modelo de datos, pero que no están definidas físicamente en la aplicación de contexto. Estas reglas se controlan por los procedimientos de ingreso de datos;

2.7.2 Métricas para Integridad

Presentamos las métricas para *integridad* agrupadas por factor [Batini+2006]:

- Métricas para el factor *integridad de dominio:*
  - *Ratio de integridad de dominio (domain integrity ratio):* Mide el porcentaje de registros que cumplen con las reglas establecidas para los atributos de una tabla.

- Métricas para el factor *integridad intra-relación:*
  - *Ratio de integridad intra-relación (relation ratio)*: Mide el porcentaje de datos que satisfacen las reglas interrelaciones;

- Métricas para el factor *integridad referencial:*
  - *Ratio de integridad referencial (referencial integrity ratio):* Mide el porcentaje de entidades que cumplen con las reglas de integridad referencial que son definidas en el modelo lógico, pero que no han sido implementadas en el modelo físico.

En la Tabla 5 se presenta un resumen de los factores y las métricas para la dimensión *integridad.*

Factor de calidad	Métrica	Descripción
Integridad de dominio	Ratio de integridad de dominio	Mide el porcentaje de registros que cumplen con las reglas establecidas para los atributos de una tabla.
Integridad intra-relación	Ratio de integridad intra-relación	Mide el porcentaje de datos que satisfacen las reglas interrelaciones.
Integridad referencial	**Ratio de integridad referencial**	**Mide el porcentaje de entidades que cumplen con las reglas de integridad referencial.**

Tabla 5- Factores y métricas para Integridad

2.8 Unicidad

La *unicidad* (*uniqueness)* indica que las entidades que están registradas en la base de datos, no estén representadas más de una vez [Monge+1997]. Responde a las preguntas: *¿Hay entidades repetidas en el sistema de información? ¿Hay datos contradictorios para una misma entidad?*

Las bases de datos pueden contener registros duplicados que corresponden a la misma entidad, pero que no son textualmente iguales. Esto puede ocurrir por diferentes razones: i) regla de unicidad no cumplida; ii) registros duplicados por errores en la clave primaria, lo cual puede ser causados por errores de digitación en el ingreso de la clave primaria (por ejemplo abreviaciones mal hechas) o cambios de criterio en el ingreso; iii) la clave primaria seleccionada, no asegura la unicidad (por ejemplo una empresa que vende celulares, define como clave primaria el número de celular de la persona); iv) una misma entidad se identifica de diferentes maneras (por ejemplo un empleado de una empresa puede identificarse con un número interno en el sistema de liquidación de sueldos, y con la cédula de identidad en el registro de clientes).

La unicidad tiene gran importancia y puede provocar costos directos e indirectos a una empresa. Por ejemplo, la duplicación de un cliente, puede provocar envío de la misma correspondencia múltiples veces al mismo cliente, lo cual deriva en el costo directo de los gastos de envío y en el costo indirecto de mala imagen hacia el cliente. Asimismo en sistemas de toma de decisiones puede acarrear errores en el momento de tomar una decisión: por ejemplo, si una persona está registrada con varios celulares, puedo otorgar nuevos servicios a personas morosas, sin poder identificar que se trata de la misma persona.

2.8.1 Factores para Unicidad

Se identificó el siguiente factor:

- *Unicidad (uniqueness)* [Monge+1997]: Controla si la misma entidad aparece duplicada o multiplicada en la base de datos de forma no exacta: la clave primaria no coincide, pero hay otros datos identificatorios de la entidad, como por ejemplo el nombre de una persona, que no difieren o difieren muy poco. Por ejemplo, “Francisco García” está identificado con las identificaciones ‘123’ y ‘143’.

2.8.2 Métricas para Unicidad

Las métricas para el factor *unicidad* son las siguientes:

- *Similaridad (similarity)* [Batini+2006]: Mide la distancia al registro más cercano, que puede corresponder a la misma entidad, aunque tengan la clave primaria diferente;

- *Ratio de similaridad (similarity ratio)* [Batini+2006]: Mide el porcentaje de entidades que no están duplicados en forma no exacta (con diferente clave primaria).

En la Tabla 6 se presenta un resumen del factor y la métrica seleccionada para la dimensión *unicidad*.

Factor de calidad	Métrica	Descripción
Unicidad	Similaridad	Mide la distancia al registro más cercano, que puede corresponder a la misma entidad, aunque tengan la clave primaria diferente.
Unicidad	Ratio de similaridad	Mide el porcentaje de entidades que no están duplicados en forma no exacta (con diferente clave primaria).

Tabla 6-Factores y métricas para Unicidad

2.9 Síntesis

En este capítulo hemos realizado un relevamiento de las siguientes dimensiones de calidad: *frescura, exactitud, completitud, trazabilidad, integridad y unicidad.* Hemos profundizado en el estado del arte de las mismas, los factores relacionados y hemos estudiado cómo medirlas.

En la literatura, las definiciones de dimensiones y factores de calidad son muy amplias y diversas, y abarcan muchos aspectos de los datos, y diferentes puntos de vista. Es por esto que hemos utilizado una jerarquía de conceptos de calidad, para estudiarlas, listarlas, ordenarlas y estructurarlas en factores, para luego relacionarlas con las métricas de calidad.

Uno de nuestros aportes es brindar información que nos permita posicionarnos y comparar las diferentes definiciones realizadas. Además brindamos referencias a la literatura para poder ampliar la información. Para facilitar la comprensión, unificamos el vocabulario utilizado, dando a cada elemento un nombre único, ya sea realizando la traducción o dándole directamente un nombre.

Por otro lado, encontramos que algunas dimensiones y factores no son muy estudiados en la literatura. Para estos casos hemos tomado algunas decisiones y definiciones; en particular para *trazabilidad* la literatura es muy escasa, por lo que otro de nuestros aportes fue la definición del factor y la métrica relacionada a la dimensión. Asimismo la *unicidad* muchas veces es tratada dentro de *consistencia*, pero dada su importancia, para esta tesis la tratamos como una dimensión independiente, la descompusimos en factores extraídos de la literatura, y buscamos la mejor forma de medirla.

Por último para mejorar los resultados de las mediciones, ya que la calidad de los datos puede no ser homogénea en toda la base, describimos una forma de refinamiento de las mediciones, realizando particiones de las bases de datos en áreas que sean homogéneas con respecto a la calidad de datos.

3. Descripción de la aplicación

En este capítulo se describe la aplicación sobre la cual aplicarán los métodos de medición a desarrollar. El contexto de trabajo es un sistema de información corporativo que registra información de clientes de una Institución Financiera. Se construyó una base de datos maestra y un conjunto de réplicas, una para cada sucursal. Estas bases de datos surgieron de la sincronización de varias bases de datos de clientes existentes en la Institución, en las que por diferentes motivos, la información registrada era diferente. Se formaron equipos multidisciplinarios, con usuarios calificados, técnicos calificados, usuarios finales y profesionales legales, para analizar los datos y fijar procedimientos de priorización de fuentes de datos.

Para la actualización de la base de datos maestra, básicamente se utilizan los siguientes servicios: un software de front-end para el ingreso de clientes en forma unitaria y un proceso de carga masiva de clientes para el ingreso de más de 50 clientes. El resto de las bases de datos se actualizan en un contexto de replicación: una vez que se actualiza la base de datos maestra, se actualizan las réplicas. Se utiliza un software de integración que llamaremos Herramienta de Integración, el cual, mediante servicios, actualiza la base de datos maestra y genera las novedades para las réplicas. Las reglas de control y validación de datos están multiplicadas: en el software de front-end de cada punto de ingreso e integrado a la Herramienta de Integración.

En las siguientes sub-secciones, se presenta la arquitectura de la aplicación, una descripción de las principales tablas y un detalle del proceso de alta de clientes.

3.1 Arquitectura

En esta sub-sección describiremos la arquitectura actual de la instalación, la cual se muestra en la Figura 3.

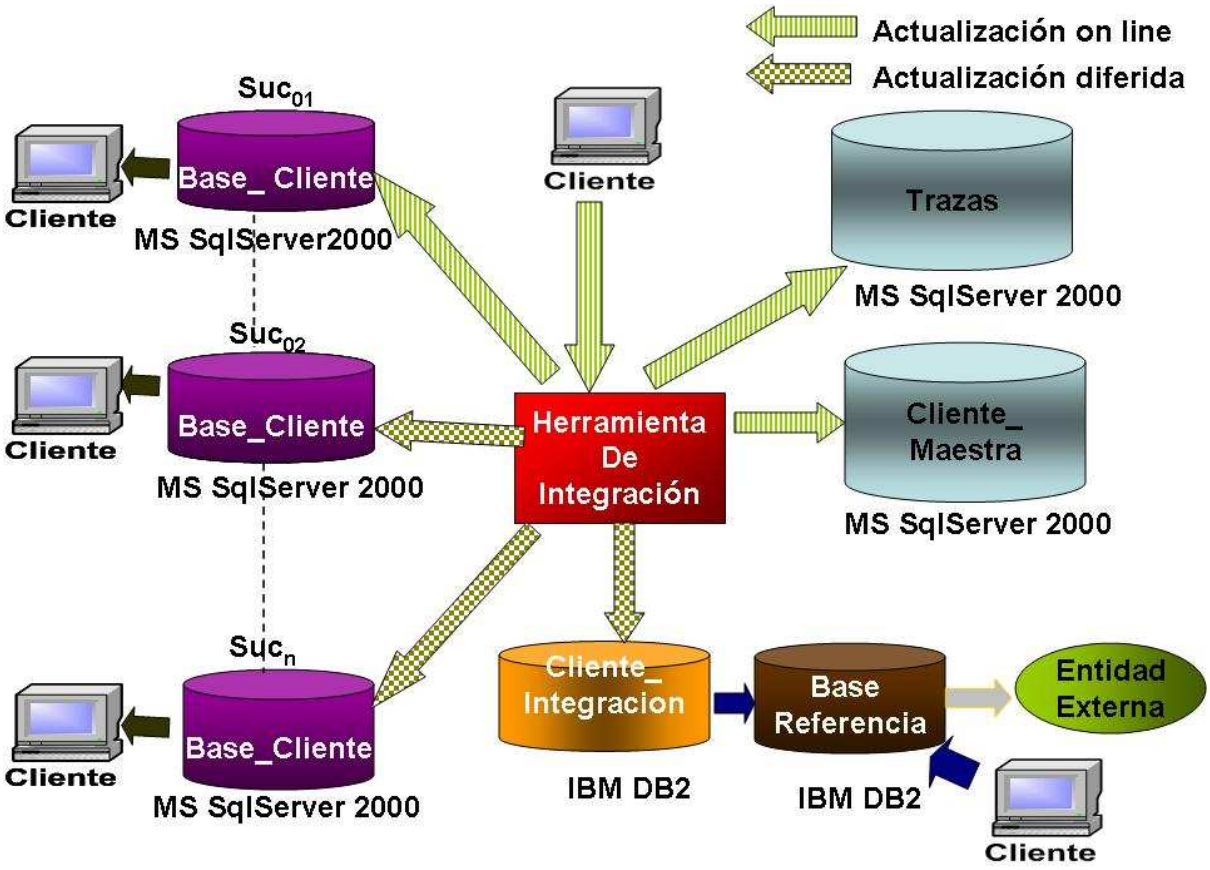

Figura 3- Arquitectura del contexto de aplicación

Los principales repositorios, a los cuales se harán referencia en esta tesis son:

- **Cliente_Maestra**
  Es la base de datos principal - maestra - en la cual reside un modelo de clientes que es actualizado a través de la Herramienta de Integración. Este modelo cuenta con reglas de integridad referencial;

- **Base_Cliente**
  Por cada sucursal de la Institución, existe un servidor local, donde reside una réplica. El esquema de las réplicas es similar al esquema de la base de datos maestra, a excepción de que estas bases de datos no cuentan con integridad referencial. Cada una de estas bases de datos es un suscriptor de la replicación a través de la Herramienta de Integración desde la base de datos Cliente_Maestra.
  A los efectos de esta tesis a las bases de datos de las sucursales las nombraremos $Base\_Cliente_{01}$, $Base\_Cliente_{02}$… $Base\_Cliente_{n}$, donde 01…n representa el código del servidor donde reside dicha base de datos.

- **Cliente_Integracion**
  Es una base de datos con un modelo similar (sistema legado) al de la base de datos Cliente_Maestra y a las bases de datos de las sucursales (Base_Cliente). Esta base de datos es un suscriptor de la replicación a través de la Herramienta

de Integración desde la base de datos Cliente_Maestra. No cuenta con reglas de integridad referencial;

- **Base_Referencia**
  Es una base de datos con un modelo totalmente diferente a las anteriores (sistema legado). No constituye un suscriptor de la replicación desde la base de datos Cliente_Maestra, sino que, mediante procesos por lotes se le envían las novedades desde la base Cliente_Integracion. Adicionalmente, mediante un software de front-end el usuario realiza correcciones de datos sobre la misma. No cuenta con reglas de integridad referencial.

- **Trazas**
  Cada registro que es insertado, modificado o borrado a través de la Herramienta de Integración, registra trazas en la base de datos Trazas.

Las bases de datos Cliente_Maestra, Base_cliente y Trazas residen en servidores abiertos con sistema operativo Ms Windows 2003 y motor de base de datos Ms SqlServer 2000 respectivamente; las bases de datos Cliente_Integracion y Base_Referencia, residen en servidores IBM System i con sistema operativo i5/OS y motor de base de datos DB2.

3.2 Actualización de los datos

En esta sub-sección se presenta la forma de actualizar la base de datos maestra y sus réplicas, lo cual se realiza de tres formas diferentes:

- Un software de front-end que permite el ingreso de información de forma unitaria, instalado en cada una de las sucursales;
- Un proceso de carga masiva (mínimo 50 clientes) que permite el ingreso de clientes desde un archivo con un formato dado. Este proceso se ejecuta desde el servidor donde reside la base de datos maestra, y luego de impactar en ésta, replica al resto mediante la Herramienta de Integración;
- Un conjunto de scripts que corren de forma local en la base de datos maestra y en cada una de las bases de datos replicadas, con el objetivo de realizar correcciones o carga masivas de datos. Estas correcciones se realizan de forma excepcional y surgen por errores identificados por los usuarios o cuando se cambia el modelo de base de datos por motivos reglamentarios o de negocio.

Para la actualización mediante software de front-end, el procedimiento es el siguiente:

1. El usuario ingresa la información desde la sucursal, de forma unitaria mediante el sistema de front-end. Las validaciones del negocio, de tipo, de obligatoriedad y de dominio de los datos se realiza a nivel de reglas implementadas en la herramienta de ingreso;
2. La información es impactada en primer lugar en la base de datos maestra a través de los servicios implementados en la capa de la Herramienta de Integración. Las validaciones de tipo, obligatoriedad y de dominio son realizadas mediante reglas definidas en la base de datos. Las reglas de negocio son validadas a nivel de la capa de la Herramienta de Integración;
3. Luego se impacta la información en la base de datos local, desde donde partió la actualización. Una vez impactadas las dos bases de datos (la maestra y la

local), se le devuelve el control al usuario (“Actualización on line” en la Figura 3);
4. Luego de impactada la base de datos maestra se generan mensajes de replicación para actualizar las bases de datos suscriptoras. Esta replicación se ejecuta de forma diferida, no siendo de impacto para el usuario (“Actualización diferida” en la Figura 3). Si alguna de la replicaciones es fallida, el software de la Herramienta de Integración tiene definidas políticas de reintentos y de alarmas, por si no logra resolver las replicaciones.

Para la actualización mediante carga masiva, el procedimiento es el siguiente:

1. Se ingresa la información de forma masiva mediante un archivo (Planilla MS Excel) con un formato dado, el cual contiene una cantidad mínima de atributos. El resto de los atributos, que están definidos como obligatorios en la base de datos, se completan con valores por omisión en los procedimientos de alta en la base de datos. Las validaciones de tipo, de obligatoriedad y de dominio de los datos de los atributos incluidos en la planilla, se hacen a nivel de reglas implementadas en la planilla Excel;
2. Análogamente la información es impactada, siguiendo los pasos 2. y 4. del esquema de actualización mediante software de front-end.

Para la actualización mediante script, el procedimiento es el siguiente:

1. Se generan los scripts de actualización para los dos ambientes diferentes: MS Sqlserver 2000 y DB2. Los scripts son ejecutados de forma local e independiente en cada una de las bases de datos, ejecutados mediante un servicio de distribución de software.
   En la base de datos maestra las validaciones de tipo, obligatoriedad y de dominio son realizadas mediante reglas definidas en la base de datos. En las bases de datos de réplica las validaciones de tipo y obligatoriedad son realizadas mediante reglas definidas en la base de datos. No se validan reglas del negocio.
2. Luego de las cargas se hacen verificaciones por muestreo de forma manual.

3.3 Descripción de las tablas

En esta sub-sección se detallan las principales tablas de datos, sobre las cuales enfocaremos nuestro trabajo, no considerando necesario detallar las tablas de códigos. Las tablas de datos no registran histórico de datos, por lo que si se produce un cambio, sea en un dato de una tabla, o en una relación, se modifican o se borran los registros sin dejar un histórico.
Para este trabajo sólo se utilizarán la base de datos maestra (Cliente_Maestra), tres de sus réplicas (Base_cliente) y la base de datos de referencia (Base Referencia). No se utilizará la réplica Cliente_Integracion.
En el Anexo I se presentan los esquemas de las bases de datos que son de interés para esta tesis.

3.3.1 Base de datos Maestra

A continuación, se presenta una lista de las tablas de las bases de datos **Cliente_Maestra** y **Base_Cliente**. Las listas de tablas y atributos en dichas bases son iguales. Sólo difieren en que Cliente_Maestra cuenta con integridad referencial y controles de dominio por base de datos, mientras que Base_Cliente no cuenta con estas restricciones a nivel del motor de base de datos.

Tabla	Descripción
Persona	Se registra información común a todas las personas que son clientes en la Institución. La clave es el atributo Identificación, en el cual se registra la Cédula de Identidad, Ruc u otros tipos de documentos de la persona. En esta tabla se registran todos los tipos de clientes: individuales y conjuntos.
Particular	Categorización de Persona. Registra la información propia de Personas Físicas. La clave es el atributo Identificación.
Empresa	Categorización de Persona. Registra la información propia de Personas Jurídicas. La clave es el atributo Identificación.
Cuenta	Relaciona la Identificación del cliente (Identificación de Persona), con el número de cuenta en la Institución. Una persona puede tener de 0..n cuentas relacionadas. La clave es el número de cuenta (atributo Cuenta en la tabla). El atributo que corresponde a la identificación de la persona es Identificacion. Si una cuenta tiene varios titulares, se relaciona un sólo registro del número de la cuenta, con una identificación ficticia que es creada para estos tipos de clientes (Identificación del "cliente conjunto")
Integracion_Cuenta	Relaciona todos los titulares de una cuenta, cuando ésta se abre con más de un titular (cliente "conjunto"). La clave es el atributo Identificación del "cliente conjunto" (atributo IdentificacionCC) + Identificación de la persona que es titular (atributo Identificacion). La persona debe estar registrada anteriormente en la tabla Persona y en Particular o Empresa según corresponda por el tipo de persona. Para un cliente con más de un titular (cliente "conjunto") en la tabla Integracion_Cuenta deben haber N registros (N>1). Una persona puede integrar de 0..n clientes "conjuntos". Un "cliente conjunto" está formado por 2 ó más integrantes.
Documentos	Registra todos los documentos relacionados al cliente (Declaraciones juradas, Estados contables, Certificados de BPS y DGI, entre otros) La clave única es la Identificación del cliente+Tipo de documento (atributos Cliente+Código). Un cliente puede tener 1 sólo documento vigente de cada tipo.
EnvioCorrespondencia	Registra domicilios para el envío de correspondencia para cada cuenta, en caso que éste difiera del domicilio principal del cliente. La clave única es el Número de cuenta + OrdinalCta. (Atributos Cuenta+OrdinalCta). El atributo OrdinalCta se carga – por el momento - siempre con el valor 1.

Tabla	Descripción
GruposEconomicos	Relaciona diferentes clientes que conforman un grupo económico. La clave única es Identificación del cliente (Cliente) + Identificación del grupo (NroGrupo).

Tabla 7- Descripción de las tablas de las bases Cliente_Maestra y Base_Cliente

3.3.2 Base de datos de Referencia

La base de datos de **Base_Referencia** cuenta con una única tabla llamada Trabajo que contiene algunos atributos relacionados a los clientes.

Tabla	Descripción
Trabajo	Se registran algunos datos relacionados a los clientes de la Institución. Estos datos son considerados "conjunto de datos mínimo" con los que debe contar un cliente. La clave es: atributo Identificación y atributo TipoIdentificación. En el atributo Identificación se registra la Cédula de Identidad, Ruc u otros tipos de documentos de la persona. En el atributo TipoIdentificación se indica qué tipo de documento de identificación tiene el cliente: si es un RUC, Cédula de Identidad, Pasaporte, entre otros. Los atributos que son considerados "conjunto de datos mínimo" y que nos interesan para nuestro trabajo son: Nombres y Apellidos, Código de Actividad, País del Documento, Tipo de Persona, Residencia, Sexo y Fecha de Nacimiento. Se cuenta además con una fecha que indica cuando fue informado a la Entidad Externa por última vez. Asimismo se registra el número de cuenta del cliente en la Institución con el mismo criterio que se toma para vincular el cliente con el número de cuenta, que en la tabla Cuenta.

Tabla 8- Descripción de la tabla de Base_Referencia

3.3.3 Base de datos Trazas

Esta base de datos cuenta con varias tablas, pero a efectos de esta tesis sólo nos va a interesar la tabla que se presenta en la Tabla 9.

Tabla	Descripción
TrazaMovimiento	Registra cada movimiento realizado a través de la Herramienta de Integración, incluyendo el siguiente detalle de los movimientos: resultado exitoso o no, fecha y hora del movimiento, identificación del cliente, programa o proceso que lo invoca, usuario que ejecutó la petición de actualización, entre otros.

Tabla 9- Descripción de la tabla de Trazas

3.4 Detalle del proceso de alta de clientes

En esta sub-sección se presenta un resumen de los procesos de alta de cliente según su tipo: individual o con más de un titular (“cliente conjunto”), así como también el otorgamiento del número de cuenta.

3.4.1 Alta de Cliente individual

Cuando se da de alta un cliente individual se ingresa un registro en la tabla Persona y en las tablas Particular o Empresa, según el tipo de persona (atributo TipoCliente de la tabla Persona).
Para el tipo de cliente unipersonal, se registran datos en las tres tablas mencionadas.
La persona tiene una identificación, que puede ser entre otros: Cédula de Identidad, RUC, DNI Argentino o Pasaporte. Los tipos de identificación aceptados están predefinidos.
Para determinar que la persona no está registrada en la base de datos, por lo que se trataría de un alta, se realiza una búsqueda por su identificación. Este dato es la clave primaria de las tablas Persona, Particular y Empresa

3.4.2 Alta de Cliente con más de un titular

Cuando se da de alta un cliente con más de un titular – “cliente conjunto” - se registran cada una de las personas como clientes individuales como se explicó anteriormente. Luego se da de alta un registro en Persona, con un número de identificación ficticio el cual se conforma concatenando un prefijo (‘1234’) al número de identificación de uno de los titulares, agregando ceros entre medio de ambos números hasta completar un string de largo 16 (Ejemplo: ‘1234000012345678’).
Luego se relacionan las identificaciones de los titulares individuales al número ficticio, registrándolos en la tabla Intregracion_Cliente. Se crean tantos registros como personas que participen del “cliente conjunto”. Ejemplo: Para un “cliente conjunto” que esté conformado por las personas con identificación 12345678 y 87654321, se crean dos registros de relación: 1234000012345678-12345678 y 1234000012345678-87654321.
Si la persona seleccionada en primer lugar, ya es titular de un “cliente conjunto”, se selecciona otro de los integrantes. Si resulta que todos los integrantes ya están registrados con sus identificaciones como “cliente conjunto”, entonces se da un número ficticio, que no cumple las reglas.

3.4.3 Alta de cuenta

A un cliente individual o con más de un titular se le relaciona un número de cuenta. El número de cuenta es único y sólo puede tener un titular. Un cliente puede tener de 0 a N cuentas relacionadas.
Esta información se carga en la tabla Cuenta. El número de cuenta está representado por el atributo Cuenta y la identificación del cliente por el atributo Identificacion.

4. Identificación de propiedades de calidad de interés para la aplicación

En este capítulo se presentan los detalles de las actividades para la identificación, elección y especialización de las dimensiones, factores y métricas de calidad que son de interés para la aplicación estudiada.

La calidad de una base de datos, no se estudia en un "todo y absoluto" sino que se mide relativo a los objetivos y necesidades de calidad que puedan tener los usuarios y a los problemas de calidad que pueden identificar los diferentes actores de una organización, incluidos los técnicos responsables de la aplicación.

Al igual que otros proyectos [Jarke+1999] [Akoka+2007] [Etcheverry+2008], que se basan en el paradigma Goal-Question-Metric (GQM) [Basili+1994], seguiremos un enfoque top-down, propuesto inicialmente en dicho paradigma. GQM propone tres niveles de abstracción: i) un nivel conceptual donde los objetivos de calidad de alto nivel son definidos para productos y procesos; ii) un nivel operacional donde un conjunto de preguntas asociadas a factores de calidad, caracterizan la forma de medir un objetivo de calidad, y iii) un nivel cuantitativo donde un conjunto de métricas de calidad es asociado con cada pregunta, con el objetivo de responderla de una forma cuantitativa.

La siguiente sub-sección presenta el enfoque seguido en esta tesis.

4.1 Descripción del enfoque

A continuación presentamos un enfoque top-down aplicado en esta tesis, basado en el paradigma GQM. Nuestra propuesta tiene 5 pasos: 1) identificar objetivos de calidad; 2) descomponer los objetivos en preguntas asociadas a factores de calidad; 3) elegir métricas de calidad que permitan cuantificar las preguntas de calidad; 4) instanciar las métricas adaptándolas al contexto de aplicación y 5) definir métodos de medición. A continuación explicaremos cada paso.

1) Identificar objetivos de calidad

Para lograr una medición útil y eficaz adaptada al contexto de aplicación elegido, que sea de interés para usuarios y técnicos de la empresa, comenzaremos identificando los problemas de calidad de la aplicación y definiendo objetivos de calidad que apunten a solucionar esos problemas.

Desarrollamos esta actividad enfocándonos principalmente en la memoria organizacional de la empresa, con tres objetivos en mente:

i) identificar cambios en la instalación - de arquitectura, estructura de la base de datos, entre otros - que son una de las principales causas de los problemas de calidad, desde la creación de la base de datos a la fecha;

ii) analizar los tipos de incidencias reportadas por los usuarios y clientes de los sistemas, relacionadas a la calidad de datos de la base de datos de clientes;

iii) recopilar documentación relacionada a problemas de calidad detectados y mejoras identificadas, elaborada por técnicos o personas claves de la organización.

En base a la información obtenida, se analizan los datos de las bases de datos y se detectan casos reales de problemas de calidad de datos, relacionados con las causas identificadas.
Una vez identificados los problemas de calidad, definimos los objetivos de calidad especificando: una descripción del objetivo, su propósito, los objetos del sistema de información involucrados y el punto de vista (usuarios, técnicos, etc.).
Por ejemplo: supongamos que definimos el siguiente objetivo de calidad: *mejorar el tiempo que lleva replicar información desde una base de datos maestra a bases de datos suscriptoras*.

Los objetivos de calidad, son refinados con diferentes preguntas en el siguiente paso.

2) Descomponer los objetivos en preguntas asociadas a factores de calidad

Una vez definidos los objetivos de calidad, se los descompone en un conjunto de preguntas que caracterizan la manera de alcanzar los objetivos y los relacionan con factores de calidad. Por ejemplo, en la descomposición del objetivo anterior, se llega a la pregunta *"¿Cuál es el tiempo necesario para actualizar las bases de datos suscriptoras?"*, la cual hace referencia al factor *actualidad*.

Si el objetivo de calidad es muy complejo, se puede descomponer en sub-objetivos más simples. Por ejemplo, el objetivo *"corregir los errores de la base de datos de clientes"* puede descomponerse en los siguientes sub-objetivos más simples: *"corregir errores de digitación en datos de los clientes"*, *"actualizar la información de contacto de los clientes"*, *"unificar datos de clientes duplicados"* y *"completar datos faltantes"*. La idea es descomponer sucesivamente los objetivos hasta llegar a preguntas simples, cada una asociada directamente a un factor de calidad.

Los factores de calidad se obtienen de un catálogo de conceptos de calidad, que contiene las definiciones de dimensiones, factores y métricas de calidad propuestas en la literatura. El catálogo contiene actualmente los conceptos de calidad descritos en el capítulo 2 pero es extensible para incorporar nuevos conceptos. Al seleccionar los factores en el catálogo, se los puede renombrar para ajustarse a la jerga de la aplicación o para adaptarlos a los datos asociados. De esta forma permite al usuario localizar e interpretar más fácilmente los factores "en su idioma", por ejemplo: *correctitud sintáctica, errores sintácticos, sintaxis de las calles, sintaxis de las fechas, correctitud sintáctica de nombres propios.*

En el siguiente paso cada pregunta se refina en un conjunto de métricas.

3) Elegir métricas de calidad asociadas a las preguntas de calidad

A continuación se determina un conjunto de métricas que permitan cuantificar cada factor de calidad seleccionado. Para cada factor definimos las métricas a utilizar por ejemplo *actualidad* o *ratio de correctitud semántica*. Las métricas permiten responder a las preguntas de una manera cuantitativa y de esta forma constituyen un primer paso hacia lograr los objetivos de calidad: realizar un diagnóstico de la calidad del SI. Si se realiza una recolección periódica de valores de cada métrica nos puede dar una idea del comportamiento del sistema frente a cada factor de calidad. Además, si se realizan acciones correctivas (por ejemplo data clearing) se podría realizar un análisis de la evolución de los valores de calidad permitiendo cuantificar los efectos de dichas acciones.

Las métricas también se obtienen del catálogo de calidad. En el siguiente paso se las adapta a las características de la aplicación

4) Instanciar métricas

Si bien hay muchas propuestas de métricas en la literatura, como se describe en el capítulo 2, a la hora de utilizarlas en una aplicación concreta, se necesita adaptarlas a las particularidades de la aplicación y relacionarlas a los objetos que se quiere medir. Esa adaptación, llamada *instanciación de una métrica*, implica diferentes actividades:

- se relacionan las métricas con los objetivos del SI;
- se definen las unidades de medición;
- se fija la granularidad de la medición;
- se renombran las métricas.

Primeramente se determina a qué objetos del sistema de información (tablas, tuplas, atributos, celdas, entre otros) se necesita acceder para responder a las preguntas. Por ejemplo, la métrica anterior podemos calcularla para las tablas *Persona*, *Particular* y *Empresa*. Además de los objetos del SI, se puede necesitar definir parámetros que indiquen cómo realizar la medición, por ejemplo, determinar qué diccionario de calles se utilizará para validar las direcciones de los clientes.

A continuación se definen las unidades de medición. Siguiendo el ejemplo anterior, la *Actualidad* de una réplica puede medirse en días, horas, minutos, segundos, etc. Las unidades de medición pueden ser absolutas (como en el ejemplo anterior), normalizadas respecto a una escala (por ej. en el intervalo [0,1]) o relativas a las expectativas del usuario (porcentaje de satisfacción de sus requerimientos). Por ejemplo, si un usuario indica la demora aceptable de actualización de una réplica, se puede expresar la métrica de la siguiente manera:

(Demora aceptable para el usuario / Actualidad) * 100

La definición de las métricas también implica estudiar la granularidad de la medición y en caso de necesitar agregaciones, elegir la forma más apropiada de realizarlas. Por ejemplo, podemos calcular la *Actualidad* de una tabla como promedio/máximo de la *Actualidad* de los registros.

Para una métrica puede haber más de una forma de instanciarla, ya sea fijando diferentes granularidades, definiendo diferentes unidades o asociándola a diferentes objetos del SI.

El renombrado es análogo al de los factores, para adaptar las métricas a la jerga de la aplicación. Por ejemplo, las métricas anteriores podrían llamarse "*Promedio de tiempo de actualización de una réplica*" y "*Máximo tiempo de actualización de una réplica*".

El último paso consiste en definir métodos de medición para las métricas.

5) Definir métodos de medición

Por último se definen los métodos de medición que implementan los cálculos necesarios para computar las métricas definidas en los pasos anteriores del proceso. Para implementar una métrica se pueden definir varios métodos de medición que sigan diferentes estrategias de cálculo. Podemos definir *métodos de cálculo* o *métodos de agregación* como se describe en el capítulo 2.

De forma análoga a la selección e instanciación de métricas, desarrollamos un catálogo de métodos paramétricos, que pueden ser seleccionados y adaptados para aplicaciones concretas. La instanciación de un método corresponde a proporcionar los parámetros necesarios para su ejecución en una aplicación concreta. Continuando con el ejemplo planteado en los pasos anteriores, para calcular la métrica *Promedio de tiempo de actualización de una Réplica*, se utilizó el método **CalculoActualidad** que recibe como parámetros: base de datos y tabla para la que se quiere medir, un atributo de la tabla que contiene la fecha de actualización de cada registro y la función de agregación a aplicar para obtener una medida de granularidad tabla.

Si el catálogo no cuenta con métodos apropiados, se puede implementar nuevos métodos, tratando de que sean lo más paramétricos posibles siguiendo el espíritu de la reutilización. El catálogo contiene además una serie de rutinas de propósito general que pueden ser invocadas por diferentes métodos.

En la Tabla 10 se presenta un cuadro resumen del enfoque, mostrando la descomposición de un objetivo de calidad hasta llegar a la instanciación de métodos.

Objetivo: ***Mejorar el tiempo de respuesta de la réplica de información desde una base de datos maestra a diferentes bases de datos suscriptoras***			
Pregunta: ¿Cuál es el tiempo para actualizar las bases de datos suscriptoras?		**Factor: Actualidad**	
Métrica	Objetos SI	Método	**Parámetros**
Promedio de tiempo de actualización de una réplica	Persona Particular Empresa	CalculoActualidadPromedio	**Tabla** **Atributo** **Func. Agreg.**
Máximo tiempo de actualización de una réplica	**Persona** **Particular** **Empresa**	**CalculoActualidadMaximo**	**Tabla** **Atributo** **Func. Agreg.**

Tabla 10- Descomposición de un objetivo de calidad en un enfoque top-down

4.2 Metamodelo de calidad

Como hemos mencionado anteriormente, se eligió para instanciar los métodos de medición y rutinas desarrolladas en esta tesis, la plataforma para medición de calidad, llamada ´Qbox-Foundation´ presentada en [Etcheverry+2008], la cual utiliza un metamodelo basado en sucesivos refinamientos del paradigma GQM.

La Figura 4 presenta el diseño del metamodelo, que surge como resultado de la adaptación del metamodelo de 'Qbox-Foundation' a los requisitos de esta tesis. Se señala en los recuadros rojos realizados con guión-punto las adaptaciones realizadas para esta tesis.

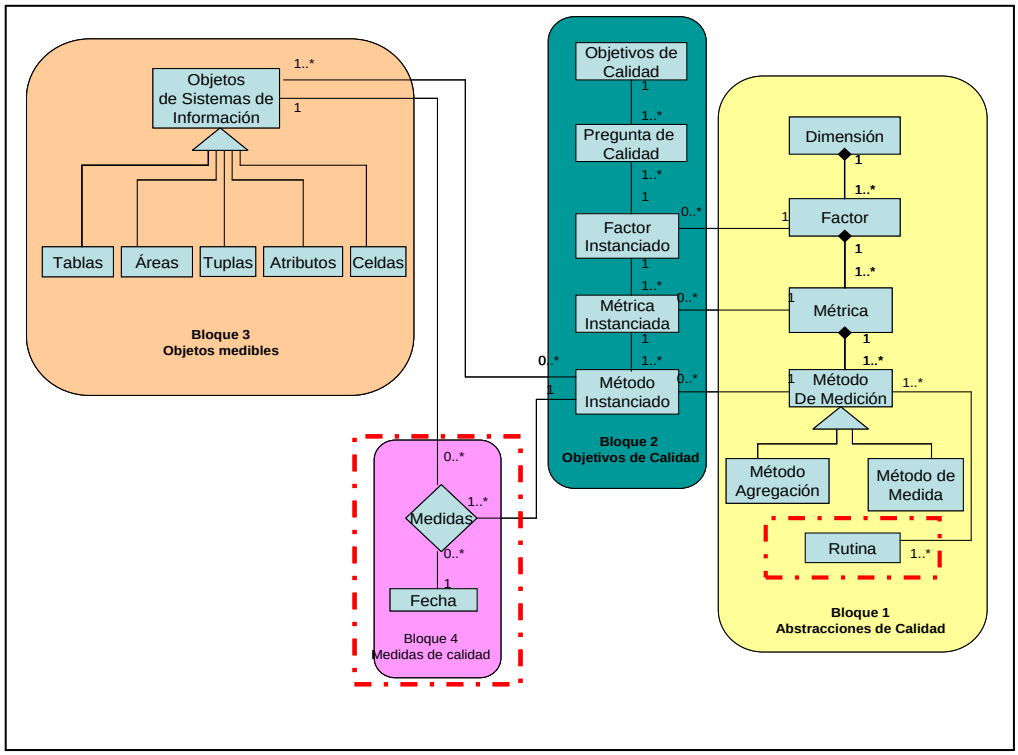

Figura 4- Metamodelo de calidad

El primer bloque de este metamodelo de calidad constituye una librería de tipos de datos abstractos, definidos en el capítulo 2, los cuales serán utilizados para caracterizar los objetivos de calidad definidos. Una de las ventajas de esta librería de abstracciones es que es extensible, en el sentido de que se pueden agregar nuevos conceptos, con el objetivo de extender los aspectos de calidad manejados. Para esta tesis se agregó la representación de las rutinas que serán invocadas por los métodos para realizar las mediciones. La idea es poseer un conjunto de rutinas básicas, que puedan ser utilizadas por varios métodos de medición, de manera de facilitar la implementación.

El segundo bloque representa el enfoque top-down del paradigma de GQM, presentado en la sub-sección anterior. La relación con el primer bloque corresponde a la instanciación de factores, métricas y métodos.

El tercer bloque corresponde a los objetos medibles de nuestro sistema de información los cuales serán utilizados en la instanciación del metamodelo: tabla, tupla, atributo, celda o área (define la partición). Se relaciona con el segundo bloque mediante la instanciación de métricas y métodos, indicando los objetos del sistema de SI para los que se realiza la medición.

El cuarto bloque del metamodelo tiene que ver con el resultado de la medición: se almacenan los valores de calidad obtenidos. La medición permite evaluar las preguntas de calidad y analizar los valores obtenidos en la perspectiva de mejorar la calidad de los objetos medidos. Cada medición de objetivo es llamada un *escenario de medición* y está compuesto por el conjunto de valores asociados al conjunto de preguntas definidas para el objetivo de calidad [Etcheverry+2008]. Para esta tesis cambiamos ligeramente el metamodelo para representar explícitamente la fecha de la medición de manera de guardar históricos de las mediciones. Una de las ventajas de guardar un histórico, es

poder medir la evolución de la calidad. Además cuando las empresas se ven abocadas a tareas para mejorar la calidad (por ejemplo data cleaning), se explota y analiza la información registrada en los históricos y se toman decisiones basadas en dicho análisis.

Las siguientes sub-secciones muestran la utilización del enfoque en una aplicación real.

4.3 Objetivos de calidad identificados

Para poder identificar las dimensiones de calidad que sean de interés medir, comenzamos por identificar los objetivos de calidad y las causas que han ido en desmedro de la calidad de las bases de datos del contexto de aplicación.
Más específicamente, enfocándonos en nuestro contexto de aplicación, y consultando la memoria organizacional de la empresa, sabemos que dicha base de datos, data de mucho tiempo atrás por lo que ha pasado por diferentes procesos los cuales son en general las principales causas de los problemas de calidad existentes:

- Migraciones de la base de datos, cambios en los modelos de datos y procedimientos de registración;
- Cambios en la arquitectura;
- Incorporación de datos de bases de datos de otras empresas, debido a la fusión de empresas;
- Adaptación a nuevas reglamentaciones;
- Campañas comerciales, por las cuales se redefinen algunas reglas de registración de forma de agilizar y facilitar el cumplimiento de las metas comerciales fijadas.

En tal sentido, hemos detectado problemas de calidad de datos, que quisiéramos medir, con el objetivo de conocer los niveles de calidad, y poder adoptar acciones correctivas donde sea necesario.

Hay otros tipos de errores que son causados por errores de diseño en los esquemas de las bases de datos. Algunos de los más comunes son los tipos de datos no adecuados, que no soportan el tipo de dato que se quiere registrar. Por ejemplo: para registrar el nombre de una calle, muchas veces nos encontramos que el largo del atributo no es el adecuado, y se termina abreviando el nombre de la calle, lo cual puede derivar en errores posteriores de interpretación. En este trabajo no vamos a considerar la calidad de los modelos sino que nos vamos a enfocar en la medición de la calidad de los datos.

A continuación describiremos los principales problemas de calidad detectados en el contexto de aplicación seleccionado:

- Antigüedad de la información: La base de datos data de mucho tiempo, por lo que mucha información registrada no está actualizada. Uno de los problemas que se han detectado es la devolución de envío de estados de cuenta, por no estar actualizado el domicilio de la persona. El principal objetivo de la Institución es *contar con información actualizada sobre los clientes*.

- Adaptación a nuevos requisitos: Existen reglamentaciones (del gobierno, casa matriz, entidades reguladoras, etc.) o cambios por nuevas reglas de negocio, que exigen la incorporación de nuevos datos y reglas de validación de la información. La información que ha sido registrada con anterioridad a la entrada en vigencia de dichos cambios, no ha pasado, naturalmente, por los procedimientos de control ahora existentes, por lo que hay segmentos de información en la base de datos que

no cumplen con ciertas reglas de control que existen en el día de hoy, en especial, muchos atributos se completan con valores *dummy*. Se quiere controlar que *todos los datos cumplan con las reglas de validación existentes*.

- Mantener actualizadas las réplicas. Las réplicas solían estar desfasadas entre sí. Para solucionar este problema se diseñó la arquitectura actual y nuevos procedimientos de actualización. Otro objetivo de calidad es *verificar si las réplicas están actualizadas con respecto al nodo madre*. En particular, dadas las exigencias del mercado también se requiere que la información esté disponible de forma inmediata.

- Cambios en los modelos de datos. En las diferentes migraciones de la base de datos y herramientas de front-end se ha cambiado el modelo de base de datos. Tal es el caso de los domicilios: antes se registraba en un mismo atributo la calle, número de puerta, localidad y país; en la actualidad, existe un atributo para cada uno de estos conceptos. Sin embargo, algunos registros viejos pueden desviarse del formato, por ejemplo en localidad puede estar registrado el valor “Paris-Francia”. Otro error, es que algunos atributos obligatorios que fueron agregados - como por ejemplo el número de puerta - pueden haber quedado sin datos. Interesa *llevar todos los datos al formato actual*.

- Desvíos de la información respecto a la realidad. Se han realizado fusiones con otras empresas, volcando otras carteras de clientes a la base de datos de la Institución y se han realizado varias migraciones de la base de datos. La diferencia de los modelos de datos y el costo de procedimientos que insumiría la verificación de la información real ha ido en desmedro de la calidad de la base de datos, ya que se han asumido valores por omisión. Nuestro objetivo es *detectar desvíos de la información registrada con respecto a la información real*.

- Valores por omisión y comodines. Los cambios en el modelo de datos y procedimientos de registración, provocaron el completado de algunos atributos obligatorios con valores por omisión (dummies) o se han adoptado “comodines” para saltar los controles de clave primaria o foránea. Por ejemplo: en una migración se cambió la identificación del cliente (de cuenta a identificación de la persona), y en casos que no se conocía el documento de la persona, se utilizaron valores “comodines”. Se quiere *detectar y completar la omisión de datos obligatorios*.

- Clientes duplicados. Una consecuencia del problema anterior, sumado a errores de digitación es que una misma persona figura con varias identificaciones. En este caso, nuestro principal objetivo es *detectar y consolidar los clientes registrados múltiples veces*, buscando similitudes entre los clientes registrados.

- Trazabilidad del ingreso de los datos. Para el ingreso de información, existen múltiples fuentes de ingreso y actualización (como se describe en la sección 3.2), cada uno de los cuales cuenta con diferentes procesos de actualización y registro. Cuando se detectan errores se desea conocer la fuente de información para poder tomar acciones correctivas. Se define como objetivo *conocer de dónde proviene la información*.

En la Tabla 11 presentamos la lista de los objetivos identificados.

Objetivo	Descripción
1	Contar con información actualizada sobre los clientes
2	Controlar que todos los datos cumplan con las reglas de validación existentes.
3	Verificar si las réplicas están actualizadas con respecto al nodo madre
4	Llevar todos los datos al formato actual y con valores permitidos
5	Detectar desvíos de la información registrada con respecto a la información real
6	Detectar y completar la omisión de datos obligatorios
7	Detectar y consolidar los clientes registrados múltiples veces
8	Conocer de dónde proviene la información

Tabla 11- Objetivos de calidad

La etapa siguiente es la descomposición de los objetivos en preguntas de calidad y la asociación de las mismas a factores y métricas de calidad lo cual es presentado en la siguiente sub-sección.

4.4 Definición de preguntas de calidad

En esta sub-sección descomponemos las metas presentadas en la sub-sección anterior en un conjunto de preguntas, que apuntan cada una, a describir un aspecto particular del problema. Luego asociamos estas preguntas a factores de calidad.

Para favorecer la definición de preguntas analizamos previamente la información registrada en las bases de datos, detectando valores incorrectos o fuera de dominio, valores mal formateados o valores que no respetan las reglas de validación o las preferencias de los usuarios.

El objetivo 1 se refiere a la antigüedad de los datos y su relación con la vida real, ya que las bases de datos con que estamos trabajando datan de mucho tiempo y la información registrada que no ha sido refrescada puede ocasionar diferentes trastornos en la empresa. El objetivo 2 surge de la falta de datos para algunas entidades, el completado de atributos obligatorios y el incumplimiento de las reglas de validación; está relacionado con los agregados de nuevos atributos por reglamentación o por negocio. El objetivo 3 se debe al ambiente de replicación con que estamos trabajando; queremos medir la eficiencia de dichos procedimientos desde dos puntos de vista: la actualización de las réplicas y la pérdida de datos durante la replicación. El objetivo 4 posee 3 facetas: saber si los atributos están bien registrados para un concepto, tratar de identificar errores de domicilios e identificar datos que estén fuera de los rangos permitidos. El objetivo 5 intenta corregir los errores de los datos de los clientes a nivel general, comparando con la información de la vida real. El objetivo 6 surge por los cambios en el modelo de las bases de datos; se van asumiendo datos por omisión para cubrir nuevos atributos de los que no se conocen valores para los datos históricos. Por otro lado, el objetivo 7 busca medir el nivel de duplicación de los datos. Por último, el objetivo 8 busca identificar y cuantificar las fuentes de los datos, de manera de poder trazar la historia de un registro, desde su creación y todas las sucesivas modificaciones.

La Tabla 12 resume las preguntas definidas y los factores de calidad asociados.

Objetivo	Pregunta		Factor
1	1	¿Cuán antiguos son los datos de los clientes?	Edad
	2	¿Los datos de los clientes se corresponden con la realidad?	Correctitud Semántica
2	3	¿Los datos cumplen con las reglas de validación existentes?	Integridad de dominio
	4	¿Todos los datos obligatorios están registrados?	Densidad
	5	¿Las entidades cumplen con las reglas de integridad referencial?	Integridad referencial
3	6	¿Las réplicas están actualizadas con respecto al nodo madre?	Actualidad
	7	¿Está funcionando bien la Herramienta de Integración, replicando todos los elementos? ¿O se pierden elementos en la replicación?	Cobertura
4	8	¿Están completos todos los datos agrupados bajo un concepto?	Precisión
	9	¿Están los datos de domicilios registrados en el atributo que corresponde?	Correctitud Sintáctica
	10	¿Están los datos dentro de los rangos permitidos?	Correctitud Sintáctica
	11	¿Los datos tienen valores válidos?	Correctitud Sintáctica
5	12	¿La información que tengo en las bases de datos, se corresponde con el mundo real?	Correctitud Semántica
6	13	¿Cuántos valores nulos o dummy tengo?	Densidad
7	14	¿Tengo clientes repetidos en mi base de datos?	Unicidad
8	15	¿Puedo identificar la historia de modificaciones de un registro?	Verificabilidad

Tabla 12- Asociación de los objetivos de calidad identificados a factores de calidad

En la sub-sección siguiente se seleccionan métricas para cada factor de calidad.

4.5 Selección de métricas de calidad

A continuación, para cada factor de calidad seleccionado se definen las métricas a utilizar. De las métricas existentes para cada factor, se eligieron las más adaptadas para cada pregunta. En todos los casos la selección de la métrica apropiada fue bastante directa. La dificultad mayor estuvo en la descomposición de los objetivos en preguntas y su asociación a factores de calidad. Destacamos que, para la pregunta 6, encontramos dos métricas que resultaron apropiadas y que respondían a aspectos complementarios, por lo que decidimos medir ambas. En la Tabla 13 se presenta un cuadro resumen que resultó del proceso de selección.

Dimensión	Factor de calidad	Métrica	Preguntas
Frescura	Edad	Edad	1
	Actualidad	Actualidad	6
		Ratio de Frescura	
Exactitud	Correctitud Semántica	Ratio de correctitud semántica	2
		Desviación de correctitud semántica	12
	Correctitud Sintáctica	Ratio de correctitud sintáctica	9;10
		Desviación de correctitud sintáctica	11
	Precisión	Granularidad	8
Completitud	Densidad	Ratio de densidad	4; 13
	Cobertura	Ratio de cobertura	7
Trazabilidad	Verificabilidad	Ratio de verificabilidad	15
Integridad	Integridad referencial	Ratio de integridad referencial	5
	Integridad de dominio	Ratio de integridad de dominio	3
Unicidad	Unicidad	Similaridad	14

Tabla 13- Dimensiones, factores y métricas de calidad que son de interés para la aplicación estudiada

En el siguiente capítulo, desarrollaremos la instanciación para cada una de las métricas relacionando con los objetos del sistema de información y describiremos los métodos o programas de medición.

5. Instanciación de métricas y métodos de medición

Las siguientes sub-secciones presentan las métricas de calidad instanciadas para la aplicación y los métodos de medición correspondientes. Por una cuestión de organización, las agrupamos por dimensión de calidad.

Para cada métrica definimos a qué objetos del sistema de información del contexto de aplicación está relacionada y determinamos la granularidad con que se medirá. Podemos definir varias instanciaciones de la misma métrica. Por ejemplo para la métrica *edad* definimos dos instancias, ambas calculadas como agregaciones de las edades de los registros. Luego definimos los métodos de medición para poder calcular dichas métricas y presentamos un resumen de las instanciaciones de cada método. En algunos casos es necesario definir rutinas genéricas - las cuales son detalladas en el Anexo II - que pueden ser utilizadas en varios métodos; por ejemplo la función que verifica si una celda tiene valor nulo.

5.1 Instanciación de métricas y métodos de medición de Frescura

Para la dimensión *Frescura* se instancian las métricas *edad, actualidad y ratio de frescura*, las cuales se detallan a continuación.

5.1.1 Métrica Edad para factor Edad

La *edad* mide la antigüedad de un dato, o sea el tiempo transcurrido desde que fue modificado por última vez hasta la fecha de medición (now).

Tomaremos medidas de granularidad **tabla**. Para determinar la *edad* de una tabla, calcularemos la *edad* de cada registro y luego aplicaremos una función de resumen, por ejemplo promedio o máximo.

Se plantean dos instanciaciones, ambas calculadas como agregaciones de las *edades* de los registros:

1) Edad-Promedio de tiempo en días
2) Edad-Máximo tiempo en días

En la aplicación, hay tres tablas para las que nos interesa medir la *edad*:

- Persona
- Particular
- Empresa

Se usará la fecha de última actualización de cada registro que está almacenada en un atributo del propio registro llamado *Fecultmod*.

Para realizar la medición, implementamos el método Calcular_Edad que calcula la *edad* de cada registro y realiza la agregación. Su cabezal es el siguiente:

Calcular_Edad (BM: BaseDatos, T: Tabla, A: Atributo, F: FuncionAgregacion)

Este método, recibe como parámetro la base de datos y la tabla para la cual se quiere medir, un atributo que contiene la fecha de modificación de cada registro y la función de agregación a aplicar.

El método calcula la *edad* de cada registro usando la fecha de última actualización del mismo (parámetro *A*) según la fórmula, EdadRegistro = (Now-A) donde Now es la fecha en la cual se realiza la medición. Luego aplica la función de agregación (parámetro *F*). Si la fecha de última actualización (parámetro *A*) es nula, no se considera para el cálculo.

Contaremos con una rutina genérica Chequear_Nulo, definida en el Anexo II para determinar si una celda es nula.

La Tabla 14 resume las instanciaciones, indicando el nombre de la métrica instanciada, los objetos del sistema de información para los cuales se toman medidas, el método de medición y sus parámetros.

Métrica instanciada	**Objeto de la medición**	**Método**	**Parámetros método**
Edad-Promedio de tiempo en días	Persona	Calcular_Edad	**- BM: Cliente_Maestra** **- T: Persona** **- A: FecUltMod** **- F: Average**
	Particular	Calcular_Edad	**- BM: Cliente_Maestra** **- T: Particular** **- A: FecUltMod** **- F: Average**
	Empresa	Calcular_Edad	**- BM: Cliente_Maestra** **- T: Empresa** **- A: FecUltMod** **- F: Average**
Edad-Máximo tiempo en días	Persona	Calcular_Edad	**- BM: Cliente_Maestra** **- T: Persona** **- A: FecUltMod** **- F: Max**
	Particular	Calcular_Edad	**- BM: Cliente_Maestra** **- T: Particular** **- A: FecUltMod** **- F: Max**
	Empresa	**Calcular_Edad**	**- BM: Cliente_Maestra** **- T: Empresa** **- A: FecUltMod** **- F: Max**

Tabla 14- Instanciación de métricas y métodos para Edad, factor Edad

Restricciones:

- El resultado se medirá en días. No se puede medir en horas, ya que la hora no es un dato que se registre en el atributo *FecUltMod*.

- Sólo se puede medir para las tres tablas principales, por no contar con la fecha de última actualización de los registros en el resto del modelo.

5.1.2 Métrica Actualidad para factor Actualidad

La *actualidad* mide el retardo de propagación de un dato entre el nodo madre y una réplica, o sea el tiempo transcurrido desde que el dato fue modificado en el nodo madre hasta que se modifica en la réplica.

Tomaremos medidas de granularidad **tabla**. Para determinar la *actualidad* de una tabla, calcularemos la *actualidad* de cada registro y luego aplicaremos una función de resumen, por ejemplo promedio o máximo.

Se plantean dos instanciaciones, ambas calculadas como agregaciones de la *actualidad* de los registros:

1) Actualidad-Promedio de tiempo en días
2) Actualidad-Máximo tiempo en días

En cada réplica, hay tres tablas para las que nos interesa medir la *actualidad*:

- Persona
- Particular
- Empresa

Se usará la fecha de última actualización de cada registro que está almacenada en un atributo del propio registro llamado *Fecultmod*.

Para realizar la medición, implementamos el método Calcular_Actualidad que calcula la actualidad de cada registro y realiza la agregación. Su cabezal es el siguiente:

Calcular_Actualidad (BM: BaseDatos, B: BaseDatos, T1: Tabla, T2: Tabla, A: Atributo, F: FuncionAgregacion)

Este método, recibe como parámetro las bases de datos de réplica y maestra, las tablas de ambas bases para la cual se quiere medir, un atributo que contiene la fecha de modificación de cada registro y la función de agregación a aplicar. Notar que el nombre del atributo es idéntico en ambas bases de datos.

El método calcula la *actualidad* de cada registro de la tabla (parámetro *T2*) para la base réplica (parámetro *B)*. Para el cálculo realiza una comparación de la fecha de última actualización del registro (parámetro *A*) de las tablas (parámetros *T1* y *T2*) de ambas bases de datos (parámetros *BM* y *B*), según la fórmula, ActualidadRegistro = (B.T2.A – BM.T1.A). Luego aplica la función de agregación (parámetro *F*). La búsqueda de registros en la Base_Maestra se realiza por clave primaria.

No se aplica la fórmula de cálculo si:

i. la fecha de última actualización (parámetro *A*) de cualquiera de las dos tablas es nula;
ii. el registro no existe en la réplica;
iii. la fecha de última actualización de la tabla de Cliente_Maestra es mayor a la fecha de última actualización de la tabla de Base_Cliente.

Los últimos dos casos corresponden a escenarios en los que el registro no ha sido replicado. Para determinar si una celda es nula se utiliza la rutina genérica Chequear_Nulo definida en el Anexo II. En resumen, se realizan las siguientes instanciaciones:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Actualidad-Promedio de tiempo en días	Persona	Calcular_Actualidad	**- BM: Cliente_Maestra - B: Base_Cliente$_{01}$ - T1, T2: Persona - A: FecUltMod - F: Average**
	Particular	Calcular_Actualidad	**- BM: Cliente_Maestra - B :Base_Cliente$_{01}$ - T1, T2: Particular - A: FecUltMod - F: Average**
	Particular	Calcular_Actualidad	**- BM: Cliente_Maestra - B :Base_Cliente$_{01}$ - T1, T2: Empresa - A: FecUltMod - F: Average**
Actualidad-Máximo tiempo en días	Persona	Calcular_Actualidad	**- BM: Cliente_Maestra - B :Base_Cliente$_{01}$ - T1, T2: Persona - A: FecUltMod - F: Max**
	Particular	Calcular_Actualidad	**- BM: Cliente_Maestra - B :Base_Cliente$_{01}$ - T1, T2: Particular - A: FecUltMod - F: Max**
	Empresa	**Calcular_Actualidad**	**- BM: Cliente_Maestra - B :Base_Cliente$_{01}$ - T1, T2: Empresa - A: FecUltMod - F: Max**

Tabla 15- Instanciación de métricas y métodos para Actualidad, factor Actualidad

Se realizaran estas instanciaciones para todas las réplicas consideradas para esta tesis: BaseDatos$_{01}$, Base_Cliente$_{02}$ y Base_Cliente$_{03}$.

Restricciones:

- El resultado se medirá en días. No se puede medir en horas, ya que la hora no es un dato que se registre en el atributo *FecUltMod*.
- Sólo se puede medir para las tres tablas principales, por no contar con la fecha de última actualización de los registros en el resto del modelo.

5.1.3 Métrica Ratio de Frescura para factor Actualidad

El *ratio de frescura* mide el porcentaje de datos que están actualizados, o sea el porcentaje de elementos de una tabla de una réplica que coinciden con los registros de la tabla en el nodo madre.

Tomaremos medidas de granularidad **tabla**, y se plantea la siguiente instanciación:

1) Ratio de Frescura

Para instanciar esta métrica, elegimos todas las tablas de datos que son replicadas mediante la Herramienta de Integración:

- Persona
- Particular
- Empresa
- Intregracion_Cliente
- Cuenta
- Documentos
- EnvioCorrespondencia
- GruposEconomicos

Para realizar la medición, implementamos el método Calcular_Ratio_Frescura que compara los registros de las tablas de la base de datos del nodo madre y de las réplicas, contando las que coinciden. Su cabezal es:

Calcular_Ratio_Frescura (BM: BaseDatos, B: BaseDatos, T1: Tabla, T2: Tabla)

Este método, recibe como parámetros las bases de datos maestra (BM: Cliente_Maestra) y de réplica (B: Base_cliente) y las tablas para la cual se quiere medir (T1 de BM y T2 de B).

El método calcula el *ratio de frescura* de cada registro, comparando atributo a atributo de las tablas elegidas (parámetros T1 y T2), donde T1 reside en el nodo madre (parámetro BM) y T2 reside en la base de datos del nodo replicado (parámetro B). Luego aplica la fórmula Ratio_frescura = Cantidad_atributos_actualizados/ Cantidad_atributos_registro.

Contaremos con una rutina genérica Comparar_Atributos la cual es definida en el Anexo II. Esta rutina compara dos registros devolviendo la cantidad de atributos iguales.

En resumen, se realizan las siguientes instanciaciones:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de Frescura	Persona	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: Persona**
	Particular	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
			- T1, T2: Particular
	Empresa	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: Empresa**
	Intregracion_ Cliente	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: Intregracion_Cliente**
	Cuenta	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: Cuenta**
	Documentos	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: Documentos**
	Envio Correspondencia	Calcular_Ratio _Frescura	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **- T1, T2: EnvioCorrespondencia**
	Grupos Economicos	**Calcular_Ratio _Frescura**	**- BM: Cliente_Maestra** **- B :Base_Cliente$_{01}$** **-T1, T2: GruposEconomicos**

Tabla 16- Instanciación de métricas y métodos para Ratio de Frescura, factor Actualidad

Se realizarán estas instanciaciones para todas las réplicas consideradas para esta tesis: Base_Cliente$_{01}$, Base_Cliente$_{02}$ y Base_Cliente$_{03}$.

5.2 Instanciación de métricas y métodos de medición de Exactitud

Para la dimensión *Exactitud*, se instancian las métricas *ratio de correctitud semántica, desviación de correctitud semántica, ratio de correctitud sintáctica, desviación de correctitud sintáctica* y *granularidad*, las cuales se detallan a continuación.

5.2.1 Métrica Ratio de Correctitud Semántica para factor Correctitud Semántica

El *ratio de correctitud semántica* mide el porcentaje de datos semánticamente correctos en el sistema.

Se tomarán medidas de granularidad **tabla** y se plantea la siguiente instanciación:

1) Ratio Correctitud Semántica-Porcentaje de clientes correctos de la muestra

Para comparar con el mundo real, vamos a reutilizar un trabajo de recolección, corrección y limpieza de datos que se realiza de forma mensual por usuarios expertos, cuyo alcance es un segmento de clientes de la Institución. La información corregida de clientes se registra en la **Base_Referencia,** presentada en la sección 3.3.2**.** La misma será utilizada para medir el *ratio y desviación de correctitud semántica* de la base de datos.

La **Base_Referencia** cuenta con una sola tabla que contiene menos datos que la base de datos Cliente_Maestra, por lo que para instanciar esta métrica, elegimos todas las tablas de datos que contienen algún dato en común con la **Base_Referencia:**

- Persona
- Particular
- Empresa
- Cuenta

Para realizar la medición, implementamos el método Calcular_Ratio_Correctitud_Semantica que compara cada registro y cuenta las coincidencias. Su cabezal es el siguiente:

Calcular_Ratio_Correctitud_Semantica (BM: BaseDatos, B: BaseDatos, T1: Tabla, T2: Tabla)

Este método, recibe como parámetros las bases de datos Cliente_Maestra (parámetro BM) y Base_Referencia (parámetro B) y las tablas para las cuales se quiere realizar la medición (parámetro T1 para Cliente_Maestra y parámetro T2 para Base_Referencia).

El método compara atributo a atributo de los registros de las tablas elegidas (parámetros T1 y T2), las cuales residen en el nodo madre (parámetro BM) y Base_Referencia (parámetro B) respectivamente. Luego aplica la fórmula Ratio_Correctitud_Semantica= Cantidad_datos_coincidentes/ Cantidad_datos_total, donde: Cantidad_datos_coincidentes es la cantidad de datos del registro que coinciden entre Base_Referencia y Base_Maestra y
Cantidad_datos_total es la cantidad de datos del registro que se están comparando.

Internamente para realizar la comparación de atributos invoca la rutina Comparar_Atributos definida en el Anexo II, la cual compara dos registros, devolviendo la cantidad de atributos iguales. Notar que sólo se comparan los atributos que tienen igual nombre, ya que ambos registros pueden contar con otros atributos.

En resumen, se realizan las siguientes instanciaciones para el método definido:

Métrica instanciada	**Objeto de la medición**	**Método**	**Parámetros método**
Ratio Correctitud Semántica-Porcentaje de clientes correctos de la muestra	Persona	Calcular_Ratio _Correctitud_Semantica	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Persona** **- T2: Trabajo**
	Particular	Calcular_Ratio _Correctitud_Semantica	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Particular** **- T2: Trabajo**
	Empresa	Calcular_Ratio _Correctitud_Semantica	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Empresa** **- T2: Trabajo**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
	Cuenta	**Calcular_Ratio _Correctitud_Semantica**	**- BM: Cliente_Maestra - B: Base_Referencia - T1: Cuenta - T2: Trabajo**

Tabla 17- Instanciación de métricas y métodos para Ratio de Correctitud Semántica, factor Correctitud Semántica

Consideraciones:

- La Base_Referencia no tiene la totalidad de los clientes, por lo que el universo será acotado a los clientes contenidos en la misma. Los resultados luego se extrapolan a la totalidad de registros.
- La estructura de la Base_Referencia, contiene menos atributos que Cliente_Maestra, por lo que los atributos a comparar se restringen a los de la primera.

5.2.2 Métrica Desviación de Correctitud Semántica para factor Correctitud Semántica

La *desviación de correctitud semántica* mide la distancia semántica entre un dato del sistema y su correspondiente dato en la vida real.

Se tomarán medidas de granularidad **celda**. Se plantea la siguiente instanciación:

1) Desviación de correctitud semántica- Distancia entre celdas

Al igual que para el *ratio de correctitud semántica*, para medir la *desviación de correctitud semántica* y debido a lo costoso que sería comparar con la vida real, utilizaremos la **Base_Referencia,** presentada en la sección 3.3.2.

Seleccionamos los siguientes atributos - clasificados por tabla - para instanciar esta métrica, los cuales están presentes en la Base_Referencia y Cliente_Maestra:

- Tabla Persona:
  - Nombre
  - RamoActividad

- Tabla Particular:
  - Sexo
  - Primer apellido
  - Segundo apellido
  - Primer nombre
  - Segundo nombre
  - Fecha Nacimiento

Para realizar la medición, implementamos tres métodos que calculan la *desviación de correctitud semántica* de cada celda, dependiendo del tipo de datos (fecha, alfanumérico y numérico):

- Calcular_Desviación_Correctitud_Semantica_Fecha cuyo cabezal es:
  Calcular_Desviación_Correctitud_Semantica_Fecha (BM: BaseDatos, B: BaseDatos, T1: Tabla; T2: Tabla, A: Atributo)

- Calcular_Desviación_Correctitud_Semantica_Alfanumérico, cuyo cabezal es:
  Calcular_Desviación_Correctitud_Semantica_Alfanumérico (BM: BaseDatos, B: BaseDatos, T1: Tabla; T2: Tabla, A: Atributo)

- Calcular_Desviación_Correctitud_Semantica_Numérico, cuyo cabezal es:
  Calcular_Desviación_Correctitud_Semantica_Numérico (BM: BaseDatos, B: BaseDatos, T1: Tabla; T2: Tabla, A: Atributo)

Estos métodos, reciben como parámetro la base de datos maestra (Cliente_Maestra), la base de datos referencia (Base_referencia), las tablas T1 y T2 de ambas bases respectivamente, el nombre del atributo para el que se va a medir.

Cada método calcula la distancia entre los valores de cada celda del atributo (parámetro A) en las tablas (parámetros T1 y T2) de la base de datos (parámetros BM y B). Notar que si bien se recibe como parámetro un atributo, se toman medidas para cada celda.

Dependiendo del tipo de datos del atributo (parámetro *A*) la distancia en el método correspondiente se medirá de diferente manera:

- Tipo de datos Fecha
  Utilizaremos la función del motor de base de datos datediff;
- Tipo de datos Alfanumérico
  Contaremos con una rutina genérica para medir la distancia entre dos celdas, llamada Calcular_Distancia definida en el Anexo II;
- Tipos de datos Numérico
  Utilizaremos la funcionalidad del motor de base de datos, para calcular la diferencia entre dos números.

En resumen, se realizan las siguientes instanciaciones:

Métrica instanciada	**Objeto de la medición**	**Método**	**Parámetros método**
Desviación de correctitud semántica-Distancia entre celdas	Celdas del atributo FechaNacimiento de la tabla Particular	Calcular_Desviación_Correctitud_Semantica_Fecha	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1:Particular** **- T2: Trabajo** **- A: FechaNacimiento**
	Celdas del atributo Nombre de la tabla Persona	Calcular_Desviación_Correctitud_Semantica_Alfanumérico	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Persona** **- T2: Trabajo** **- A: Nombre**
	Celdas del atributo	Calcular_Desviación_Correctitud	**- BM:**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
	Sexo de la tabla Particular	_Semantica_Alfanumérico	**Cliente_Maestra** **- B: Base_Referencia** **- T1: Particular** **- T2: Trabajo** **- A: Sexo**
	Celdas del atributo PrimerApellido de la tabla Particular	Calcular_Desviación_Correctitud _Semantica_Alfanumérico	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Particular** **- T2: Trabajo** **- A: PrimerApellido**
	Celdas del atributo SegundoApellido de la tabla Particular	Calcular_Desviación_Correctitud _Semantica_Alfanumérico	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Particular** **- T2: Trabajo** **- A: SegundoApellido**
	Celdas del atributo PrimerNombre de la tabla Particular	Calcular_Desviación_Correctitud _Semantica_Alfanumérico	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Particular** **- T2: Trabajo** **- A: PrimerNombre**
	Celdas del atributo SegundoNombre de la tabla Particular	Calcular_Desviación_Correctitud _Semantica_Alfanumérico	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1:Particular** **- T2: Trabajo** **- A: SegundoNombre**
	Celdas del atributo RamoActividad de la tabla Persona	**Calcular_Desviación_Correctitud _Semantica_Numérico**	**- BM: Cliente_Maestra** **- B: Base_Referencia** **- T1: Persona** **- T2: Trabajo** **- A: RamoActividad**

Tabla 18- Instanciación de métricas y métodos para Desviación de Correctitud Semántica, factor Correctitud Semántica

Consideraciones:

- La Base_Referencia no tiene la totalidad de los clientes, por lo que el universo será acotado a los clientes contenidos en la misma.

5.2.3 Métrica Ratio de Correctitud Sintáctica para factor Correctitud Sintáctica

El *ratio de correctitud sintáctica* mide el porcentaje de celdas que están dentro de un rango o conjunto de valores permitidos para determinado atributo así como que estén registrados en el atributo correcto.

Tomaremos medidas de granularidad **atributo**. Se plantean dos instanciaciones ambas calculadas como agregación del *ratio de correctitud sintáctica* de las celdas de los atributos seleccionados.

1. ***Ratio Correctitud Sintáctica- Valores correctos***
2. ***Ratio Correctitud Sintáctica- Atributos correctos***

Presentamos los siguientes atributos, agrupados por tablas para las que nos interesa medir su *ratio de correctitud sintáctica-valores correctos*:

- Persona
  - Nacionalidad
  - RamoActividad
  - TipoDocumento
  - PaisDocumento
  - PaisResidencia
- Particular
  - Sexo
  - EstadoCivil
  - TipoIngreso
- Documento
  - Código
  - Estado

Presentamos los siguientes atributos que constituyen el domicilio de la persona agrupados por tablas para las que nos interesa medir su *ratio de correctitud sintáctica-atributos correctos* de acuerdo a la pregunta 9 presentada en la sección 4.4:

- Persona
  - Calle
  - NroPuerta
  - Localidad
  - Pais

Para realizar la medición, implementamos un método para cada una de las instanciaciones definidas: el método Calcular_Ratio_Correctitud_Sintáctica_Valores que verifica el dominio de cada atributo y Calcular_Ratio_Correctitud_Sintáctica_Atributos que verifica que los datos estén registrados en los atributos que corresponden. Ambos métodos calculan el *ratio de correctitud sintáctica* a nivel de atributo.

Sus cabezales son los siguientes:

Calcular_Ratio_Correctitud_Sintáctica_Valores (B: BaseDatos; T: Tabla, A: Atributo, R: Tabla)

Calcular_Ratio_Correctitud_Sintáctica_Atributos (B: BaseDatos; T: Tabla, A: Atributo, R: Tabla)

Estos métodos, reciben como parámetro la base de datos (parámetro B), tabla (parámetro T), el atributo (parámetro A) para la cual se quiere medir y la tabla donde se encuentran las reglas a controlar (parámetro R).

El método Calcular_Ratio_Correctitud_Sintáctica_Valores controla si cada celda cumple las reglas de dominio definidas para el atributo, aplicando la fórmula Ratio_Correctitud_Sintáctica_Valores= Cantidad_celdas_correctas_valores / Cantidad_celdas_total.

Contaremos con una rutina genérica Chequear_Regla_Dominio definida en el Anexo II para determinar si para cada celda la regla de dominio se cumple.

El método Calcular_Ratio_Correctitud_Sintáctica_Atributos controla que cada celda que contiene información del domicilio del cliente, tenga la información esperada para el atributo. Se aplica la fórmula Ratio_Correctitud_Sintáctica_Atributos= Cantidad_celdas_correctas_atributos / Cantidad_celdas_total.

Contaremos con una rutina genérica Chequear_Regla_Atributo definida en el Anexo II para determinar si cada celda almacena la información esperada.

En resumen, se realizan las siguientes instanciaciones para los métodos definidos:

Métrica instanciada	**Objeto de la medición**	**Método**	**Parámetros método**
Ratio Correctitud Sintáctica	Nacionalidad	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Persona - A: Nacionalidad - R: TablaReglaDominio**
	RamoActividad	Calcular_Ratio _Correctitud_Sintáctica _Valores	**- B: Cliente_Maestra - T: Persona - A: RamoActividad - R: TablaReglaDominio**
	TipoDocumento	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Persona - A: TipoDocumento - R: TablaReglaDominio**
	PaisDocumento	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Persona - A: Nacionalidad**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
			- R: TablaReglaDominio
	PaisResidencia	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Persona - A: PaisResidencia - R: TablaReglaDominio**
	Sexo	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Particular - A: Sexo - R: TablaReglaDominio**
	EstadoCivil	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Particular - A: EstadoCivil - R: TablaReglaDominio**
	TipoIngreso	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Particular - A: TipoIngreso - R: TablaReglaDominio**
	Código	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Documento - A: Código - R: TablaReglaDominio**
	Estado	Calcular_Ratio _Correctitud_Sintáctica_Valores	**- B: Cliente_Maestra - T: Documento - A: Estado - R: TablaReglaDominio**
	Calle	Calcular_Ratio _Correctitud_Sintáctica_Atributos	**- B: Cliente_Maestra - T: Persona - A: Calle - R: TablaReglaAtributo**
	NroPuerta	Calcular_Ratio _Correctitud_Sintáctica_Atributos	**- B: Cliente_Maestra - T: Persona**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
			- A: NroPuerta **- R: TablaReglaAtributo**
	Localidad	Calcular_Ratio _Correctitud_Sintáctica_Atributos	**- B: Cliente_Maestra** **- T: Persona** **- A: Localidad** **- R: TablaReglaAtributo**
	Pais	**Calcular_Ratio _Correctitud_Sintáctica_Atributos**	**- B: Cliente_Maestra** **- T: Persona** **- A: Pais** **- R: TablaReglaAtributo**

Tabla 19- Instanciación de métricas y métodos para Ratio de Correctitud Sintáctica para factor Correctitud Sintáctica

5.2.4 Métrica Desviación de Correctitud Sintáctica para factor Correctitud Sintáctica

La *desviación de correctitud sintáctica* mide la distancia sintáctica entre un dato del sistema y algún dato vecino que sea sintácticamente correcto.

Se tomarán medidas de granularidad **celda** y se plantea la siguiente instanciación:

1) Desviación de Correctitud Sintáctica

En la aplicación, hay dos atributos de la tabla Persona para los que nos interesa medir la *desviación de correctitud sintáctica:*

- País
- Localidad

Para realizar la medición, implementamos el método que calcula la *desviación de correctitud sintáctica* por celda. Su cabezal es el siguiente:

Calcular_Desviación_Correctitud_Sintactica (B: BaseDatos, T1: Tabla; A1: Atributo, T2: Tabla de código; A2: Atributo de la tabla de código)

Este método, recibe como parámetro la base de datos maestra (Cliente_Maestra), la tabla y el atributo que se quiere medir, la tabla y el atributo contra los que se compara (referencial).

El método calcula la *desviación de correctitud sintáctica* de cada celda para el atributo dado (parámetro A1) para la tabla (parámetro T1) de la base de datos (parámetro BM), comparando con las celda del atributo (parámetro A2) de la tabla de códigos (parámetro T2) de la misma base de datos, de acuerdo a la siguiente fórmula: Desviación_correctitud_sintáctica = $\min_i$(distancia (x,z_i)), donde x es la celda que se

está analizando (parámetro A1) y z_i toma el valor de cada una de las celdas del atributo (parámetro A2) para la tabla de códigos (parámetro T2) .

Contaremos con la rutina genérica Calcular_Distancia definida en el Anexo II para determinar la distancia entre dos celdas.

En resumen, se realizan las siguientes instanciaciones para el método definido:

Métrica instanciada	**Objeto de la medición**	**Método**	**Parámetros método**
Desviación de Correctitud Sintáctica	Celdas del atributo País para la tabla Persona	Calcular_Desviación _Correctitud_Sintactica	**- B: Cliente_Maestra** **- T1: Persona** **- A1 : Pais** **- T2 : Tabla_Paises** **- A2 : NombrePais**
	Celdas del atributo Localidad para la tabla Persona	**Calcular_Desviación _Correctitud_Sintactica**	**- B: Cliente_Maestra** **- T1: Persona** **- A1 : Localidad** **- T2: DatosLocalidad** **- A2: LocalidadDescripcion**

Tabla 20- Instanciación de métricas y métodos para Desviación de Correctitud Sintáctica para factor Correctitud Sintáctica

5.2.5 Métrica Granularidad para factor Precisión

La *granularidad* mide el porcentaje de atributos usados para representar un concepto simple. Una métrica simple consiste en contar los atributos válidos – no nulos – que representan el concepto.

Se toman medidas de granularidad **registro** (conjunto de celdas correspondientes a un concepto). Se plantea la siguiente instanciación:

1) Granularidad a nivel de registro

En la aplicación de contexto, hay dos conceptos diferentes para los que nos interesa medir su *granularidad:*

- Domicilio
- Nombre

El domicilio está almacenado en la tabla Persona y está formado por los siguientes atributos:

- Calle
- NroPuerta
- Localidad
- CodPostal
- País

El nombre está almacenado en la tabla Particular y está formado por los siguientes atributos:

- PrimerApellido

- PrimerNombre

Para realizar la medición, implementamos el método Calcular_Granularidad que calcula la *granularidad* de cada conjunto de atributos definido. Su cabezal es el siguiente:

Calcular_Granularidad (B: BaseDatos, T: Tabla; C: ConjuntoAtributos)

Este método, recibe la base de datos, la tabla y el conjunto de atributos que representa un concepto para el que mediremos la *granularidad*.

El método calcula la *granularidad* de cada concepto (conjunto de celdas) verificando para cada celda que integra el concepto, si es nulo o no, según la fórmula, Granularidad= Cantidad_atributos_no_nulo/Cantidad_Atributos_Concepto. El resultado es el porcentaje de celdas no nulas para el concepto.

Para cada atributo que se deba verificar, invocará la función genérica Chequear_Nulo definida en el Anexo II.

Por ejemplo, para el registro que se presenta en la Tabla 21 la granularidad es **0.60**:

Calle	Nro. Puerta	Localidad	CodPostal	País
Tacuarembó	Null	Montevideo	11100	Null

Tabla 21- Ejemplo de aplicar el método Calcular_Granularidad

En resumen, se realizan las siguientes instanciaciones para el método definido:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Granularidad a nivel de registro	Registros de la tabla Persona (área Domicilio)	Calcular_Granularidad	**- B: Cliente_Maestra - T: Persona - C: (Calle, NroPuerta, Localidad, CodPostal, País)**
	Registros de la tabla Persona (área Nombre)	**Calcular_Granularidad**	**- B: Cliente_Maestra - T: Particular - C: (PrimerApellido, PrimerNombre)**

Tabla 22 -Instanciación de métricas y métodos para Granularidad para factor Precisión

5.3 Instanciación de métricas y métodos de medición de Completitud

Para la dimensión *Completitud*, se instanciarán las métricas *ratio de densidad y ratio de cobertura*, las cuales se detallan a continuación

5.3.1 Métrica Ratio de Densidad para factor Densidad

El *ratio de densidad* mide que todos los atributos que están definidos como obligatorios – no nulos – estén con datos y que tampoco contengan valores cero o espacios (dummy). Estas reglas no están definidas en la base de datos.

Tomamos medidas con granularidad **registro**. Se plantea la siguiente instanciación:

1) Ratio de densidad

En la aplicación, nos interesa medir el *ratio de densidad* para las siguientes tablas:

- Persona
- Particular
- Cuenta
- Documentos
- EnvioCorrespondencia

Para realizar la medición implementamos el método Calcular_Ratio_Densidad que controla si una celda tiene datos no nulos y no dummy y luego calcula el ratio a nivel de registro. Su cabezal es el siguiente:

Calcular_Ratio_Densidad (B: Base de datos; T: Tabla, C: ConjuntoAtributos)

Este método, recibe como parámetro la base de datos, la tabla y el conjunto de atributos que no deben ser nulos.

El método calcula el *Ratio de densidad* de cada registro contando la cantidad de celdas en nulo o dummy en el registro. Luego aplicará la siguiente fórmula, Ratio Densidad=Cantidad_celdas_no_nulas_o_dummy/Cantidad_celdas_obligatorias.

Para cada atributo obligatorio invocará la función Chequear_Nulo y cuando el atributo no es nulo, invocará a las funciones Chequear_Blanco o Chequear_Cero dependiendo del tipo de dato, para determinar si es un valor dummy. Estas funciones están definidas en el Anexo II.

La siguiente tabla resume las instanciaciones del método:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de densidad	Persona	Calcular_Ratio _Densidad	**- B : Base_Maestra - T: Persona - C : (Nombre, Domicilio, Localidad, Pais)**
	Particular	Calcular_Ratio _Densidad	**- B : Base_Maestra - T: Particular - C : (PrimerApellido, PrimerNombre, Sexo, FechaNacimiento)**
	Cuenta	Calcular_Ratio _Densidad	**- B : Base_Maestra - T: Cuenta**

Métrica instanciada	Objeto de la medición	Método	Parámetros método
			- C : (Identificacion, Sucursal)
	Documentos	Calcular_Ratio _Densidad	**- B : Base_Maestra** **- T: Documentos** **- C : (FechaPresentacion, NroDocumento)**
	EnvioCorrespondencia	**Calcular_Ratio _Densidad**	**- B : Base_Maestra** **- T: EnvioCorrespondencia** **- C : (Direccion, Localidad, Pais)**

Tabla 23- Instanciación de métricas y métodos para Ratio de Densidad para factor Densidad

5.3.2 Métrica Ratio de Cobertura para factor Cobertura

El *ratio de cobertura* en un ambiente de replicación sirve para verificar que todos los individuos que estén en la base de datos del nodo madre, estén también en las bases de datos replicadas.

Tomamos medidas con granularidad **tabla**. Se plantea la siguiente instanciación:

1) Ratio de Cobertura

En la aplicación nos interesa medir el *ratio de cobertura* para las siguientes tablas de datos que son replicadas mediante la Herramienta de Integración:

- Persona
- Particular
- Empresa
- Cuenta
- EnvioCorrespondencia
- Intregracion_Cliente
- GruposEconomicos

Para realizar la medición, implementamos el método Calcular_Ratio_Cobertura que calcula el *ratio de cobertura* de cada registro y luego realiza la agregación. Su cabezal es el siguiente:

Calcular_Ratio_Cobertura (BM: BaseDatos; B: BaseDatos; T1: Tabla; T2: Tabla)

Este método, recibe como parámetro las bases de datos del nodo madre y una réplica (parámetros BM y B respectivamente) y las tablas para la cuales se quiere realizar la medición (T1 y T2 respectivamente).

El método calcula el *ratio de cobertura* de cada registro buscando en la tabla de la réplica (parámetro T2), cada uno de los registros que existen en la tabla del nodo madre (parámetro T1). Se devuelve un booleano, según si se encuentra o no el registro en la base de datos replicada. El acceso a las tablas es por la clave primaria, la cual será obtenida de la metadata de las tablas.

Luego se aplica la siguiente fórmula: RatioCobertura= Cantidad_registros_encontrados_en_réplica/ Cantidad_registros_tabla_nodo_madre.

La siguiente tabla resume las instanciaciones:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de Cobertura	Persona	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2: Persona**
	Particular	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2: Particular**
	Empresa	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2: Empresa**
	Cuenta	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2 : Cuenta**
	EnvioCorrespondencia	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2: EnvioCorrespondencia**
	Intregracion_Cliente	Calcular_Ratio _Cobertura	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2: Intregracion_Cliente**
	GruposEconomicos	**Calcular_Ratio _Cobertura**	**- BM: Base_Maestra - B: Base_Cliente$_{01}$ - T1, T2:: GruposEconomicos**

Tabla 24- Instanciación de métricas y métodos para Ratio de Cobertura para factor Cobertura

Se realizaran estas instanciaciones para todas las réplicas consideradas para esta tesis: Base_Cliente$_{01}$, Base_Cliente$_{02}$ y Base_Cliente$_{03}$.

Consideraciones:
Una mejora para este método y para un posible trabajo de data cleaning, puede ser obtener una medida con granularidad registro, identificando los registros no replicados en las bases suscriptoras; pero a los efectos de responder a la pregunta planteada, sólo calculamos el porcentaje de cobertura a nivel de archivo.

5.4 Instanciación de métricas y métodos de medición de Trazabilidad

Para la dimensión *Trazabilidad* se instancia la métrica *ratio de verificabilidad* la cual se detalla a continuación.

5.4.1 Métrica Ratio de Verificabilidad para factor Verificabilidad

El *ratio de verificabilidad* mide que la información esté bien documentada, sea verificable y fácilmente atribuible a una fuente de datos. En la aplicación, nos interesa

medir el *ratio de verificabilidad* para conocer cuál fue el usuario que realizó el alta o actualización del registro, en una fecha determinada.

Sólo consideraremos los registros modificados o dados de alta después de la implantación de la Herramienta de Integración, por lo que obtendremos medidas con granularidad **área** (todos los registros cuya fecha de modificación sea mayor a una determinada fecha).

Se plantea la siguiente instanciación:

1) Ratio de verificabilidad

La tabla sobre la cual mediremos el *ratio de verificabilidad* es la siguiente:

- Persona

Se utilizarán los atributos *Identificación*, que identifica al cliente y *FecUltMod* que identifica la fecha última modificación del registro. Ambos atributos están en la tabla Persona.

La información de las trazas está disponible en la base de datos Trazas a partir de la fecha que se implantó la Herramienta de Integración, por lo que utilizaremos dicha fecha como criterio de partición. Realizaremos la medición sólo para los registros posteriores a esa fecha (para los anteriores no contamos con trazabilidad disponible en el ambiente seleccionado para esta tesis).

Para realizar la medición, implementamos el método Calcular_Verificabilidad busca para cada registro si puede determinar su *trazabilidad* y luego calcula el ratio para toda el área. La identificación y condición del área será registrada en una tabla de parámetros de la base de datos. Su cabezal es el siguiente:

Calcular_Verificabilidad (BM: BaseDatos; B: BaseDatos; T1: Tabla; T2: Tabla; A: Condición)

Este método, recibe como parámetro las bases de datos Cliente_Maestra y Trazas (parámetros BM y B respectivamente), la tabla para la cual se quiere medir (parámetro T1), la tabla que contiene las trazas (parámetro T2) y la condición (parámetro A) que determinará el área en que se medirá el *ratio de trazabilidad*. Aplica la función de cálculo Ratio Verificabilidad = Cantidad_registros_con_trazas/ Cantidad_registros_área.

Contaremos con una rutina genérica Analizar_trazas definida en el Anexo II para determinar si para un registro se pueden identificar las trazas.

En la siguiente tabla se presenta la instanciación que tendrá el método:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de verificabilidad	**Área de la tabla Persona**	**Calcular _Verificabilidad**	**- BM: Cliente_Maestra - B: Trazas - T1: Persona - T2: TrazaMovimiento - A: FecUltMod>"Fecha"**

Tabla 25- Instanciación de métricas y métodos para Ratio de Verificabilidad para factor Verificabilidad

Consideraciones:

- El método a ser invocado utiliza la información registrada por la Herramienta de Integración, por lo que sólo podemos medir el *ratio de verificabilidad* a partir de la fecha de implantación de dicha herramienta. Este criterio será considerado en la definición del área para el cual se realizará la medición.

5.5 Instanciación de métricas y métodos de medición de Integridad

Para la dimensión *Integridad* se instancian las métricas *ratio de integridad referencial* y *ratio de integridad de dominio* las cuales se detallan a continuación.

5.5.1 Métrica Ratio de Integridad Referencial para factor Integridad Referencial

El *ratio de integridad referencial* mide el porcentaje de registros que satisfacen restricciones de integridad referencial.

Para determinar el *ratio de integridad referencial* de una tabla, contaremos la cantidad de registros que satisfacen las restricciones, obteniendo medidas con granularidad **tabla.** Se plantea la siguiente instanciación:

1) Ratio de integridad referencial

En la aplicación existen restricciones referenciales que sólo se cumplen bajo determinadas condiciones (por ejemplo si un cliente es una empresa). Por tal motivo, no fueron declaradas como restricciones de la base de datos, por lo que nos interesa medir su satisfacción.

Específicamente dependiendo del atributo *TipoCliente* de la tabla Persona, el cliente debe referenciar a una de las siguientes tablas:

- Persona-Particular
- Persona-Empresa
- Persona-Integracion_Cuenta

Las reglas a controlar son las siguientes:

1. Si TipoCliente=’P’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Particular;
2. Si TipoCliente=’E’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Empresa;
3. Si TipoCliente=’C’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Integracion_Cuenta.

Para realizar la medición, implementamos el método Calcular_Ratio_Integridad_Referencial que calcula el *ratio de integridad referencial* de cada tabla. Su cabezal es el siguiente:

Calcular_Ratio_Integridad_Referencial (B: Base de datos; T1: TablaMadre; T2: TablaReferenciada; C: Condición)

Este método, recibe la base de datos (parámetro B) y las tablas madre y referenciada (parámetros T1 y T2) y la condición (parámetro C) por la cual debe hacer el control de integridad.

El método calcula el *ratio de integridad referencial* de acuerdo a la siguiente fórmula: RatioIntegridadReferencial= Cantidad_registros_cumplen_control_integridad / Cantidad_registros_tabla_ para_condición_dada).

Para cada registro de la tabla madre (parámetro T1) que cumple con la condición (parámetro C) controla si pertenece a la tabla referenciada (parámetro T2). El acceso se realizará por clave primaria lo cual se obtendrá de la metadata de la base de datos.

En la siguiente tabla se presenta las instanciaciones que tendrá el método:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de integridad referencial	Persona-Particular	Calcular_Ratio _Integridad _Referencial	- B : Cliente_Maestra - TM: Persona - TR: Particular - C: TipoCliente=’P’
	Persona-Empresa	Calcular_Ratio _Integridad _Referencial	- B : Cliente_Maestra - TM: Persona - TR: Empresa - C: TipoCliente=’E’
	Persona-Integracion _cuenta	Calcular_Ratio _Integridad _Referencial	- B : Cliente_Maestra - TM: Persona - TR: Integracion_Cuenta - C: TipoCliente=’C’

Tabla 26- Instanciación de métricas y métodos para Ratio de Integridad Referencial para factor Integridad Referencial

5.5.2 Métrica Ratio de Integridad de Dominio para factor Integridad de Dominio

El *ratio de integridad de dominio* mide la cantidad de registros que cumplen con las reglas de validación.

Para determinar el *ratio de integridad referencial* de un atributo, contaremos la cantidad de celdas que satisfacen las restricciones, obteniendo medidas con granularidad **atributo.** Se plantea la siguiente instanciación:

1) Ratio de integridad de dominio

En la aplicación, existen los siguientes atributos - agrupados por tablas - para las que nos interesa medir el *ratio de integridad de dominio:*

- Particular
  - Actividad
  - TipoIngreso
- Persona
  - CanalCte
  - TipoDocumento
  - TipoEmpresa
  - TipoCliente

Para realizar la medición, implementamos el método Calcular_Ratio_Integridad_Dominio el cual determina si un atributo cumple o no con las reglas de validación de cada atributo. Su cabezal es el siguiente:

Calcular_Ratio_Integridad_Dominio (B: BaseDatos; T: Tabla, A: Atributo, R: Tabla)

Este método recibe como parámetro la base de datos (parámetro B), la tabla (parámetro T) y el atributo (parámetro A) que queremos medir. Contaremos con una tabla (parámetro R) que contiene las restricciones a controlar.

El método controla si cada celda cumple con la restricción dada. Luego aplicará la fórmula RatioIntegridadDominio= Cantidad_Celdas_Cumplen_con_control_integridad / Cantidad_celdas_controladas.

Contaremos con una rutina genérica Chequear_Restricciones definida en el Anexo II para determinar si una celda cumple o no con la constraint establecida.

En resumen, se realizan las siguientes instanciaciones:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Ratio de integridad de dominio	Actividad	Calcular_Ratio _Integridad_Dominio	**- B : Cliente_Maestra - T: Particular - A: Actividad - R: TablaRestricción**
	TipoIngreso	Calcular_Ratio _Integridad_Dominio	**- B : Cliente_Maestra - T: Particular - A: TipoIngreso - F: Average - R: TablaRestricción**
	CanalCte	Calcular_Ratio _Integridad_Dominio	**- B : Cliente_Maestra - T: Persona - A: CanalCte - F: Average - R: TablaRestricción**
	TipoDocumento	Calcular_Ratio _Integridad_Dominio	**- B : Cliente_Maestra - T: Persona - A: TipoDocumento - F: Average - R: TablaRestricción**
	TipoEmpresa	Calcular_Ratio _Integridad_Dominio	**- B : Cliente_Maestra - T: Persona - A: TipoEmpresa - F: Average - R: TablaRestricción**
	TipoCliente	**Calcular_Ratio _Integridad_Dominio**	**- B : Cliente_Maestra - T: Persona - A: TipoCliente - F: Average - R: TablaRestricción**

Tabla 27- Instanciación de métricas y métodos para Ratio de Integridad de Dominio para factor Integridad de dominio

5.6 Instanciación de métricas y métodos de medición de Unicidad

Para la dimensión *Unicidad* se instancia la métrica *similaridad* la cual se detalla a continuación.

5.6.1 Métrica Similaridad para factor Unicidad

La *similaridad* mide la distancia al registro más cercano, que puede corresponder al mismo cliente, aunque tengan la clave primaria diferente.

Tomaremos medidas de granularidad **celda.** Se plantea la siguiente instanciación:
1) Similaridad

En la aplicación hay dos atributos de la tabla Persona para la que nos interesa medir la *similaridad*:

- Nombre
- Identificación

Para realizar la medición, implementamos el método Calcular_Similaridad que calcula la *similaridad* de cada celda. Su cabezal es el siguiente:

Calcular_Similaridad (B: BaseDatos; T: Tabla; A: Atributo)

Este método, recibe como parámetro la base de datos (parámetro B), la tabla (parámetro T) y el atributo (parámetro A) para los cuales se quiere medir.

El método calcula la *similaridad* de cada celda para el atributo dado (parámetro A de la tabla T) según la fórmula, Similaridad= $\min_i$(distancia (x,z_i)) donde x es la celda que se está analizando (parámetro A) y z_i toma el valor de cada una de las celdas del resto de la tabla para el mismo atributo (parámetro A).

Para calcular la distancia entre las celdas, utilizaremos la función Calcular_Distancia definida en el Anexo II.

En la siguiente tabla se presenta las instanciaciones que tendrá el método:

Métrica instanciada	Objeto de la medición	Método	Parámetros método
Similaridad	Atributo Nombre de la tabla Persona	Calcular_Similaridad	**- B : Cliente_Maestra - T: Persona - A: Nombre**
	Atributo Identificación de la tabla Persona	**Calcular_Similaridad**	**- B : Cliente_Maestra - T: Persona - A: Identificación**

Tabla 28- Instanciación de métricas y métodos para Similaridad para factor Unicidad

Consideraciones:

- Con los resultados obtenidos, se podrá calcular la Tasa de Similaridad, definiendo para cada atributo un umbral α a partir del cual considero dos valores

como similares. Por ejemplo para el caso de identificación, al ser la clave primaria de la tabla no van a existir resultados con similaridad 0 por lo que se puede definir el umbral α=1.

5.7 Síntesis

Hemos presentado las métricas de calidad instanciadas para las métricas seleccionadas en la sección 4.5. Para cada una de estas métricas definimos su granularidad, los objetos del sistema de información con los cuales se instanciarían y los métodos asociados. En algunos casos, para una misma métrica, encontramos interesante instanciarla más de una vez, como lo hicimos por ejemplo con la métrica *edad*. Asimismo para algunas métricas, definimos más de un método, como por ejemplo para la métrica *desviación de correctitud semántica*. En todos los casos, para hacer los métodos más genéricos definimos como parámetro la base de datos donde se realizará la medición, por lo que es importante notar que los mismos pueden ser utilizados en otras aplicaciones. Definimos rutinas genéricas - las cuales son detalladas en el Anexo II - que son utilizadas en varios métodos; por ejemplo la función que verifica si una celda tiene valor nulo.

En resumen, instanciamos cada una de las métricas seleccionadas, obteniendo las diferentes granularidades definidas para esta tesis: área, registro, tabla, celda y atributo. A partir de estas definiciones, se realizará la implementación de los métodos, lo cual se detalla en la sub-sección siguiente.

A continuación se presenta una síntesis de las métricas instanciadas para cada factor y métrica seleccionada.

Factor de Calidad	Métrica	Métrica instanciada
Edad	Edad	**Edad-Promedio de tiempo en días**
		Edad-Máximo tiempo en días
Actualidad	Actualidad	**Actualidad-Promedio de tiempo en días**
		Actualidad-Máximo tiempo en días
	Ratio de frescura	**Ratio de Frescura**
Correctitud Semántica	Ratio de correctitud semántica	**Ratio de correctitud semántica Porcentaje de clientes correctos de la muestra**
	Desviación de correctitud semántica	**Desviación de correctitud semántica- Distancia entre celdas**
Correctitud Sintáctica	Ratio de correctitud sintáctica	**Ratio correctitud sintáctica-valores correctos**
		Ratio correctitud sintáctica-atributos correctos
	Desviación de correctitud Sintáctica	**Desviación de Correctitud Sintáctica**
Precisión	Granularidad	**Granularidad a nivel de**

Factor de Calidad	Métrica	Métrica instanciada
		registro
Densidad	Ratio de densidad	**Ratio de densidad**
Cobertura	Ratio de cobertura	**Ratio de Cobertura**
Verificabilidad	Ratio de verificabilidad	**Ratio de verificabilidad**
Integridad Referencial	Ratio de integridad referencial	**Ratio de integridad referencial**
Integridad de dominio	Ratio de integridad de dominio	**Ratio de integridad de dominio**
Unicidad	**Similaridad**	**Similaridad**

Tabla 29- Cuadro resumen con factores, métricas, métricas instanciadas

En la Tabla 30 se presenta una síntesis de los métodos definidos y sus parámetros para cada una de las métricas instanciadas.

Métrica instanciada	Método	Parámetros método
Edad-Promedio de tiempo en días	Calcular_Edad	- **BM: BaseDatos** - **T: Tabla** - **A: Atributo** - **F: Función**
Edad-Máximo tiempo en días	Calcular_Edad	- **BM: BaseDatos** - **T: Tabla** - **A: Atributo** - **F: Función**
Actualidad-Promedio de tiempo en días	Calcular_Actualidad	- **BM: BaseDatos** - **B: BaseDatos** - **T: Tabla** - **A: Atributo** - **F: Función**
Actualidad-Máximo tiempo en días	Calcular_Actualidad	- **BM: BaseDatos** - **B: BaseDatos** - **T: Tabla** - **A: Atributo** - **F: Función**
Ratio de Frescura	Calcular_Ratio_Frescura	- **BM: BaseDatos** - **B: BaseDatos** - **T1: Tabla** - **T2: Tabla** - **F: Función**
Ratio de correctitud semántica Porcentaje de clientes correctos de la muestra	Calcular_Ratio_Correctitud _Semantica	- **BM: BaseDatos** - **B: Base de datos** - **T1: Tabla** - **T2: Tabla** - **F:Función**
Desviación de correctitud semántica-	Calcular_Desviación _Correctitud_Semantica _Alfanumérico	- **BM: BaseDatos** - **B: BaseDatos** - **T1: Tabla**

Métrica instanciada	Método	Parámetros método
Distancia entre celdas		- **T2: Tabla** - **A : Atributo** - **F: Función**
	Calcular_Desviación_ _Correctitud_Semantica _Numérico	- **BM: BaseDatos** - **B: BaseDatos** - **T1: Tabla** - **T2: Tabla** - **A : Atributo**
	Calcular_Desviación _Correctitud_Semantic_Fecha	- **BM: BaseDatos** - **B: BaseDatos** - **T1: Tabla** - **T2: Tabla** - **A : Atributo**
Ratio correctitud sintáctica- valores correctos	Calcular_Ratio _Correctitud_Sintáctica _Valores	- **B: BaseDatos** - **T:Tabla** - **A:Atributo** - **R: TablaReglaDominio**
Ratio correctitud sintáctica- atributos correctos	Calcular_Ratio _Correctitud_Sintáctica _Atributos	- **B: BaseDatos** - **T:Tabla** - **A:Atributo** - **R: TablaReglaAtributo**
Desviación de Correctitud Sintáctica- Promedio de distancia sintáctica	Calcular_Desviación _Correctitud_Sintactica	- **B: BaseDatos** - **T1: Tabla** - **A1 : Atributo** - **T2 : Tabla** - **A2 : Atributo**
Granularidad a nivel de registro	Calcular_Granularidad	- **B: BaseDatos** - **T: Tabla** - **C : ConjuntoAtributos**
Ratio de densidad	Calcular_Ratio_Densidad	- **B : BaseDatos** - **T: Tabla** - **C: ConjuntoAtributos**
Ratio de Cobertura	Calcular_Ratio_Cobertura	- **BM : BaseDatos** - **B: BaseDatos** - **T1: Tabla** - **T2: Tabla**
Ratio de verificabilidad	Calcular_Verificabilidad	- **BM : BaseDatos** - **B: BaseDatos** - **T1: Tabla** - **T2: Tabla** - **A: Condición**
Ratio de integridad referencial	Calcular_Ratio_Integridad _Referencial	- **B : BaseDatos** - **TM: Tabla** - **TR: Tabla** - **C : Condición**
Ratio de integridad	Calcular_Ratio_Integridad	- **B : BaseDatos**

Métrica instanciada	Método	Parámetros método
de dominio	_Dominio	- **T: Tabla** - **A: Atributo** - **F: Función** - **R: Tabla**
Similaridad	**Calcular_Similaridad**	- **B : BaseDatos** - **T: Tabla** - **A: Atributo**

Tabla 30- Cuadro resumen de métodos y parámetros para cada métrica instanciada

6. Experimentación y resultados

En este capítulo describimos la aplicación que se desarrolló, incluyendo la arquitectura y la definición de las tablas y presentamos los resultados obtenidos de la medición realizada.

Para simplificar y agilizar el trabajo de medición, se dispuso de un servidor de base de datos utilizando Ms SqlServer 2005, donde se crearon los siguientes ambientes de trabajo:

1. Un ambiente similar al de la instalación real. Se copiaron las bases de datos Cliente_Maestra, tres bases de datos suscriptoras ($Base\_Cliente_{01}$, $Base\_Cliente_{02}$, $Base\_Cliente_{03}$) y las bases de datos Base_Referencia y Trazas;

2. Un ambiente de trabajo para realizar las mediciones, donde se almacenó el metamodelo de calidad descrito en el capítulo 4.

Para la implementación de la propuesta se utilizó la herramienta ´Qbox-Foundation´ [Etcheverry+2008], la cual provee una interfaz gráfica para definir objetivos y preguntas de calidad, y permite asociarlos a factores y métricas de calidad. La herramienta utiliza un catálogo de conceptos de calidad que almacena definiciones de factores, métricas y métodos de medición para ser utilizados y reutilizados por diversas aplicaciones. 'Qbox-Foundation' provee los mecanismos para ejecutar los métodos instanciados y almacenar las medidas obtenidas.

Debemos resaltar que si bien 'Qbox-Foundation' provee la infraestructura para definir los conceptos de calidad, a la hora de utilizarla, el catálogo estaba prácticamente vacío. Esto se debe a que la herramienta sólo se había utilizado con pequeños casos de estudio orientados a testear la correctitud y robustez de la programación, pero no había sido utilizada para medir la calidad en aplicaciones reales.

La primera tarea de implementación consistió entonces en proveer un catálogo inicial de conceptos de calidad para 'Qbox-Foundation', cargando las dimensiones de calidad utilizadas en esta tesis, así como sus factores, métricas y métodos de medición. Para ello se implementaron los métodos propuestos así como los mecanismos de instanciación apropiados para cada uno y las rutinas auxiliares usadas para éstos. La implementación fue realizada en lenguaje SQL. Dichos métodos serán invocados por la herramienta ´Qbox-Foundation´ a través de una interfaz JAVA. Una vez definido el catálogo, pudimos definir los objetivos y preguntas de calidad de nuestra aplicación e instanciar las métricas y métodos tal cual fue detallado en el capítulo anterior.

Para cargar los objetos del sistema de información utilizado en esta tesis, se utilizó la metadata (catálogo) de las diferentes bases de datos utilizadas de la aplicación de contexto y se extrajo la lista de tablas y atributos que participarían en los métodos implementados. La identificación de celdas y registros se obtienen de las tablas de datos registradas en la base de datos. Para el particionamiento de tablas se utilizó la fecha de implantación de la nueva herramienta llamada *Herramienta de Integración*. Con este criterio definimos dos áreas de la tabla Personas las cuales fueron tomadas en cuenta para realizar las mediciones con granularidad área. Una vez preparados los ambientes

necesarios, se ejecutaron los métodos, obteniéndose medidas de calidad correspondientes

En la siguiente sub-sección describimos la implementación de todos los objetos necesarios para realizar las mediciones deseadas.

6.1 Descripción de la implementación

El mecanismo de medición implementado consta de tres partes: i) métodos de medición, implementados en lenguaje SQL mediante funciones y procedimientos almacenados; ii) rutinas genéricas también implementadas en lenguaje SQL mediante funciones y procedimientos almacenados; iii) metadatos que dan soporte a la medición, almacenados en una base de datos relacional.

Para cumplir con nuestro objetivo de que los métodos de medición sean paramétricos, se implementaron utilizando SQL Dinámico, para lo cual se tomaron algunas decisiones de implementación las cuales son presentadas en la sub-sección 6.3.

En la siguiente sub-sección se presenta el esquema relacional diseñado para almacenar los objetos instanciados y almacenar las medidas obtenidas.

6.1.1 Esquema relacional del metamodelo

En la Figura 5 se presenta el esquema relacional utilizado para representar los objetos instanciados y almacenar las medidas obtenidas. Dicho esquema es la implementación del metamodelo conceptual presentado en la sección 4.2 el cual está basado en el modelo de la herramienta ´**Qbox-Foundation**´. En [Chiruzzo+2007] se presenta otra implementación del metamodelo conceptual de ´Qbox-Foundation´ el cual tiene algunas diferencias con el nuestro.
No utilizamos directamente el esquema relacional planteado en [Chiruzzo+2007] por los siguientes motivos: primeramente, esta tesis se desarrolló en paralelo a la primera implementación de la herramienta ´Qbox-Foundation´ por lo que tuvimos que definir las tablas a utilizar antes de poder contar con la primera liberación de la herramienta. Además, nuestra propuesta de rutinas reutilizables y nuestro mecanismo de parametrización de los métodos requirieron la implementación de estructuras adicionales que no estaban previstas en el metamodelo inicial de ´Qbox-Foundation´. Finalmente, la propuesta [Chiruzzo+2007] implementó sólo las funcionalidades de ´Qbox-Foundation´ necesarias para manipular un caso de estudio en el dominio biológico, mientras que otras funcionalidades como el almacenamiento de medidas de calidad de granularidad celda o atributo no fueron implementadas. Es más, su versión sólo manipulaba medidas de granularidad fuente o experimento (conjunto de tablas) por lo que tuvimos que brindar un diseño completo del bloque 4 del metamodelo de la sección 4.2 para soportar las granularidades de celda, atributo, tupla, área y tabla.

A continuación describimos el desarrollo necesario para cada bloque:

- Para el bloque *1- Abstracciones de calidad* se definen tablas para almacenar cada una de las abstracciones de calidad. Nuestros aportes para este bloque son las rutinas que son invocadas por los métodos y los parámetros tanto de métodos como de rutinas;

- Para el bloque 2- *Objetivos de calidad* se definen tablas para almacenar los objetivos y preguntas de calidad, así como los factores, métricas y métodos

instanciados. Definimos también una tabla para almacenar los parámetros con los cuales son instanciados los métodos;

- El bloque *3- Objetos medibles* no fue implementado físicamente sino que se utilizan los metadatos de la base de datos (para obtener la lista de tablas y atributos) y las instanciaciones de dichas tablas (para obtener la lista de tuplas y celdas de las mismas). Las áreas se almacenarán en una tabla de parámetros, donde se registrará la identificación del área y la condición de cómo se determina;

- Para el bloque *4- Medidas de calidad* se creó una tabla para almacenar las medidas correspondientes a cada granularidad (tabla, área, tupla, atributo, celda). Realizamos esta separación porque los objetos a los cuales se asocian los resultados de las mediciones (tablas, celdas, tuplas, atributos y áreas) se identifican de forma diferente. Notar que como no implementamos físicamente las colecciones de objetos del sistema de información – correspondientes al bloque 3 del metamodelo de calidad de la sección 4.2 - los atributos correspondientes a la identificación de tablas, celdas, tuplas, atributos y áreas no están representados como claves foráneas en el bloque 4.

En la Figura 5 se representan en color blanco las tablas que surgieron de un mapeo directo de los conceptos del metamodelo conceptual de ´Qbox-Foundation´ y que coinciden con las tablas propuestas en [Chiruzzo+2007]. Se representan sombreadas las tablas que surgen de los cambios propuestos en esta tesis. La notación del esquema de la Figura 5 es presentada en el Anexo IV.

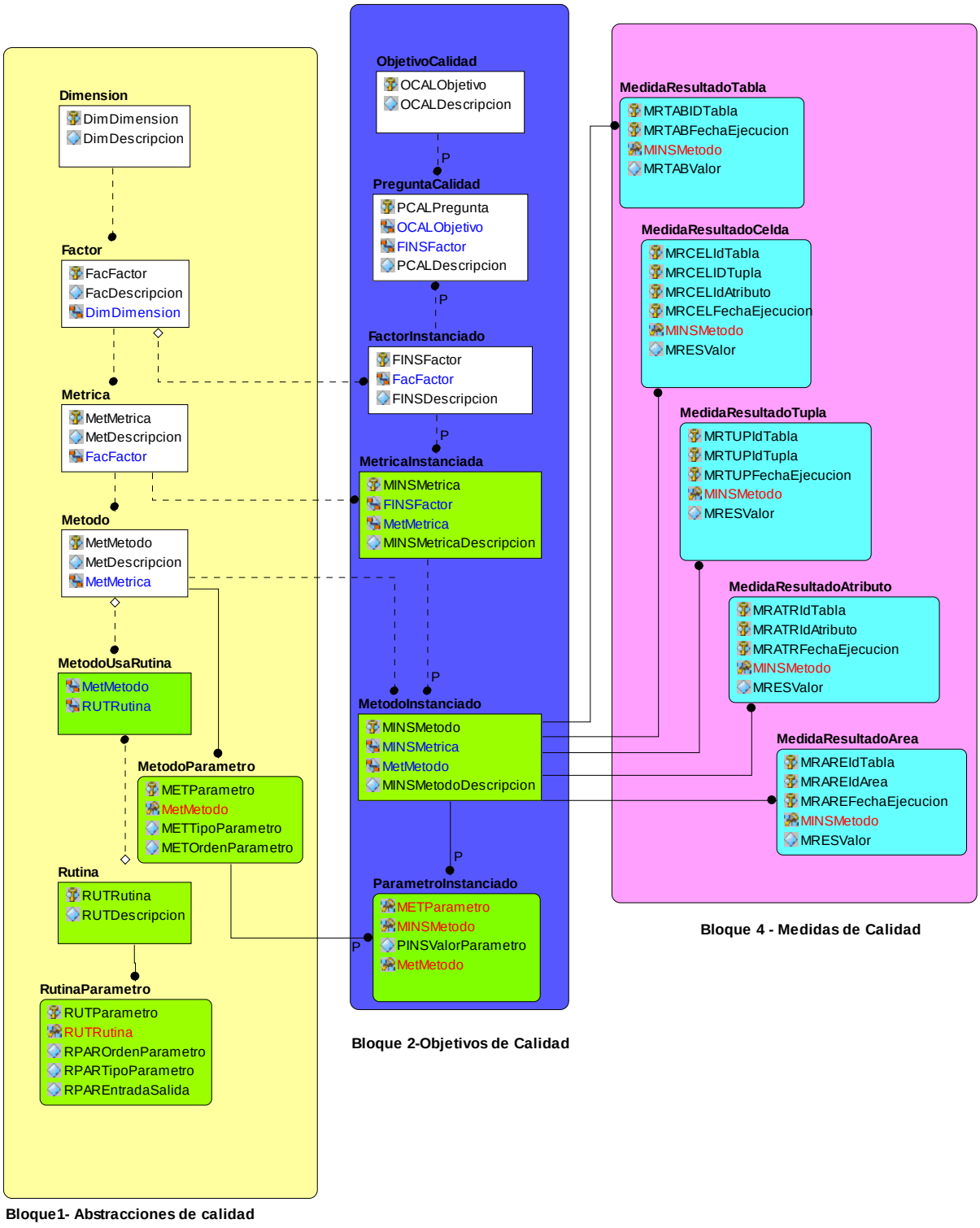

Figura 5- Esquema relacional del metamodelo

Los siguientes cuadros describen las tablas que componen cada bloque. En el Anexo III se extiende la descripción de las tablas sombreadas del modelo, incluyendo la lista de atributos de cada una.

Tabla	Descripción
Dimension	Almacena la identificación y descripción de las dimensiones de calidad de alto nivel.
Factor	Almacena la identificación y descripción de los factores de calidad de alto nivel.
Metrica	Almacena la identificación y descripción de las métricas de calidad de alto nivel.
Metodo	Almacena la identificación y descripción de los métodos de medición de alto nivel.
MetodoParametro	Registra la colección de parámetros que recibe el método, el tipo y el orden en que debe recibirlos.
Rutina	Registra la colección de rutinas genéricas que pueden ser invocadas desde un método. Las rutinas aquí almacenadas son las que se incluyen en el Anexo II.
RutinaParametro	Registra la colección de parámetros que recibirá la rutina, el orden y el tipo, y si es de entrada o de salida. Esta tabla es dependiente de la tabla Rutina, por lo que hereda de ésta la clave primaria. Se agrega el atributo parámetro a la clave primaria.
MetodoUsaRutina	Registra las rutinas utilizadas en la implementación de cada método.

Tabla 31- Tablas del bloque 1 – Abstracciones de calidad

Tabla	Descripción
ObjetivoCalidad	Registra la identificación y descripción de los objetivos de calidad seleccionados.
PreguntaCalidad	Registra la identificación y descripción de las preguntas de calidad realizadas.
FactorInstanciado	Registra todos los factores a ser instanciados guardando su factor de alto nivel, identificación y descripción.
MetricaInstanciada	Registra todas las métricas a ser instanciadas guardando su métrica genérica, identificación, descripción y factor instanciado.
MetodoInstanciado	Registra todos los métodos a ser instanciados, su método genérico, identificación, descripción y la métrica instanciada de la cual va a realizar la medición.
ParametroInstanciado	Registra el valor de cada uno de los parámetros utilizados para instanciar los métodos.

Tabla 32- Tablas del bloque 2- Objetivos de calidad

Tabla	Descripción
MedidaResultadoTabla	Registra el valor de las mediciones de granularidad tabla. La clave primaria está compuesta por: identificación del método instanciado + identificación de la tabla + la fecha de ejecución.
MedidaResultadoArea	Registra el valor de las mediciones de granularidad área. La clave primaria está compuesta por: identificación del método instanciado + identificación de la tabla + identificación del área + la fecha de ejecución.
MedidaResultadoAtributo	Registra el valor de las mediciones de granularidad atributo. La clave primaria está compuesta por: identificación del método

Tabla	Descripción
	instanciado + identificación de la tabla + identificación del atributo + la fecha de ejecución.
MedidaResultadoTupla	Registra el valor de las mediciones de granularidad tupla. La clave primaria está compuesta por: identificación del método instanciado + identificación de la tabla + identificación de la tupla + la fecha de ejecución.
MedidaResultadoCelda	Registra el valor de las mediciones de granularidad celda. La clave primaria está compuesta por: identificación del método instanciado + identificación de la tabla + identificación del atributo + identificación de la tupla + la fecha de ejecución.

Tabla 33- Tablas del bloque 4- Medidas de calidad

En la siguiente sub-sección se presenta una descripción de cómo se implementaron los métodos.

6.1.2 Implementación de los métodos

La implementación se hizo en tres niveles: i) funciones para la inserción en las tablas del modelo; ii) construcción de los métodos; iii) funciones para instanciar los métodos. Se detalla a continuación cada uno de los niveles.

1. ***Funciones para la inserción en las tablas del modelo***
   Para estandarizar el criterio de inserción en la base de datos, se implementaron funciones para la inserción en las tablas del modelo. Éstas reciben como parámetros todos los valores a ser insertados en las tablas y devuelven un estado que indica si la operación fue exitosa o no. Sólo se implementaron para las tablas que están sombreadas, ya que para el resto se asume que serán cargadas por la herramienta ´Qbox-Foundation´. Estas tablas son cargadas en dos etapas: i) previo a la instanciación de los métodos, las tablas MetodoUsaRutina, MetodoParametro, Rutina, RutinaParametro, MetricaInstanciada y MetodoInstanciado fueron cargadas con todos los elementos definidos en el capítulo 5; ii) el resto de las tablas que están sombreadas - ParametroInstanciado, MedidaResultadoTabla, MedidaResultadoCelda, MedidaResultadoTupla, MedidaResultadoAtributo, MedidaResultadoArea – son cargadas cuando corresponde en la instanciación de los métodos.

2. ***Métodos***
   Los métodos reciben los parámetros de la instanciación, y mediante SQL Dinámico, realizan la medición. Según la granularidad, hay dos tipos de métodos:

- *Método con granularidad tabla, atributo o área*, recibe los parámetros, realiza el cálculo por registro, luego realiza la agregación e inserta los resultados obtenidos en la tabla correspondiente según la granularidad. Se presenta un pseudocódigo de ejemplo:

```
/*
Recibe los parámetros a ser instanciados: la base de datos, la
tabla, el atributo y  la función de agregación a ser aplicada.
*/
```

```
Metodo_Calcular_Edad ( @B: BaseDatos, @T: Tabla, @A: Atributo,
@F: FuncionAgregacion): @edad
/*
Arma la sentencia de SQL Dinámico con los parámetros recibidos
para realizar el cálculo. Calcula la medida (ej. edad) aplicando
la fórmula EdadRegistro=(Now-@A) para cada registro, descartando
los registros cuya fecha (parametro @A) está en nulo. Luego
aplica la funcion de agregación (parámetro @F). Asigna al
parámetro de salida (@edad) el resultado obtenido.
*/
@Edad= 'FuncionAgregacion' (select datediff(day, datetime()
,'Atributo',) from 'Tabla' where
Chequear_Nulo('BaseDatos','Tabla','Atributo','IdTupla')=False)
/*La inserción en la tabla definitiva se hará en la función que
invoca el método, llamando a la rutina que insertará en
MedidaResultadoTabla*/
```

- *Método con granularidad celda o tupla* recibe los parámetros, realiza el cálculo por celda o tupla el cual guarda en una tabla temporal. Estos resultados, luego serán insertados en la tabla de resultados, según la granularidad, utilizando las funciones implementadas a tales efectos. Se presenta un pseudocódigo de ejemplo:

```
/*
Recibe como parámetros las bases de datos, las tablas y el
atributo que serán utilizados para la medición
*/
Metodo_Calcular_Desviación_Correctitud_Semantica_Alfanumérico
(@BM: BaseDatos, @B: BaseDatos, @T1: Tabla; @T2: Tabla, @A:
Atributo)
/*
Calcula la medida por celda, invocando en este caso la rutina
generica 'Calcular_distancia', para cada celda del atributo
 (parámetro @A) de cada una de las tablas (parámetros @T1 y @T2)
de las bases de datos recibidas (parámetros @BM y @B).El
resultado lo inserta en una tabla temporal ##resultadocelda.
*/
Insert into ##resultadocelda (Tabla, IdTupla, IdAtributo,
medida) select @Tabla, IdTupla, @IdAtributo, calcular_distancia
(@BM. @T1. @A, @B. @T2. @A)
/*La inserción en la tabla definitiva se hará en la función que
invoca el método, llamando a la rutina que tomará la información
desde ##resultadocelda e insertará en MedidaResultadoCelda*/
```

3. Instanciación de Métodos

Los métodos serán instanciados por la herramienta **´Qbox-Foundation´,** pero como este trabajo se desarrolló en paralelo con la primera versión de dicha herramienta, desarrollamos funciones para la instanciación de los métodos. Un trabajo futuro será realizar la sustitución de estas funciones para que los mismos sean instanciados desde la herramienta **´Qbox-Foundation´.** Se presenta un pseudocódigo de ejemplo:

```
Invoco_Metodo_Edad_Maximo_de_tiempo_en_dias
/*
Inserta en la tabla ParametroInstanciado,invocando a la
función que realiza la inserción en dicha tabla, para cada
parámetro instanciado. Pasa como parámetros el nombre y valor
del parámetro, el método y el método instanciado
*/
```

```
exec TM_ins_parametroinstanciado 'BaseDatos', Calcular_Edad',
@B: BaseDatos,'Edad_Maximo_de_tiempo_en_dias'
exec TM_ins_parametroinstanciado, 'Tabla','Calcular_Edad', @T:
Tabla, 'Edad_Maximo_de_tiempo_en_dias'
exec TM_ins_parametroinstanciado 'Atributo', 'Calcular_Edad',
@A: Atributo, 'Edad_Maximo_de_tiempo_en_dias'
exec TM_ins_parametroinstanciado 'Funcion', 'Calcular_Edad', @F:
FuncionAgregacion, 'Edad_Maximo_de_tiempo_en_dias'
/*
Invoca el método con todos los parámetros a ser instanciados
(Base de datos= ClienteMaestra, Tabla= Personas,
Atributo=FecUltMod y Funcion=Max).
*/
exec  TM_Metodo_Calcular_Edad
(@B:'Cliente_Maestra',@T:'Personas', @A:'FecUltMod', @F: 'Max'):
@edad
/*
Si no hay error y granularidad es tabla:
Invoca la función genérica que inserta el resultado en la tabla
de resultados, según la granularidad del resultado. Envía como
parámetro la tabla (parámetro @T), el método instanciado
(parámetro @M), el valor de la medida calculada (@edad) y la
fecha de ejecución (parámetro @Fecha) como la fecha del sistema
*/
Si no hay error:
exec  TM_Alta_MedidaResultadoTabla @T:'Tabla', @M:
'Edad_Maximo_de_tiempo_en_dias', @edad, @Fecha: getdate()
```

6.1.3 Métodos implementados

En una primera etapa implementamos algunos métodos de medición, agendando los demás como trabajo futuro. Para la selección contemplamos diferentes aspectos tales como para cubrir todas las dimensiones de calidad, tener medidas de diferente granularidades e implementar métodos de diferente complejidad. En el cuadro siguiente se muestra la lista de métodos seleccionados en esta primera etapa:

Métrica	Métrica instanciada	Método	Granularidad
Edad	Edad-Promedio de tiempo en días	Calcular_Edad	**Tabla**
	Edad- Máximo tiempo en días		
Actualidad	Actualidad-Promedio de tiempo en días	Calcular_Actualidad	**Tabla**
	Actualidad-Máximo tiempo en días		
Desviación de correctitud semántica	Desviación de correctitud semántica-Distancia entre celdas	Calcular_Desviacion _Correctitud _Semantica_Alfanumerico	**Celda**
		Calcular_Desviacion _Correctitud	

Métrica	Métrica instanciada	Método	Granularidad
		_Semantica_Numerico	
Granularidad	Granularidad a nivel de registro	Calcular_Granularidad	**Registro**
Ratio de Cobertura	Ratio de Cobertura	Calcular_Ratio_Cobertura	**Tabla**
Ratio de Verificabilidad	Ratio de Verificabilidad	Calcular_Verificabilidad	**Área**
Ratio de integridad referencial	**Ratio de integridad referencial**	**Calcular_Ratio_Integridad _Referencial**	**Tabla**

Tabla 34- Lista de métricas implementadas

En la siguiente sub-sección se muestran los resultados obtenidos al aplicar dichos métodos.

6.2 Resultados obtenidos

En esta sub-sección presentamos el resultado de las mediciones realizadas sobre el contexto de aplicación.

Por cuestiones de confidencialidad, los datos y medidas presentados en este documento son ficticios pero se respetan las proporciones de las medidas obtenidas.

En los casos de las medidas con granularidad registro o celda, se mostrará un promedio de los resultados obtenidos. La información detallada a nivel de celdas y registros que quedó registrada en la base de datos, será puesta a disposición de la empresa mediante un reporte interno.

6.2.1 Dimensión Frescura

En las gráficas presentadas en las Figuras 6, 7 y 8 se representan los resultados de las mediciones realizadas para la dimensión *frescura* para las métricas:

- Edad-Promedio de tiempo en días
- Edad-Máximo tiempo en días
- Actualidad-Promedio de tiempo en días
- Actualidad-Máximo tiempo en días

Las mediciones de edad se realizaron en tablas de Base_Maestra. Los promedios obtenidos fueron valores bajos y aceptables, mientras que los máximos mostraron que existen datos que no han sido actualizados desde hace más de 10 años. Esto puede ocurrir debido a que es una base de datos que data de mucho tiempo, por lo que pueden existir datos que no se actualizan desde hace mucho tiempo.

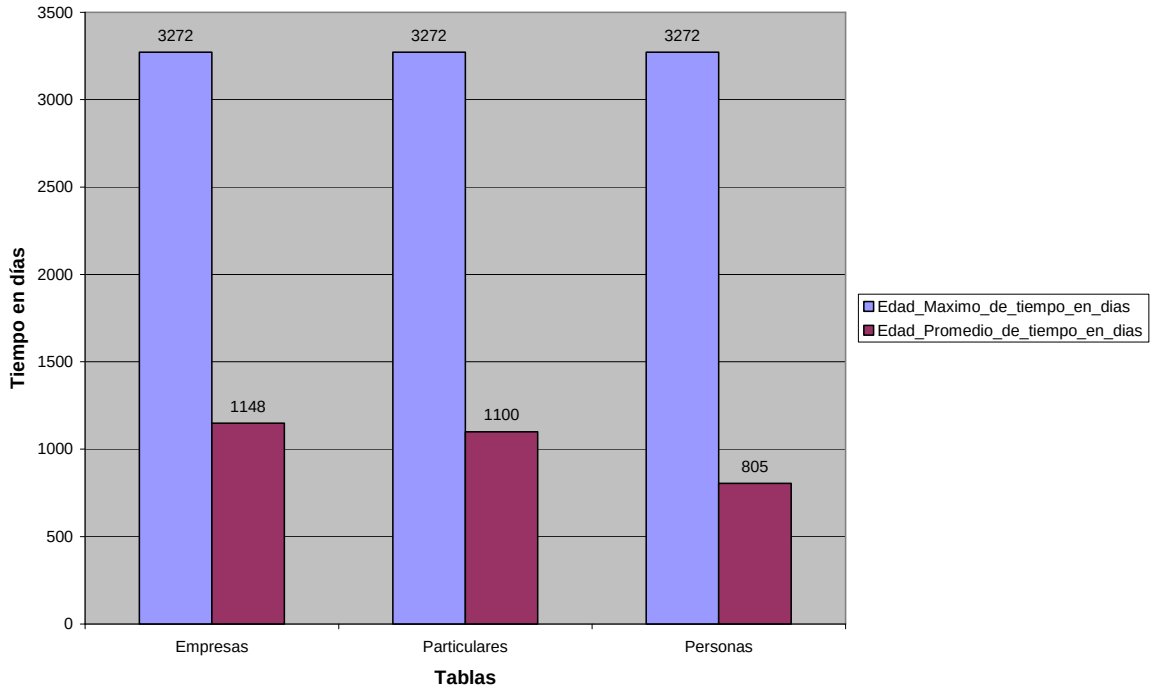

Figura 6- Medidas de Edad para tres tablas de la base maestra

La actualidad se midió en tres bases de datos suscriptoras. Los valores promedios muestran que la actualización de las réplicas suele llevar entre 1 y 2 días para las sucursales 1 y 2, pero es bastante más elevado para la sucursal 3. Esto es un elemento muy importante a considerar para analizar los procesos de actualización de las bases de datos, sea mediante la Herramienta de Integración u otros procesos de actualización de datos. Los valores máximos muestran que hay datos que no se están replicando en todas las suscriptoras o que al hacerlo no se actualiza la fecha de última actualización de registro, por lo que se deberán revisar los procedimientos de actualización de datos, sobre todo para la sucursal 3, para verificar que no existan fallas en los mismos o que se omita la actualización de la fecha de última modificación del registro.

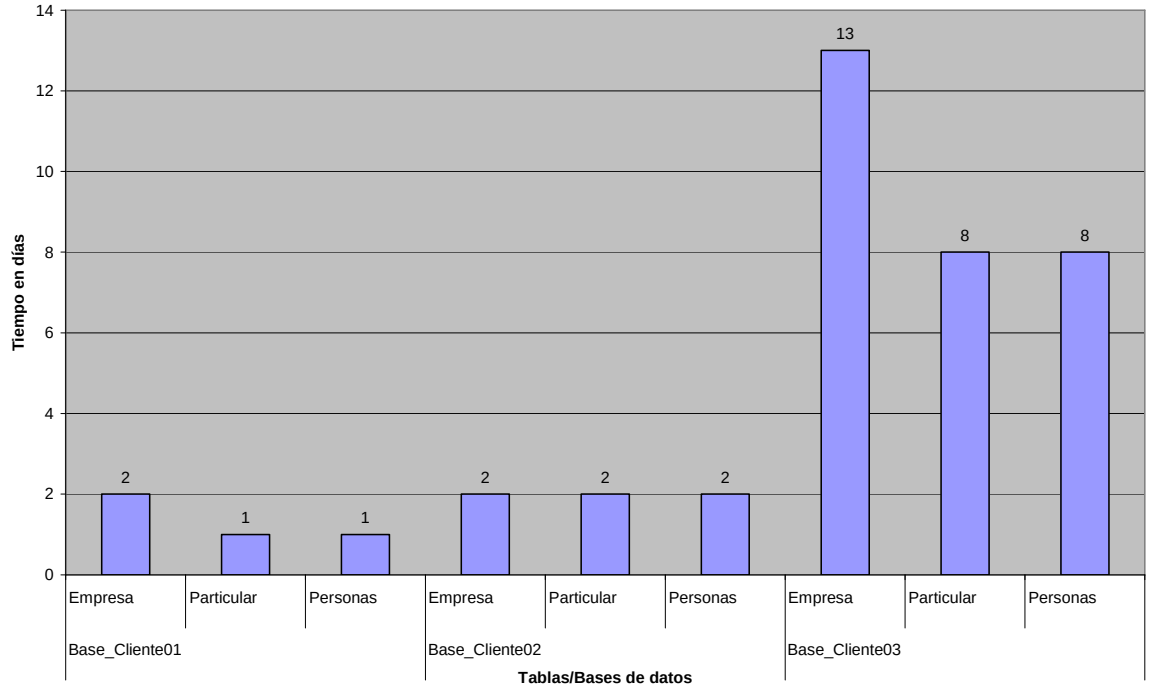

Figura 7- Medidas de Actualidad- Promedio de tiempo en días para tres bases de datos suscriptoras

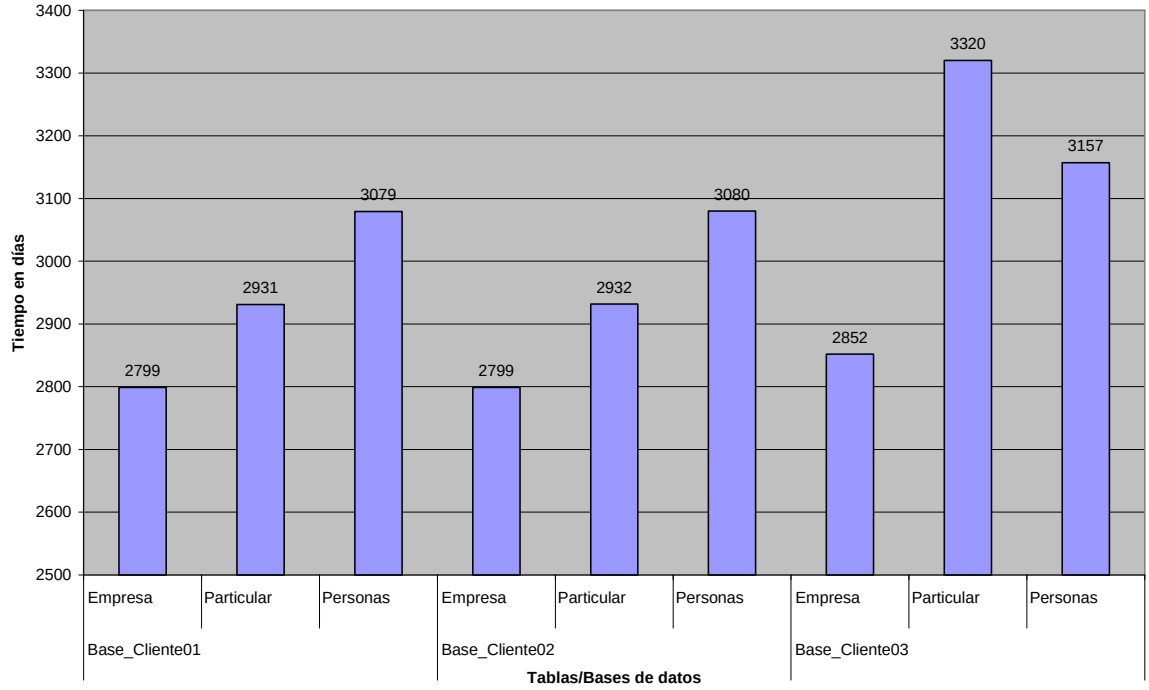

Figura 8- Medidas de Actualidad- Máximo de tiempo en días para tres bases de datos suscriptoras

6.2.2 Dimensión Exactitud

En la Figura 9 se presenta el resultado de la medición de *desviación de correctitud semántica* la cual es calculada midiendo la distancia entre los valores de cada celda de los atributos de la Base_Maestra versus la Base_Referencia. Seleccionamos las celdas

de tipo alfanumérico - nombre, primer apellido, primer nombre, ramo de actividad, segundo apellido, segundo nombre, sexo – y numérico – ramo de actividad-. Si bien esta métrica se obtiene a nivel de celda, para poder representar los resultados de una forma más simple en este documento, se calculó el promedio por atributo, lo cual es mostrado en la gráfica de la Figura 9. Los resultados de la medición a nivel de celda, para todos los atributos, están disponibles para ser consultados por la empresa, para su análisis.

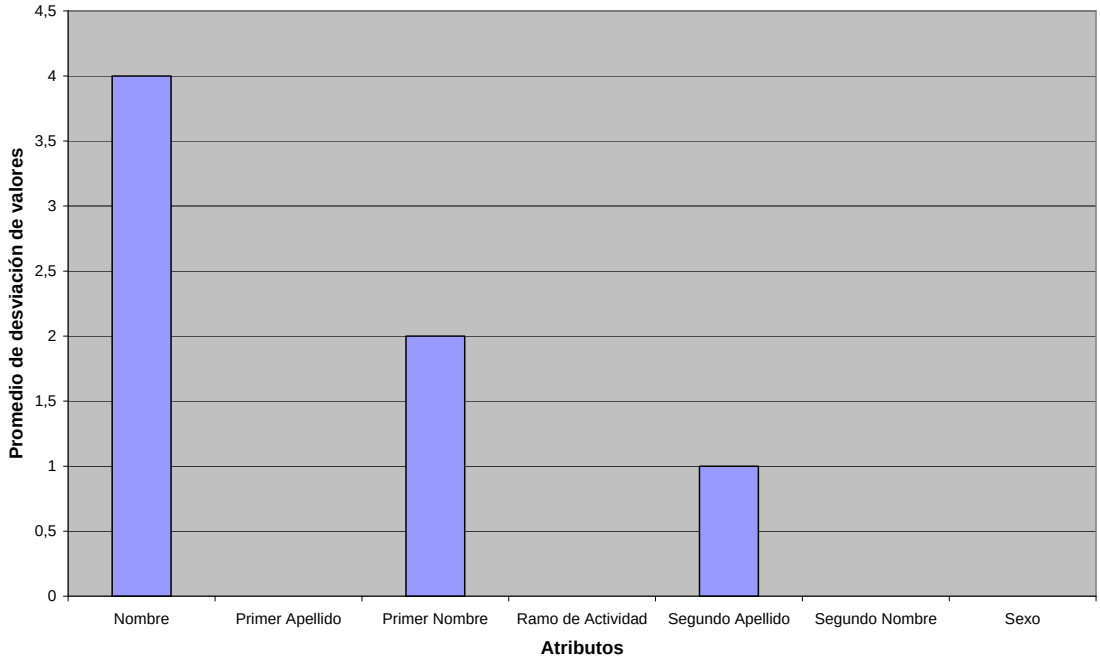

Figura 9- Promedios de medidas de Desviación de Correctitud Semántica de la Base_Maestra versus Base_Referencia para atributos alfanuméricos y numéricos

Se puede observar que los atributos Nombre, Primer Nombre y Segundo Apellido son los que presentan una mayor desviación de los valores reales. Las siguientes gráficas muestran la distribución de distancia para los atributos mencionados. Cada gráfica representa el porcentaje de celdas que desvían con el valor de distancia medida.

En la Figura 10 se puede observar para el primer nombre que el 87% tiene distancia 0, mientras que el 13 % restante tiene distancia entre 1 y 10.

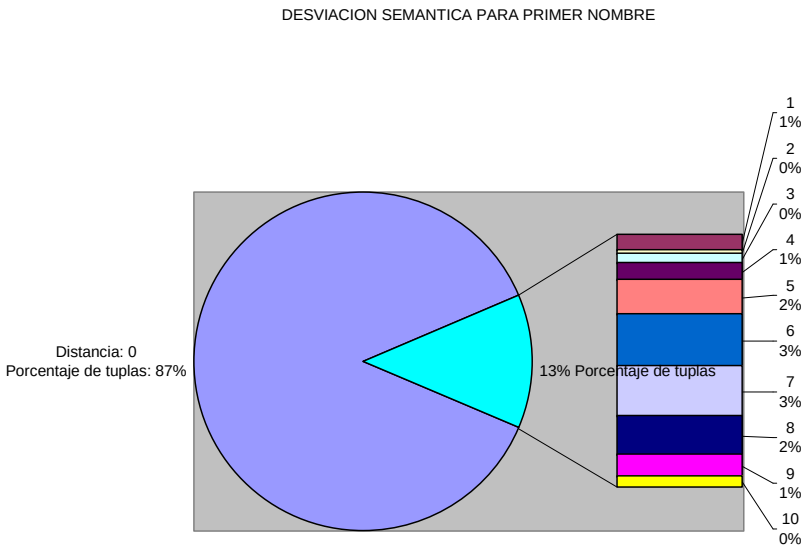

Figura 10- Medidas de desviación de correctitud semántica para el atributo primer nombre

En la Figura 11 se puede observar para el segundo apellido que el 82,1% tiene distancia 0, mientras que el 17,9% restante tiene distancia entre 1 y 11.

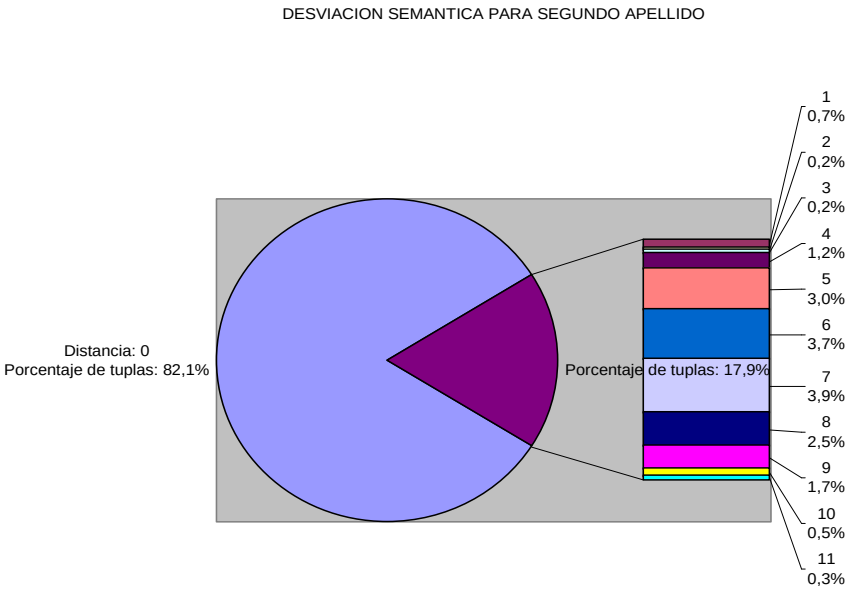

Figura 11- Medidas de desviación de correctitud semántica para el atributo segundo apellido

En la Figura 12 se puede observar que para el nombre que el 50% tiene distancia 0, mientras que el 50% restante tiene distancia entre 1 y 24.

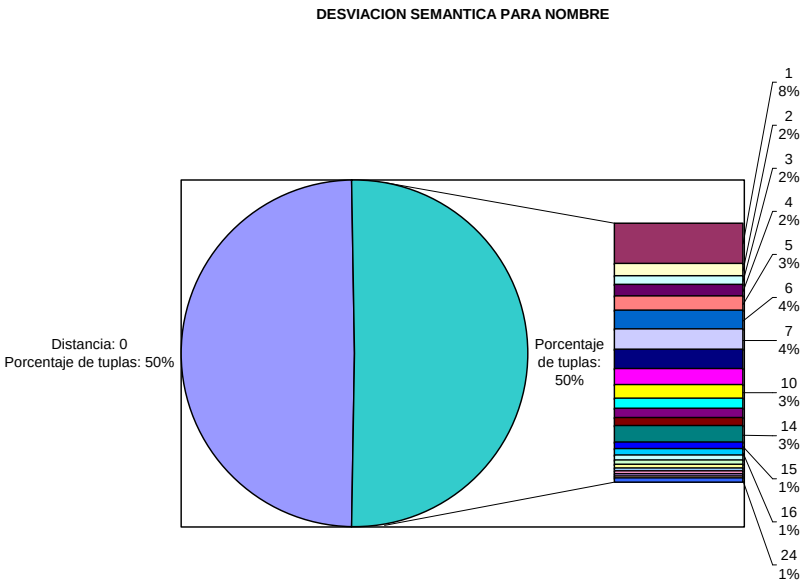

Figura 12- Medidas de desviación de correctitud semántica para el atributo nombre

Para el resto de los atributos, el promedio de distancia a nivel de atributo es despreciable, por lo que no se representan en las gráficas. Podemos concluir entonces que para los tres campos – primer nombre, segundo apellido y nombre – se debería realizar un trabajo de mejoramiento de los datos registrados en la Base_Maestra, refrescándolos de forma automática a partir de los datos de la Base_Referencia.

Otra de las medidas obtenidas para la dimensión *exactitud*, fue la *granularidad* para los conceptos *Domicilio (*compuesto por los atributos *calle, nropuerta, localidad, codpostal, país)* y *Nombre (*compuesto por los atributos *primerapellido y primernombre)*. Las mediciones se realizaron en la Base_Maestra. El método calculó la *granularidad* de cada concepto (conjunto de celdas) verificando para cada celda que integra el concepto, si es nulo o no. El resultado devuelto es el porcentaje de celdas no nulas para cada concepto de cada registro.

En las siguientes gráficas se muestra cómo se distribuyó la *granularidad* - en porcentaje de tuplas – para toda la tabla medida. En la Figura 13 se muestra la distribución de la *granularidad* para el concepto *domicilio* medido en la tabla Personas. Podemos observar que el 29% de registros tiene *granularidad* 100%, o sea todas las celdas de domicilio tienen valores no nulos, mientras que hay sólo 1% de registros que tienen *granularidad* 0%, o sea que todas las celdas de domicilio están en nulo. El 70% restante se distribuye entre *granularidad* 20% y 80%.

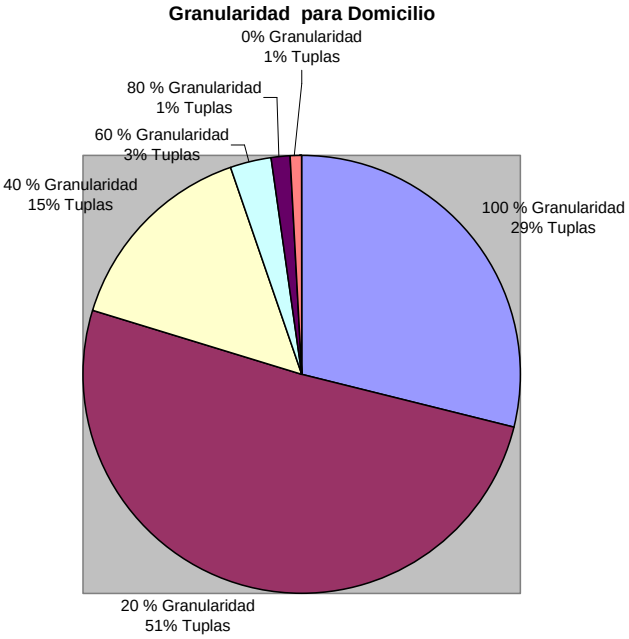

Figura 13- Medidas de granularidad de domicilio para la tabla Personas de Base_Maestra

En la Figura 14 se muestra la distribución de la *granularidad* para el concepto *nombre* medido en la tabla Particular. Podemos observar que el 79% de registros tiene *granularidad* 100%, o sea ninguna celda de las que componen el nombres están en nulo, mientras que el 21% de registros tienen *granularidad* 0%.

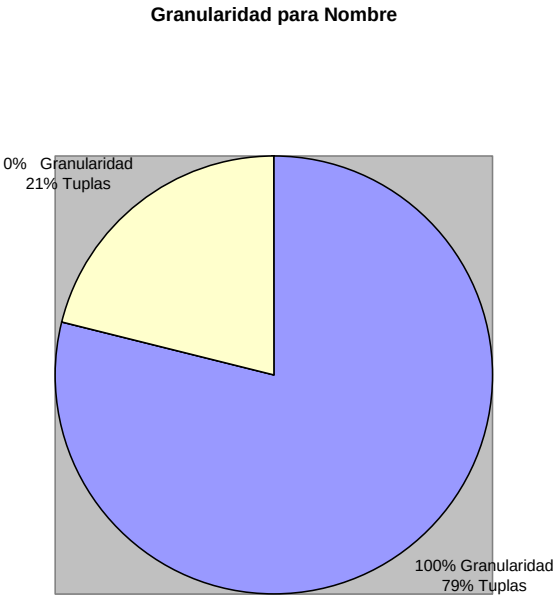

Figura 14- Medidas de granularidad de nombre para la tabla Particular de Base_Maestra

Por las métricas obtenidas, consideramos que para ambos conceptos se deberían tomar acciones correctivas, de manera de poder mejorar su calidad de los datos.

Adicionalmente, la métrica de *granularidad* para *domicilio* se podría refinar, ponderando según la importancia del dato que está en nulo. Por ejemplo, es más importante que la *calle* esté en nulo a que esté el *código postal* en nulo. Para el caso del concepto *nombre* ambos atributos seleccionados son de igual importancia.

6.2.3 Dimensión Completitud

En la Figura 15 se muestra el resultado de la medición del Ratio de Cobertura en las tablas de tres bases suscriptoras. Las mediciones de completitud se realizaron para todas las tablas de las bases de datos de 3 sucursales (suscriptoras). Para cada uno de los registros que existen en las tablas de Cliente_Maestra, se buscó si existe el registro de igual clave en la tabla de igual nombre de cada una de las bases suscriptoras. El resultado es el promedio de registros que existen en ambas bases (nodo madre y base suscriptora).
Podemos observar que:

- La base de datos Base_clientes03 presenta un 99% de cobertura para cuatro tablas, mientras que las otras dos bases de datos presentan un 99% de cobertura sólo para una tabla; para el resto de las tablas presentan un 100% de cobertura;
- La tabla Cuenta presenta un 99% de cobertura para las tres bases de datos suscriptoras

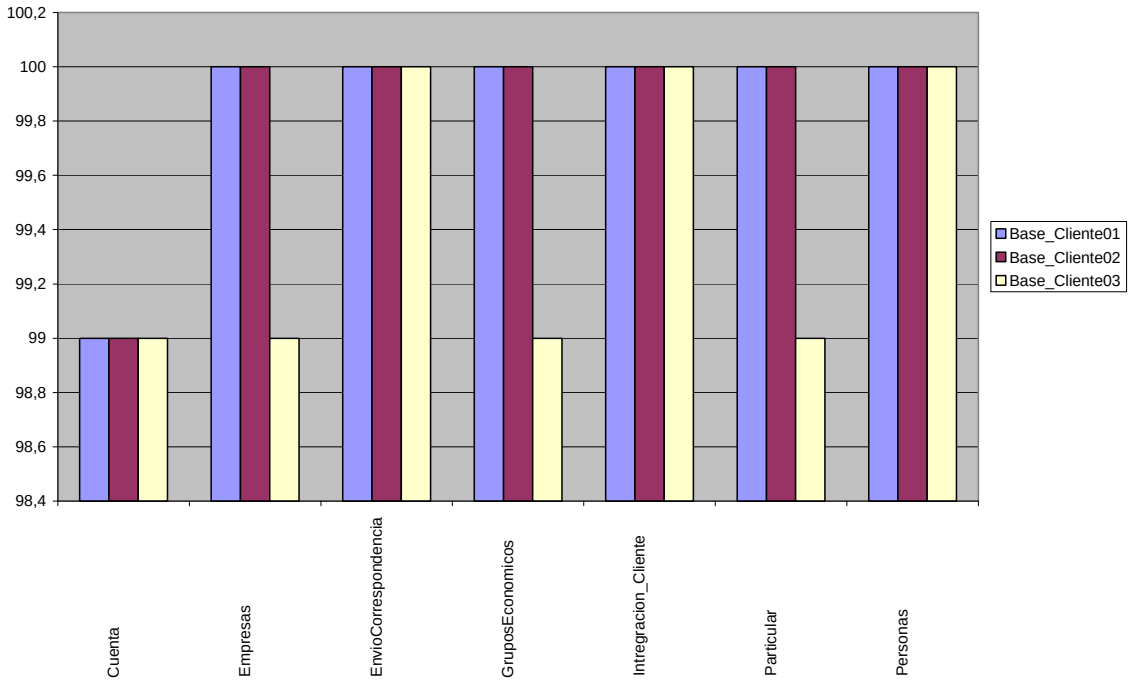

Figura 15- Medidas de ratio de cobertura para las tablas de tres bases suscriptoras

Por lo que podemos concluir que se deberán revisar los procedimientos de replicación de la tabla Cuenta para todas las bases de datos y **todos los procedimientos** de replicación para la base de datos Base_cliente03.

6.2.4 Dimensión Integridad

Para la dimensión integridad, medimos el ratio de integridad referencial y sus resultados se muestran en la Figura 16. Para medir el porcentaje de integridad, sólo se tomó en

cuenta la tabla Persona de la Base_Maestra y se verificó si los clientes estaban registrados en las tablas correspondientes, según las siguientes condiciones:

1. Si TipoCliente=’P’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Particular;
2. Si TipoCliente=’E’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Empresa;
3. Si TipoCliente=’C’ entonces Identificacion de la tabla Persona debe referenciar a un registro de la tabla Integracion_Cuenta.

Como podemos observar en la Figura 16 la integridad referencial respecto a la tabla Empresa es del **100 %** por lo que todos los registros de la tabla Persona que cumplen la condición de TipoCliente=’E’ tienen su correspondiente registro en la tabla Empresa. Por otro lado respecto a la tabla Particular la integridad fue de **99%** y respecto a la tabla Integracion_Cuenta la integridad fue de **84%.**

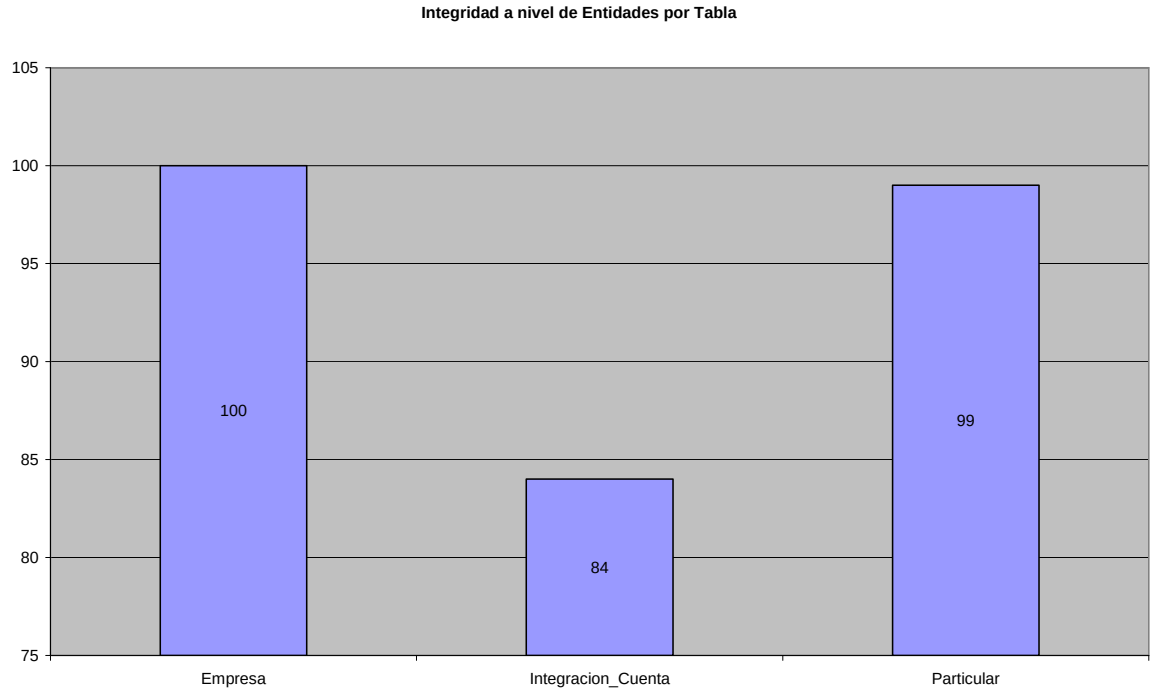

Figura 16- Medidas ratio de integridad referencial en Base_Maestra

Podemos observar que la tabla más afectada en su integridad referencial es Integracion_Cuenta, seguida por la tabla Particular, mientras que Empresa no presenta problemas. Por lo tanto, las medidas obtenidas demuestran la necesidad de revisar los procedimientos de ingreso de datos para los tipos de cliente particular y cliente con más de un integrante.

6.2.5 Dimensión Trazabilidad

Por un problema de volumen en la base de datos creada para las mediciones de esta tesis, transferimos sólo las trazas de los últimos tres meses de uso de la base de datos. Por lo tanto, a los efectos de la medición, el área quedó restringida a un **13%** del área original.

El resultado obtenido para el *ratio de trazabilidad* es **100% para la muestra tomada.** Si bien esto no nos permite afirmar que todos los registros modificados desde la

implantación de la Herramienta de Integración tendrían trazabilidad, esta primera medición es muy alentadora. Tenemos planteado realizar mediciones de otros períodos de tiempo para confirmarlo.

6.3 Dificultades encontradas

Durante el desarrollo de los métodos y funciones de medición se encontraron diferentes dificultades, las cuales se analizaron y en consecuencia se adoptaron acciones para solucionarlas. A continuación presentamos las que consideramos más significativas, ya sea porque constituyeron una limitación en las soluciones o resultados obtenidos y pueden servir de apoyo para interpretar los resultados, o en el entendido que son experiencias que pueden ser útiles para futuros trabajos relacionados:

i. *Rendimiento de los métodos*: uno de los problemas encontrados fue el rendimiento de algunos métodos, en particular se encontró un rendimiento muy bajo para los casos que la granularidad es a nivel de celda o registro. Se adoptaron por tanto, acciones correctivas, agregando índices en el ambiente creado para la tesis. Tal es el caso de la tabla Trabajo de la Base_Referencia. Este es un tema a tener en cuenta, ya que si se quisieran aplicar los métodos en la instalación real, se deberá analizar la creación de nuevas estructuras para poder obtener un rendimiento aceptable.

ii. *Estructuras disímiles*: Otras dificultades encontradas fueron relacionadas a la estructura de las bases de datos, a saber:
   - La tabla Trabajo cuenta con diferente estructura y nomenclatura que las tablas de las otras bases de clientes, por lo que la comprensión y lectura de la misma se hace muy dificultosa. Para solucionar este problema se creó una vista lógica que utiliza la misma nomenclatura y el mismo criterio de almacenamiento de datos, que las tablas de la base Cliente_Maestra. Asimismo, se agregó a la vista la información necesaria para que se pueda acceder por el atributo clave – Identificación - de las tres tablas principales de datos de Cliente_Maestra (Persona, Empresa y Particular);
   - Cuando es necesario hacer una unión entre dos o más tablas, el acceso a las mismas es por atributos con diferentes nombres o diferentes criterios de almacenamiento; para solucionarlo se extendió la metadata donde se registran los Objetos del Sistema de Información (bloque 3 de la Figura 4), agregando la identificación de los atributos por los que se accede cada tabla (clave primaria) y en el caso de ser necesario para hacer unión de tablas – por ejemplo para medir la integridad - cuál es su relación con la tabla de la cual hereda la clave.

iii. *Gran volumen de información*: el volumen de información también constituyó una dificultad en el momento de armar el ambiente para realizar las mediciones. En algunos casos, tales como la base de Trazas, se optó por tomar una muestra de un 13% de la partición definida inicialmente, ya que el volumen de datos es muy grande y no fue posible almacenar una réplica en el servidor dispuesto para realizar las mediciones. Asimismo, se debe tener en cuenta el volumen de información a ser registrado en la base de resultados, cuando se trata de granularidad registro o celda.

iv. *Utilización de procedimientos o funciones del Sistema del motor de base de datos (System Function o System Stored Procedures):* para cargar el catálogo de la base de datos se utilizaron algunos procedimientos almacenados o funciones del sistema, los cuales tienen restricciones, por lo que, para lograr los objetivos de esta tesis, se tuvieron que reescribir. Tal es el caso del procedimiento almacenado de sistemas llamado "*sp_pkeys*" que se utiliza para identificar la clave primaria y era necesario para la implementación de los métodos. Asimismo, al utilizar SQL dinámico, nos era dificultoso capturar el resultado de su ejecución, por lo que realizamos una investigación de la implementación del "*sp_executesql*" y para obtener una solución más eficiente, adoptamos y reescribimos parte de su implementación dentro de los métodos.

v. *Complejidad.* Para implementar la métrica *Similaridad*, nos encontramos que la complejidad y el costo de la implementación para la solución planteada eran muy altos. Hicimos algunas experimentaciones utilizando soluciones alternativas con SQL Dinámico (utilizando *cross apply*), pero de todas formas el rendimiento no fue satisfactorio. Consultamos al representante local de Microsoft y la bibliografía para optimizar la solución [Monge+1997] y concluimos que debíamos experimentar con otras herramientas para mejorar la solución, por ejemplo *.net*. Esta solución no se desarrolló en el marco de esta tesis y quedará para futuras implementaciones.

6.4 Síntesis

Uno de los objetivos de esta tesis es contar con un conjunto de métodos de medición paramétricos desarrollados y un conjunto de resultados que puedan ser utilizados en la empresa para análisis y eventuales acciones de mejora de la calidad.
Desde el punto de vista de los objetos construidos, podemos decir que nos aportó mucho la investigación y la experiencia de la construcción propia de los métodos de medición, poniendo nuestro enfoque en la parametrización de los mismos. Buscamos la mejor solución tomando en cuenta diferentes propiedades tales como: parametrización, mantenibilidad, reutilización, funcionalidad y performance. Las dos propiedades más fuertes que fueron combinadas en todo el desarrollo fueron **parametrización** y **performance**, ya que pudimos observar que muchas veces la parametrización iba en desmedro de la performance. En tal sentido, investigamos las diferentes funcionalidades - rutinas y procedimientos de sistemas, utilitarios, entre otros – que provee el motor de la base de datos, y buscamos soluciones en otras fuentes – internet, entre otras - para poder adaptarlos a nuestra solución de acuerdo a nuestros objetivos. A su vez, la evolución natural de los datos y de la arquitectura, nos permitió experimentar la definición de particiones para realizar las mediciones, por ejemplo para *trazabilidad*.

Cumplida la etapa de implementación, y salvadas las dificultades que se presentaron podemos constatar que nuestros conocimientos teóricos pueden ser llevados a la medición de la calidad de una aplicación real, determinando así la calidad de un sistema de información con un esquema replicado.

Concluimos que se cumplió por tanto el objetivo de construir una biblioteca de métodos parametrizables y una base de datos con medidas de calidad, la cual desde el punto de vista académico puede ser utilizada como punto de partida para analizar propiedades de las métricas de calidad y desde el punto de vista de la empresa, puede constituir un activo para analizar y mejorar la calidad de las bases de datos.

7. Conclusiones

En este capítulo presentamos las conclusiones del trabajo realizado, incluyendo un resumen de las actividades, aportes realizados y una proyección para trabajos futuros.

7.1 Resumen

Se dice que la calidad es una propiedad que se percibe más fácilmente cuando no está, ya que puede traer serias consecuencias para la eficiencia y efectividad de las organizaciones y de los negocios. Desde el punto de vista empresarial, existen varios motivos para querer mejorar la calidad de los datos, tanto para los procesos operativos como para los procesos de toma de decisiones.

Es por esto que hemos estudiado cómo medir la calidad de las bases de datos de clientes de una empresa. Comenzamos nuestro trabajo analizando el contexto de aplicación – arquitectura, procedimientos de actualización, memoria organizacional - centrándonos en el análisis de los problemas. Basándonos en el paradigma 'Goal-Question-Metric', seguimos un enfoque 'top-down', identificando los objetivos de calidad, descomponiendo los objetivos en preguntas asociadas a factores de calidad. Concluimos esta etapa con una exhaustiva selección de métricas e instanciación de métodos de medición. Esta actividad nos posibilitó contrastar la teoría con lo real, realizando actividades que se desarrollan de forma intuitiva normalmente en las empresas, pero esta vez enmarcándolas en una metodología. Nos centramos en un problema real, analizamos y buscamos los principales problemas, tratamos de comprender los objetivos de los usuarios y de los técnicos, estudiamos la memoria organizacional de la empresa y nos realizamos las preguntas que estos actores se plantearían. Para las etapas siguientes, necesitamos profundizar más en la teoría para poder ligar la actividad práctica de comprensión de los problemas y rotulamiento de los objetivos, con la definición de factores y métricas de la teoría. Como resultado de esta etapa destacamos la definición de objetivos, preguntas, dimensiones, factores y métricas de calidad y la interacción que pudimos realizar entre el análisis de los requisitos de un caso real y la teoría.

En paralelo analizamos el estado del arte de cada una de las dimensiones definidas de interés para la aplicación - *frescura, exactitud, completitud, trazabilidad, integridad y unicidad.* En esta actividad básicamente, analizamos, organizamos y estudiamos, la información que encontramos sobre dimensiones, factores y métricas. A medida que fuimos avanzando en esta actividad, pudimos notar la importancia que va tomando cada vez más la calidad de datos, ya que en el estudio de cada una de las dimensiones, pudimos percibir que la noción de calidad está presente en todos los ámbitos, para diferentes dominios de aplicación y para diferentes tipos de aplicación. Esta actividad nos posibilitó extender nuestros conocimientos teóricos para luego ser experimentados en el caso práctico.

Posteriormente, definimos la instanciación de métodos y métricas, sobre lo cual basamos nuestra etapa de construcción. En esta etapa obtuvimos una biblioteca de métodos (extensible y parametrizable) que puede ser utilizada en aplicaciones de la vida real, la cual interactúa con la plataforma para medición de calidad, llamada ´Qbox-Foundation´. Para asegurarnos la reutilización y parametrización de los métodos, utilizamos para su construcción el lenguaje SQL Dinámico. En tal sentido tratamos de

combinar dos propiedades que consideramos muy importantes -**parametrización** y **performance -** ya que pudimos observar que muchas veces la parametrización iba en desmedro de la performance. Buscando el refinamiento de los datos obtenidos, definimos y aplicamos áreas de acuerdo a la calidad de las fuentes. Una vez obtenidas las mediciones, hicimos un análisis de los resultados obtenidos, dejando la información detallada disponible para que la empresa pueda continuar con el trabajo.

En síntesis, podemos decir que hemos realizado una experiencia de medición de factores y métricas de calidad en una base de datos de una aplicación real cumpliendo con nuestros objetivos de experimentar y ampliar los conocimientos teóricos de algunas de las dimensiones de calidad. Obtuvimos una biblioteca de métodos con las siguientes características: i) se trabajó sobre una aplicación real; ii) el dominio de aplicación seleccionado fue una base de datos de clientes; iii) el contexto de trabajo es un ambiente de bases de datos replicadas; iv) el área de negocios es una Institución Financiera.

Dejamos por tanto un gran aporte al sector empresarial que es la experiencia de la medición de un caso real, y la biblioteca de métodos que puede ser reutilizable, ya que es extensible y puede ser utilizada en otros dominios de aplicación, con otras arquitecturas y en otras áreas de negocio empresarial.

En la siguiente sub-sección resumimos nuestros principales aportes.

7.2 Aportes

Resumimos las principales contribuciones de esta tesis como:

1. *Estado del arte sobre medición de la calidad:* Se realizó un análisis de las dimensiones *frescura, exactitud, completitud, trazabilidad, integridad y unicidad*. El principal resultado de este análisis es la recopilación, organización y presentación de los factores y métricas de calidad asociadas a las dimensiones mencionadas. Para algunas de ellas – por ejemplo exactitud y frescura- , encontramos mucho material en la literatura, pero para otras – por ejemplo: trazabilidad– en la literatura consultada no encontramos suficiente material por lo que consideramos que se debe continuar profundizando en ellas.

2. *Análisis de necesidades de calidad en una aplicación real*: El principal aporte de esta actividad, es haber aplicado la teoría en un caso real. Del análisis de los problemas reales existentes, definimos los objetivos de calidad, y los asociamos a factores y métricas de calidad. El detalle de cómo se desarrolló la actividad es un aporte interesante, ya que sirve como guía para aplicarla en otros casos de la vida real.

3. *Propuesta de un mecanismo parametrizable de definición de métodos de medición:* Nuestra contribución fue la definición de métodos parametrizables que pueden ser instanciados en diferentes contextos de aplicación. Se empleó una metodología para la definición e instanciación de métricas y métodos. Adicionalmente definimos los metadatos para el registro de los resultados de realizar la medición instanciando los métodos. La definición fue realizada en un ambiente de replicación de bases de datos, lo cual puede ser expandido a otros tipos de arquitectura.

4. *Prototipo de una plataforma parametrizable para la medición de la calidad de datos:* El principal resultado de nuestro trabajo es la implementación de una biblioteca de dimensiones, factores, métricas y funciones parametrizables de medición, que es extensible, pudiendo incorporar nuevos conceptos de calidad, según sea requerido. Esta biblioteca será integrada a la herramienta ´Qbox-Foundation´.

5. *Explotación de las medidas obtenidas:* Por último y relacionado a las mediciones realizadas, pensamos que los resultados de medición obtenidos serán de utilidad a la empresa para continuar incrementando su esfuerzo en pos de la calidad de datos, realizando tareas tales como *mantenimiento de estadísticas, particionamiento de las tablas de acuerdo a su calidad, mejoras en la explotación de la información, data cleaning*, entre otras.

En la siguiente sub-sección proponemos trabajos futuros.

7.3 Trabajos futuros

Como trabajos futuros nos planteamos:

- Completar y refinar el desarrollo realizado, implementando y ejecutando todos los métodos planificados;

- En base a las mediciones obtenidas, refinar las preguntas de calidad y las métricas correspondientes. Por ejemplo para la métrica *Verificabilidad*, se podrían elegir algunos casos de uso para la medición tales como: “Alta de cliente particular”, “Alta de cliente empresa”, entre otros;

- Agregar algunas herramientas al desarrollo de los métodos. Por ejemplo, para la métrica *Similiaridad* se podría usar otras herramientas de programación (.net) que mejoren la performance del método;

- Afinar la integración con la herramienta ´Qbox-Foundation´. Para dicha herramienta los métodos son clases JAVA por lo que debemos construir las clases que invoquen los métodos implementados. Además sugerimos extender el esquema relacional que implementa el metamodelo de ´Qbox-Foundation´, incorporando nuestra metadata la cual es más detallada;

- Realizar estudios similares en otros contextos de aplicación;

- En el estudio de la literatura, encontramos que algunos autores se interesan en el estudio de los problemas de calidad derivados por problemas en los diseños de las bases de datos (tipos de datos inadecuados, problemas de normalización, entre otros). Con los avances de la tecnología y de los métodos de diseño, y con la realidad que tenemos, de que en muchas empresas los “legacy systems” cumplen un papel preponderante, consideramos que este es un aspecto muy interesante para seguir profundizando e investigando.

Anexo I. Esquemas de las bases de datos

En el presente anexo se incluyen los esquemas de las siguientes bases de datos a ser referenciadas en esta tesis:

- Base de datos Maestra
- Base de datos Referencia
- Base de datos Trazas

La notación utilizada por la herramienta de diseño utilizada (ER/Studio versión 6.6) es presentada en el Notación de la herramienta ERStudio.

I.1 Base de datos Maestra

En la Figura 17 se presenta el esquema de base de datos de la Base_Maestra. Por problemas de espacio, sólo representamos las entidades y atributos que son referenciados en esta tesis. Las tablas de datos se representan de color.
El esquema de base de datos de las réplicas – Base_Cliente - es similar, salvo que no cuentan con integridad referencial ni reglas de dominio.

Como se puede observar en el esquema de la Figura 17, las tablas no contienen reglas de integridad entre ellas. A continuación presentamos algunos conceptos que consideramos pueden ser de utilidad para comprender el modelo y la relación lógica entre las tablas:

- El atributo *Identificacion* es el número único que identifica a la persona.

- El atributo *Cuenta*, identifica el número de cuenta que la persona tiene en el Banco. Una persona registrada en Persona, puede tener de *0 a n* cuentas.

- En las tablas Documentos y GruposEconomicos, el atributo *Cliente* se corresponde con el atributo *Identificación* de la tabla Persona.

- En la tabla de relacionamiento Integracion_Cuenta, el atributo *Identificacion* se corresponde con el número ficticio que se le da a un cliente no individual. El atributo *IdentificacionCC* se corresponde a todas las personas que están relacionadas al mismo.

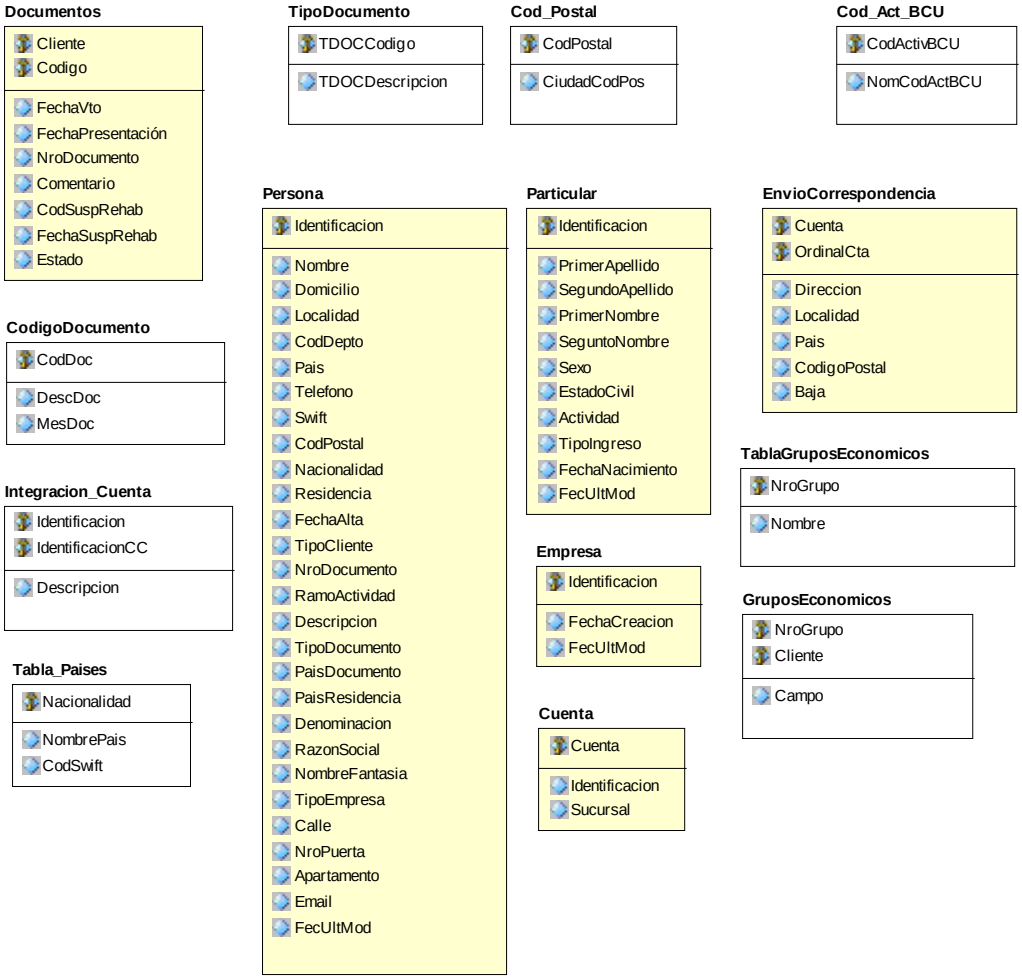

Figura 17- Esquema de la base de datos Base_Maestra

I.2 Base de datos Referencia

En la Figura 18 se presenta la vista Trabajo, definida sobre la tabla Trabajo de la base de datos Referencia. Esta vista se creó para resolver los problemas de diferencia de nomenclatura.

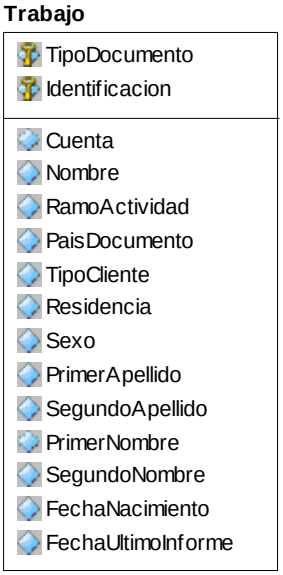

Figura 18- Esquema de base de datos Referencia

I.3 Base de datos Trazas

En la Figura 19 se presenta el esquema de la tabla TrazaMovimiento la cual es la elegida para las mediciones definidas en el marco de este trabajo. No consideramos relevante presentar el resto de las tablas del esquema, ya que no son referenciadas en este documento.

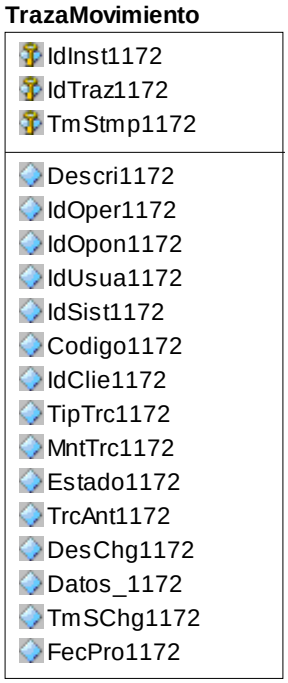

Figura 19- Esquema de la base de datos Trazas

Debido a que no es sencillo deducir de la nomenclatura el contenido de los atributos, en la Tabla 35 detallamos algunos de ellos que nos interesan para la tesis:

Atributo	Descripción
IdClie1172	Identificación del cliente para el cual se realizó el movimiento
TmStmp1172	Fecha y hora de realizado el movimiento
Descri1172	Existen 3 tipos de descripciones: *1)De inicio de proceso*: “Inicio Ejecución” *2) De información de ejecución*: si esta descripción es “Sin Error” el proceso corrió correctamente, si es distinto a ello el proceso tuvo errores y generalmente el texto ahí dispuesto es bastante explicativo. *3)De fin*: “Fin Ejecución”

Tabla 35- Atributos utilizados de la tabla TrazaMovimiento

Anexo II. Descripción de rutinas genéricas

A continuación se describen las rutinas genéricas que fueron implementadas para resolver los métodos. Estas rutinas son utilizadas con frecuencia, en todos los métodos que así lo requieran, y son invocadas en la sección 5.

II.1 Rutina Chequear_Nulo

La rutina genérica Chequear_Nulo es para determinar si un atributo es nulo o no. Su cabezal es el siguiente:

Chequear_Nulo (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano.

La rutina recibe como parámetros la celda (base de datos, tabla, atributo e identificador de tupla) para la que se quiere determinar si el valor es nulo o no.

II.2 Rutina Comparar_Atributos

La rutina genérica Comparar_Atributos calcula por registro cuantos atributos están iguales entre ambas tablas recibidas como parámetros. Su cabezal es el siguiente:

Function Comparar_Atributos (BM: BaseDatos, B: BaseDatos, T1: Tabla, T2: Tabla, I: IdTupla): Cantidad_atributos_iguales

Esta función recibe como parámetros las dos bases de datos (parámetros BM y B) a las que pertenecen las tablas a comparar (parámetros T1 y T2 respectivamente) y la identificación de la tupla a comparar.
Aplica para cada atributo del registro la fórmula de comparación, F= (BM.T1.Atributo = B.T2.Atributo) para las bases de dato y tablas recibidas como parámetro. Devuelve la cantidad de atributos que contienen igual información. Si todos los atributos son diferentes devuelve cero. La lista de atributos a ser comparados por tabla, así como la clave primaria para realizar la unión entre ambas tablas a ser comparadas, se obtendrán de la metadata de las bases de datos. Notar que se asume que el nombre de los atributos para cada tabla son iguales.

II.3 Rutina Calcular_Distancia

La rutina genérica Calcular_Distancia calcula la distancia entre dos cadenas de caracteres. Su cabezal es el siguiente:

Function Calcular_Distancia (X: String, Z: String): Distancia

Esta función recibe como parámetros dos cadenas de caracteres (parámetros X y Z) y devuelve un valor numérico que indica la distancia entre ambos.

La función Calcular_Distancia, se basa en la **Distancia de Levenshtein.** La **Distancia de Levenshtein** o **Distancia de edición** es el número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Se entiende por operación: una inserción, eliminación o la substitución de un carácter para transformar una cadena fuente “x” en una cadena objetivo “z” [Wipikedia2008] [Torres2006].

Por ejemplo:

- Si x = "hola" y z = "hola", entonces Calcular_Distancia (x,z) = 0, porque no es necesario hacer ninguna transformación para llegar de una cadena a la otra. Las cadenas fuente y objetivo son idénticas.
- Si x = "hola" y z = "ola", entonces Calcular_Distancia (x,z) = 1, porque es necesaria una transformación para que la cadena fuente y objetivo sean iguales (eliminar la “h”).
- Si x = "hola" y z = "hilo", entonces Calcular_Distancia (x,z) = 2, porque son necesarias dos transformaciones para que la cadena fuente y objetivo sean iguales (cambiar la "o" por la “i” y cambiar la "a" por la "o").

II.4 Rutina Chequear_Regla_Dominio

La rutina genérica Chequear_Regla_Dominio determina si se cumplen las reglas de dominio para un atributo de una tabla determinada. Su cabezal es el siguiente:

Function Chequear_Regla_Dominio (B: BaseDatos, T: Tabla; A: Atributo; R: TablaReglaDominio; I: IdTupla): Booleano

Esta función chequea, para cada regla de dominio que debe cumplir la celda (base de datos, tabla, atributo e identificación del registro) si cumple con las reglas de dominio establecidas en una tabla (parámetro R) de la base de datos. Devuelve un valor booleano, dependiendo si la regla se cumple o no.

En el contexto de aplicación se registran algunas excepciones que deberán ser tratadas para chequear el dominio de una celda. Debido a la evolución del modelo de datos, como se presenta en la sección 4.3, algunos atributos fueron agregados o fueron definidos como obligatorios, los cuales fueron completados con valores ceros para los numéricos o blanco para los alfanuméricos para los registros ya existentes en la base de datos. Por ejemplo: se agregó el atributo País del Documento el cual es obligatorio a partir de una nueva reglamentación. Para todos los registros que ya existían en la base de datos y que no se pudo inferir el dato automáticamente, se le cargó el valor blanco. El conjunto de valores de dominio permitidos, incluyó el valor blanco, entre otros: “UY”, “AR”, “BR”, “ “.
Por lo tanto, los atributos que registran blancos o ceros en los atributos obligatorios se tomarán como que no cumplen las reglas de dominio. No existe en el contexto de aplicación un caso similar para otros tipos de datos, como por ejemplo fecha.

Para poder evaluar los atributos en blanco o ceros, para los tipos de atributo alfanumérico y numérico respectivamente, utilizaremos las siguientes dos funciones genéricas, las cuales serán llamadas desde la función Chequear_Regla_Dominio:

Function Chequear_ Blanco (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano definida en II.6 o
Function Chequear_Cero (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano definida en II.7.

II.5 Rutina Chequear_Regla_Atributo

La rutina genérica Chequear_Regla_Atributo determina si cada celda almacena la información esperada de acuerdo a reglas establecidas. Su cabezal es el siguiente:

Function Chequear_Regla_Atributo (B: BaseDatos, T: Tabla; A: Atributo; R: TablaReglaAtributo; I: IdTupla): Booleano

Esta función chequea, para cada celda, que tenga la información esperada, según las reglas almacenadas en la tabla de reglas para atributos (parámetro R). Devuelve un valor booleano, dependiendo si la regla se cumple o no.

A continuación se presentan algunas de las reglas que vamos a controlar para los campos de domicilio:

- Localidad, País: caracteres diferentes a letras, como para poder detectar registros del estilo “Paris-Francia”.
- Calle: caracteres diferentes a letras y números, como para poder detectar registros del estilo “Tacuarembo 1589-Guichon”
- Calle: caracteres de muchos números consecutivos. Es permitido en el atributo calle contar con números, por ejemplo “18 de Julio”, pero se quiere detectar que además de la calle, la celda no almacene el número de puerta, por ejemplo “Tacuarembo 1589”.
- Número de puerta: caracteres diferentes a números, como para poder detectar registros del estilo “1234 bis”

II.6 Rutina Chequear_Blanco

La rutina genérica Chequear_Blanco chequea si una celda está en blanco. Su cabezal es el siguiente:

Function Chequear_ Blanco (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano

La rutina recibe como parámetro la celda (base de datos, tabla, atributo e identificador de la tupla) y devuelve un valor booleano dependiendo de si la celda es blanco o no.

II.7 Rutina Chequear_Cero

La rutina genérica Chequear_Cero chequea si una celda está en cero. Su cabezal es el siguiente:

Function Chequear_Cero (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano

La rutina recibe como parámetro la celda (base de datos, tabla, atributo e identificador de la tupla) y devuelve un valor booleano dependiendo de si la celda es cero o no.

II.8 Rutina Analizar_Trazas

La rutina genérica Analizar_Trazas determina si para un registro se pueden identificar las trazas. Su cabezal es el siguiente:

Function Analizar_trazas (B: BaseDatos; T1: Tabla; T2: Tabla, I: IdTupla, F: Fecha): Booleano

Para cada celda recibida como parámetro (parámetros I) accede a la tabla (parámetro T2) de la base de datos Trazas (parámetro B) y analiza si existen trazas para la fecha de modificación o creación del registro (parámetro F).

Devuelve un booleano que indica si pudo obtener el usuario que realizó la modificación o alta del registro de clientes para la identificación y fecha de modificación dadas.

II.9 Rutina Chequear_Restricciones

La rutina genérica Chequear_Restricciones evalúa si una celda cumple con las restricciones definidas en una tabla de la base de datos. Su cabezal es el siguiente:

Function Chequear_restricciones (B: BaseDatos; T: Tabla; A: Atributo; R: TablaRestricciones, I: IdTupla): Booleano

Esta función chequea, para cada restricción que debe cumplir la celda (base de datos, tabla, atributo e identificación del registro) si cumple con las restricciones de dominio establecidas en una tabla de restricciones (parámetro R) de la base de datos.

Las celdas que registran blancos o ceros y aunque sea una opción válida para la tabla de restricciones, se tomarán como que no cumplen la restricción. Para poder evaluar los atributos en blanco o ceros, para los tipos de atributo alfanumérico y numérico respectivamente, utilizaremos las dos rutinas genéricas siguientes:

Function Chequear_ Blanco (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano definida en II.6 o
Function Chequear_Cero (B: BaseDatos, T: Tabla, A: Atributo, I: IdTupla): Booleano definida en II.7.

Este control se realiza, ya que los valores cero o blanco fueron agregados a las tablas de códigos como opciones válidas, para los nuevos atributos agregados, sea por reglamentación o reglas del negocio, tal como se presenta en la sección 4.3.

II.10 Síntesis

A continuación se un cuadro resumen de las rutinas genéricas, con sus parámetros de entrada y salida.

Función	Parámetros de entrada	Salida
Chequear_Nulo	- B: BaseDatos - T: Tabla - A: Atributo - I: IdTupla	**Booleano**
Comparar_Atributos	- BM: BaseDatos - B: BaseDatos - T1: Tabla - T2: Tabla - I: IdTupla	**Cantidad de atributos iguales**
Calcular_Distancia	- X: String - Z: String	**Distancia**
Chequear_Regla_Dominio	- B: BaseDatos - T: Tabla - A: Atributo - R: Tabla - I: IdTupla	**Booleano**
Chequear_Regla_Atributo	- B: BaseDatos - T: Tabla - A: Atributo - R: Tabla - I: IdTupla	**Booleano**
Chequear_Blanco	- B: BaseDatos - T: Tabla - A: Atributo - I: IdTupla	**Booleano**
Chequar_Cero	- B: BaseDatos - T: Tabla - A: Atributo - I: IdTupla	**Booleano**
Analizar_Trazas	- B: BaseDatos - T1: Tabla - T2: Tabla - I: IdTupla - F: Fecha	**Booleano**
Chequear_restricciones	- **B:BaseDatos** - **T: Tabla** - **A: Atributo** - **R: Tabla** - **I: IdTupla**	**Booleano**

Tabla 36- Rutinas Genéricas

Anexo III. Descripción de las entidades del Modelo Lógico de los metadatos

A continuación se extiende la información de la descripción de las tablas del modelo lógico introducidas en la sub-sección 6.1.

Tabla: MetricaInstanciada		
Atributo	**Clave Primaria**	**Descripción**
MINSMetrica	Si	**Es el identificador de la métrica instanciada, por ejemplo: *Edad-Promedio de tiempo en días*.**
MINSFactor	No	**Factor instanciado a ser medido, por ejemplo: *Edad*. Es clave foránea de la tabla FactorInstanciado.**
METMetrica	No	**Nombre de la métrica genérica. Por ejemplo: Edad. Es clave foránea de la tabla Metrica.**
MINSMetricaDescripción	**No**	**Breve descripción de la métrica. Por ejemplo: *mide la antigüedad de un dato, o sea el tiempo transcurrido desde que fue modificado por última vez hasta la fecha de medición (now)*.**

Tabla 37- Atributos de la tabla MetricaInstanciada

Tabla: MetodoInstanciado		
Atributo	**Clave Primaria**	**Descripción**
METMetodo	Si	**Identificador del método, por ejemplo: *Calcular_Edad*.**
MINSMetrica	No	**Es el identificador de la métrica instanciada. Es clave foránea de la tabla MetricaInstanciada.**
MetMetodo	No	**Método genérico. Es clave foránea de la tabla Metodo.**
METMetodoDescripción	**No**	**Breve descripción del método. Por ejemplo: *calcula la edad de cada registro y realiza la agregación.***

Tabla 38- Atributos de la tabla MetodoInstanciado

Tabla: MetodoParametro		
Atributo	**Clave Primaria**	**Descripción**
METParametro	Si	**Es el nombre del parámetro a ser recibido por el método, por ejemplo: *BaseDatos*.**
METMetodo	Si	**Es el identificador del método. Tiene clave foránea con la tabla Metodo.**
METOrdenParametro	No	**Es el orden en el que el método recibe el parámetro, por ejemplo: el parámetro *BaseDatos*, será recibido en el lugar *1*.**
METTipoParametro	**No**	**Es el tipo del parámetro a ser recibido.**

Tabla 39- Atributos de la tabla MetodoParametro

Tabla: Rutina		
Atributo	**Clave Primaria**	**Descripción**
RUTRutina	Si	**Identificación de la rutina.**
RUTDescripcion	**No**	**Descripción de la rutina.**

Tabla 40- Atributos de la tabla Rutina

Tabla: RutinaParametro		
Atributo	**Clave Primaria**	**Descripción**
RUTParametro	Si	**Es el nombre del parámetro a ser recibido por la rutina, por ejemplo: *BaseDatos*.**
RUTRutina	Si	**Identificación de la rutina. Hereda este atributo de la tabla Rutina**
RPAROrdenParametro	No	**Es el orden en el que la función recibe el parámetro, por ejemplo: el parámetro *BaseDatos*, será recibido en el lugar *1*.**
RPARTipoParametro	No	**Es el tipo del parámetro a ser recibido.**
RPAREntradaSalida	**No**	**Indica si el parámetro es de E-entrada o de S-salida.**

Tabla 41- Atributos de la tabla RutinaParametro

Tabla: MedidaResultadoTabla		
Atributo	**Clave**	**Descripción**

Tabla: MedidaResultadoTabla		
	Primaria	
MRTABIdTabla	Si	**Identificación de la tabla instanciada.**
MRTABFechaEjecucion	Si	**Fecha y hora de la ejecución de la medición.**
MINSMetodo	Si	**Identificador del método instanciado. Es clave foránea de la tabla MetodoInstanciado.**
MRTABValor	**No**	**Valor obtenido de la medición.**

Tabla 42- Atributos de la tabla MedidaResultadoTabla

Tabla: MedidaResultadoCelda		
Atributo	**Clave Primaria**	**Descripción**
MRCELIdTabla	Si	**Identificador de la tabla instanciada**
MRCELIdTupla	Si	**Identificador de la tupla instanciada**
MRCELIdAtributo	Si	**Identificador del atributo instanciado**
MRCELFechaEjecucion	Si	**Fecha y hora de la ejecución de la medición.**
MINSMetodo	Si	**Identificador del método instanciado. Es clave foránea de la tabla MetodoInstanciado.**
MRCELValor	**No**	**Valor de la medición.**

Tabla 43- Atributos de la tabla MedidaResultadoCelda

Tabla: MedidaResultadoTupla		
Atributo	**Clave Primaria**	**Descripción**
MRTUPIdTabla	Si	**Identificador de la tabla instanciada**
MRTUPIdTupla	Si	**Identificador de la tupla instanciada**
MRCELFechaEjecucion	Si	**Fecha y hora de la ejecución de la medición.**
MINSMetodo	Si	**Identificador del método instanciado. Es clave foránea de la tabla MetodoInstanciado.**
MRTUPValor	**No**	**Valor de la medición.**

Tabla 44- Atributos de la tabla MedidaResultadoTupla

Tabla: MedidaResultadoAtributo		
Atributo	**Clave Primaria**	**Descripción**

Tabla: MedidaResultadoAtributo		
MRATRIdTabla	Si	**Identificador de la tabla instanciada**
MRATRIdAtributo	Si	**Identificador del atributo instanciado**
MRATRFechaEjecucion	Si	**Fecha y hora de la ejecución de la medición.**
MINSMetodo	Si	**Identificador del método instanciado. Es clave foránea de la tabla MetodoInstanciado.**
MRATRValor	**No**	**Valor de la medición.**

Tabla 45- Atributos de la tabla MedidaResultadoAtributo

Tabla: MedidaResultadoArea		
Atributo	**Clave Primaria**	**Descripción**
MRAREIdTabla	Si	**Identificador de la tabla instanciada.**
MRAREIdArea	Si	**Identificador del área instanciada.**
MRAREFechaEjecucion	Si	**Fecha y hora de la ejecución de la medición.**
MINSMetodo	Si	**Identificador del método instanciado. Es clave foránea de la tabla MetodoInstanciado.**
MRAREValor	**No**	**Valor de la medición.**

Tabla 46- Atributos de la tabla MedidaResultadoArea

Anexo IV. Notación de la herramienta ERStudio

La notación de los esquemas de base de datos construidos utilizando la herramienta ER/Studio versión 6.6 para esta tesis es la siguiente:

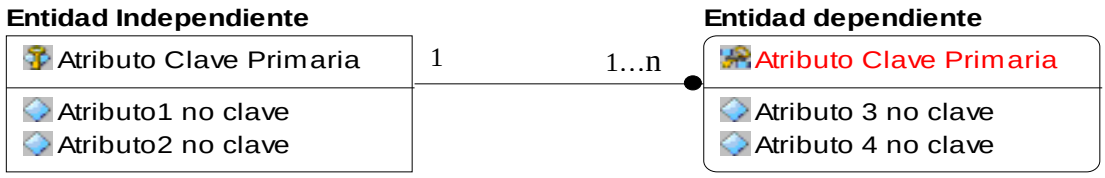

Figura 20- Relación de Identidad

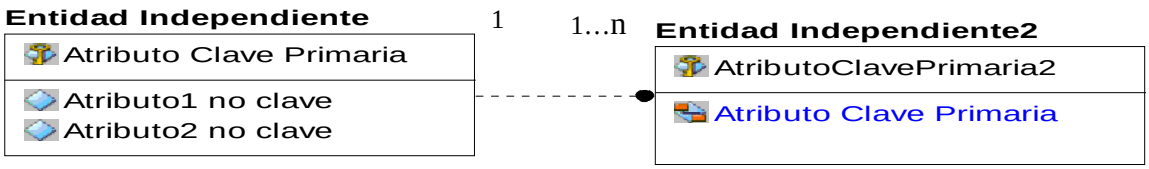

Figura 21- Relación de no identidad

Aclaraciones:

- La clave que es propia se identifica con una llave vertical, y el texto aparece en color Negro.
- La clave que es heredada, se identifica con una llave inclinada, y el texto aparece en rojo
- Las claves foráneas, se identifican con color azul.

Bibliografía específica

[Akoka+2007] Akoka, L.; Berti-Équille,O.; Boucelma, M.; Buzeghoub, M.; Comyn-Wattiau, I.; Cosquer, M.; Goasdoué-Thion, V.; Kedad, Z.; Nugier, S.; Peralta, V.; Sisaid-Cherfi, S.; : "A Framework for Quality Evaluation in Data Integration Systems", 9th International Conference on Enterprise Information Systems (ICEIS'2007), Funchal, Portugal, June 2007.

[Ballou+1985] Ballou, D.; Pazer, H.: "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems ", Management Science, Vol. 31, No. 2, Febrero 1985.

[Basili+1994] Basili, V.; Caldiera, G.; Rombach, H.: "The Goal Question Metric Approach", Encyclopedia of Software Engineering, 528-532" John Wiley & Sons, Inc, 1994.

[Batini+2006] Batini, C.; Scannapieca, M.: "Data Quality. Concepts, Methodologies an Technologies", 2006.

[Bright+2002] Bright, L.; Raschid, L.: "Using Latency-Recency Profiles for Data Delivery on the Web". In Proc. of the 28th Int. Conf. on Very Large Databases (VLDB'02), Hong Kong, China, 2002.

[Chiruzzo+2007] Chiruzzo, L.; Pais, C.: "Calidad en Microarrays", Proyecto de Grado, Facultad de Ingeniería, Universidad de la República, 2007.

[Cho+2000] Cho, J.; Garcia-Molina, H.: "Synchronizing a database to improve freshness". In Proc. of the 2000 ACM Int. Conf. on Management of Data (SIGMOD'00), pages 117-128, Dallas, USA, 2000.

[Clément+2007] Clément, D.; Laboisse, B.: "Création d'un référentiel d'indicateurs de mesure de la qualité des dones CRM", EGC Namur, 2007.

[Etcheverry+2007] Etcheverry, L.; Tercia, S; Marotta, A; Peralta, V.: "Medición de la Exactitud de Datos en Sistemas Fuentes: Un Caso de Estudio", CSI, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, abril de 2007.

[Etcheverry+2008] Etcheverry, L; Peralta, V; Bouzeghoub, M.: "Qbox-Foundation: a Metadata Platform for Quality Measurement", QDC, Nice, France, 2008.

[Fugini+2002] Fugini, M.; Mecella, M.; Plebani, P.; Pernici, B.; Scannapieco, M.: "Data quality in cooperative web information systems". Technical Report, 2002.

[Gançarski+2003] Gançarski, S.; Le Pape, C.; Valduriez, P.: "Relaxing Freshness to Improve Load Balancing in a Cluster of Autonomous Replicated Databases". In Proc. of the 5th workshop on Distributed Data and Structures (WDAS), Thessaloniki, Greece, 2003.

[Gertz+2004] Gertz, M.; Ozzu, M.; Saake, G.; Sattler, K.: "Report on the Dagstuhl Seminar: Data Quality on the Web". SIGMOD Record Vol. 33(1), March 2004.

[Goerk2004] Goerk, M. : "An Enterprise Wide Approach to Data Quality Goals", CAiSE Workshop on Data and Infomation Quality, 2004.

[Green2007] Green, B.: "Information Management Standards and Data Quality Thematic Briefing Paper (May 2007). Europe's one-stop shop on Public Sector Information re-use." URL: www.epsiplus.net.

[Hammer+1995] Hammer, J.; Garcia-Molina, H.; Widom, J.; Labio, W.;, Zhuge., Y.: " The Stanford Data Warehousing Project" , 1995.

[Jarke+1997] Jarke, M.; Vassiliou, Y.: "Data Warehouse Quality: A Review of the DWQ Project". In Proc. 2nd Conference on Information Quality (IQ'1997), Cambridge, USA, 1997.

[Jarke+1999] Jarke, M.; Jeusfeld, M.A.; Quix, C.; Vassiliadis, P.: "Architecture and Quality in Data Warehouses: An Extended Repository Approach". Information Systems, vol. 24(3), 1999.

[Kon+1995] Kon, H.; Madnick, S.; Siegel, M.: "Good Answers from Bad Data:a Data Management Strategy". Working paper 3868-95, Sloan School of Management, Massachusetts Institute of Technology, USA, 1995.

[Laboisse2005] Laboisse, B.: "BDQS, une approche dans la mesure de la qualité de données d'un CRM: principes de base (les attributs de la qualité), intégration des outils métier de marketing direct dans la mesure". Séminaire CRM & Qualité des Données, Paris, France, 2005.

[Monge+1997] Monge, A; Elkan, C.: "An efficient domain-independent algorithm for detecting approximately uplicate database records". In Proceedings of the SIGMOD 1997 Workshop on Research Issues on Data Mining and Knowledge Discovery, Tuscon, AZ, May 1997.

[Motro+1998] Motro, A.; Rakov, I.: "Estimating the quality of databases". In Proc of the 3rd Int. Conf on Flexible Query Answering Systems (FQAS'98), Roskilde, Denmark, 1998.

[Naumann+1999] Naumann, F.; Leser, U.; Freytag, J.: "Quality-driven Integration of Heterogeneous Information Systems" 25th VLDB Conference, Edinburgh, Scotland, 1999.

[Naumann+2003] Naumann, F.; Freytag, J., Leser, U.: "Completeness of Information Source", 2003.

[Office2006] Office of Management and Budget. Information Quality Guidelines for Ensuring and Maximizing the Quality, Objectivity, Utility, and Integrity of Information Disseminated by Agencies. http://www.whitehouse.gov/omb/fedreg/reproducible.html. Último acceso en el primer semestre 2008.

[ORACLE2008] http://www.oracle.com/solutions/business_intelligence/index.html. Último acceso en el primer semestre del 2008.

[Pan2008] Pan, Y.: "SQL Server Audit in SQL Server 2008 – Part 1" Database Journal, 6 de junio 2008.

[Peralta+2004] Peralta, V.; Ruggia, R.; Bouzegoub, M.: "Analyzing and Evaluating Data Freshness in Data Integration Systems". Ingénierie de Systèmes d'Information (ISI), vol. 9 (5/6), 145-162, 2004.

[Peralta2006] Peralta, V.: "Data Quality Evaluation in Data Integration Systems". PhD Thesis, Université de Versailles – France and Universidad de la República – Uruguay, 2006.

[Pipino+2002] Pipino, L.L.; Lee, Y.W.; Wang, R.: "Data Quality Assessment". Communications of the ACM, vol. 45, No. 4, April 2002.

[Redman1996] Redman, T.: "Data Quality for the Information Age". Artech House, 1996.

[Scannapieco+2004] Scannapieco, M; Missier, P; Batini, C.: "Data Quality at a Glance", 2004.

[Segev+1990] Segev, A.; Weiping, F.: "Currency-Based Updates to Distributed Materialized Views". In Proc. Of the 6th Int. Conf. on Data Engineering (ICDE'90), Los Angeles, USA, 1990.

[Shanks+1999] Shanks, G.; Corbitt, B.: "Understanding Data Quality: Social and Cultural Aspects". In Proc. Of the 10th Australasian Conference on Information Systems, Wellington, New Zealand, 1999.

[Torres2006] Padrón Torres, L.: "Similitud entre cadenas y estandarización de direcciones postales", 2006.

[Wang+1996] Wang, R.; Strong, D.: "Beyond accuracy: What data quality means to data consumers". Journal on Management of Information Systems, Vol. 12 (4):5-34, 1996.

[Wipikedia2008] Wikipedia, URL: www.wikipedia.org. Último acceso en el primer semestre de 2008.