Tesina para la Especialización en Sistemas de Información Avanzados

Tema: “Un Marco de Trabajo para Analizar y Mejorar la Calidad de Datos dentro de su Ciclo de Vida”

UNCOMA

Tesista: Lic. Gonzalo Ernesto Domingo

Directora: Mg. Agustina Buccella

Co-Directora: Dra. Alejandra Cechich

Datos Personales:

Nombre: Gonzalo Domingo

e-mail: gonzalodomingo2@hotmail.com

Todos los imperios del futuro van a ser imperios del conocimiento, y solamente serán exitosos los pueblos que entiendan cómo generar conocimientos y cómo protegerlos; cómo buscar a los jóvenes que tengan la capacidad para hacerlo y asegurarse que se queden en el país.

Albert Einstein

Contexto

Mediando el año 2005, producto de un análisis FODA que realizamos en nuestro grupo de trabajo, se observó que las aplicaciones que se utilizan como solución de problemáticas en el negocio, estaban con una debilidad que era el ser propensos a permitir que los datos ingresantes sean de baja calidad. Observábamos que nuestros sistemas son permeables a los datos de mala calidad.

Decidimos entonces transformar esta debilidad en una oportunidad de mejora, nos planteamos como objetivo para mediados del 2006 pudiéramos tener una propuesta de solución a este problema, para lo cual pusimos a tres personas a investigar el tema, nos preguntábamos: ¿existe alguna guía de recomendaciones prácticas con pautas a incluir en los sistemas para disminuir esta permeabilidad a la mala calidad?

Buscamos durante un par de meses y al no encontrarla, nos decidimos a desarrollar un marco de trabajo, basado en recomendaciones dispersas en diferentes bibliografías y en la observación directa de los sistemas que conocemos y cuyo ámbito nos resulta familiar por el trabajo cotidiano.

Investigamos y recopilamos biografía desde ese momento y hasta marzo del 2006. Luego presentamos el tema a todo el grupo de trabajo con lo recopilado hasta el momento, hicimos una tormenta de ideas de una jornada laboral para tipificar estas prácticas.

Finalmente confrontamos este marco de trabajo contra un producto de software desarrollado internamente. De esta forma bajamos a terreno práctico las ideas de mejora. Para formalizar esto se escribió un procedimiento y se modifico el proceso de desarrollo, que existía previamente.

El siguiente trabajo es el resultado de estos meses de búsquedas y análisis.

Resumen del trabajo

La palabra **calidad** tiene múltiples significados:

- Es la percepción que el cliente tiene del un producto o servicio.
- Es una fijación mental del consumidor que asume conformidad con un producto o servicio determinado, que solo permanece hasta el punto de necesitar nuevas especificaciones.
- Conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.
- Es cuando un producto o servicio satisface las necesidades del cliente o usuario.

Para este trabajo, tomamos la definición de calidad del modelo FUNDIBQ (Fundación Iberoamericana de la Calidad) que establece: *"La calidad es el conjunto de características propias de un producto, servicio, sistema o proceso imprescindibles para cumplir las necesidades o expectativas de partes interesadas"*, es decir que la calidad es un punto de acuerdo entre las partes interesadas.

En los sistemas que se desarrollan en la empresa, la calidad del dato no es analizada ni tenida en cuenta en el momento de diseñar una aplicación. Apenas se hacen esfuerzos por analizar los datos cuando se descubre que los mismos no coinciden con la realidad.

El presente trabajo, describe una metodología de trabajo, orientada a pensar los sistemas desde la óptica de la calidad de los datos desde el momento del relevamiento y hasta la puesta en producción. Aplicando prácticas que resultan en aumentar la calidad de los datos que estarán en las aplicaciones.

Se busca con esto evitar los efectos de la baja calidad de datos que son: *clientes insatisfechos, empleados insatisfechos, toma de Decisiones erróneas* y así aprovechar los beneficios de una buena calidad de datos se ven reflejados en la empresa o empresas que hacen uso de esos datos. Estos son: *mejora en el soporte a la toma de decisiones, Reducción del tiempo necesario para obtener un informe, sustitución de actividades de bajo valor por otras de mayor valor, mejora de la imagen de la empresa.*

Índice

Capítulo 1 Introducción

En este capítulo se presenta una breve descripción sobre el contexto de la tesis, su objetivo y su estructura.

1.1 Marco Teórico

El término Calidad de Datos posee varias definiciones en la literatura [7,2,6], pero todas convergen en que el concepto calidad del dato es *relativo al uso del dato* [7]. Esto implica que este concepto es relativo, datos considerados con calidad para cierto uso pueden considerarse con insuficiente calidad para otros usos. A su vez, en [2] se define a la calidad como un concepto multidimensional ya que el mismo se evalúa teniendo en cuenta varias dimensiones que analizaremos más adelante.

Siguiendo con la definición, frases como:

- *Basura adentro, basura afuera*
- *Si se ingresa información inexacta, se obtendría información inexacta*
- *Pagar ahora o pagar mas tarde más*

Son muy comunes dentro del ámbito de calidad del dato. La primera se refiere a que si la calidad de nuestros datos es mala (poseemos basura), el resultado que obtendremos de la utilización de esos datos también será basura. La segunda se desprende de la primera, si la información almacenada es inexacta, eso es lo que obtendremos, inexactitud. Por último, la frase *pagar ahora o pagar mas tarde más* se refiere a que nuestra empresa puede comenzar proyectos que incluyan actividades para prevenir y limpiar los datos almacenados en las bases de datos ahora, o puede esperar unos meses más y realizar tareas mucho mas costosas para estos proyectos. Esto se debe a que a medida que el tiempo pasa, si no se tiene en cuenta la calidad del dato, la misma va disminuyendo poco a poco. Es decir, así como la información va aumentando, la calidad va disminuyendo.

1.2 Objetivos

1.2.1. **Objetivo General**: Definir un marco de trabajo general analizando el ciclo de vida del dato que permita la evaluación y mejora de la calidad del dato.

1.2.2. **Objetivos Específicos**:

- Relevar las técnicas que favorecen la calidad de los datos durante su ciclo de vida.
- Clasificar las técnicas referidas en el punto anterior en recomendaciones, buenas prácticas o estándares aplicables a la empresa.
- Realizar un marco conceptual que permita pensar, diseñar, construir y probar los sistemas teniendo en cuenta prácticas y estándares que favorezcan la calidad de los datos.
- Aplicar el marco conceptual definido a un sistema real dentro de una empresa.

1.3 Estructura del Trabajo

En el capítulo 2 se expresa la propuesta y las definiciones que componen el marco de trabajo. A su vez, se definen las líneas fundamentales del marco de trabajo que surgieron de las actividades propuestas para mejorar los aspectos relacionados con la calidad del dato en los sistemas de información.

En este capítulo se definen las cuatro dimensiones que se utilizan para evaluar la calidad del dato, el ciclo de vida del mismo y se explicitan los beneficios desde el punto de vista conceptual de la calidad del dato en si misma y para la empresa donde se desarrollo este trabajo. Estos beneficios fueron utilizados como justificación del mismo.

En el capítulo 3 se describe detalladamente la lista de pautas para mejorar los sistemas teniendo en cuenta la calidad del dato en sus cuatro dimensiones y agrupando dichas pautas en el ciclo de vida del dato. A su vez explicamos como se clasificaron las diferentes prácticas de acuerdo a los criterios de estas cuatro dimensiones de calidad. Los procesos existentes fueron modificados para adaptarlos a una nueva forma de trabajo y se plantearon flujos formales para que el mismo pueda ser aplicado.

Por último, en el capítulo 4, abordamos la problemática de llevar a la práctica los conceptos vistos y cómo llevar adelante un proceso de puesta en marcha de un proyecto de software pensando en la calidad de los datos. De

esta manera, se presenta un caso de estudio en donde se explica la forma en que se analizó una aplicación utilizando este marco de trabajo.

Finalmente se exponen las conclusiones del presente trabajo.

Capítulo 2

En este capítulo se trazan las directrices del marco de trabajo que se ha definido, se precisan las dimensiones en las que se evaluará la calidad, se define el ciclo de vida del dato y se compara este trabajo con otros trabajos relacionados.

A su vez se detallan los beneficios desde el punto de vista conceptual de la calidad del dato tanto en si misma como para la empresa en general y en especial para la empresa en la cual se desarrollo este trabajo. Estos beneficios fueron utilizados como justificación de la elaboración e implementación del proyecto ante la empresa donde fue aplicado.

2.1 Cuatro Dimensiones para Medir Calidad

Según Brackstone [1] la calidad del dato puede ser medida de acuerdo a cuatro dimensiones. Estas serán explicadas a continuación:

- *Exactitud*: ¿Representan los datos exactamente la realidad o fuentes verificables?. La exactitud del dato esta relacionada con su fuente; es decir, el nivel de correspondencia entre el dato y el mundo real. Por ejemplo, si una base de datos almacena información sobre stock, y en la misma se encuentra registrado que existen diez computadoras almacenadas; efectivamente (o físicamente) deben existir esas diez computadoras en dicho almacén.

- *Completitud*: ¿Todos los datos necesarios están presentes?. Qué cantidad de datos no están presentes?. Esta dimensión se refiere a los datos necesarios que debe contener un sistema de información. Por ejemplo, un cliente de un banco necesitará conocer el saldo de sus cuentas bancarias. Si esta no existiese, tendríamos un problema en la completitud del dato, además de la insatisfacción del cliente.

- *Consistencia*: ¿Los datos fueron consistentemente definidos y entendidos?. Requiere disciplina en el tiempo. Se refiere a la definición de estándares y protocolos para los datos. Todos los datos

se representan en un formato compatible, que además es el más adecuado para la tarea que se está desarrollando. No es lo mismo que el género de una persona se almacene como “F" o como “Femenino" o “Fem". Debe definirse una forma (o estructura) común de almacenamiento de los datos.

- *Temporalidad*: ¿Los datos están disponibles cuando se necesitan?. Por ejemplo, los datos ¿están disponibles cuando se deben tomar decisiones?. Es sabido que los datos en una empresa, además de otras finalidades, existen para la toma de decisiones. Si los datos no están disponibles o son erróneos cuando se debe tomar una decisión, se podrán generar grandes perjuicios. Dentro de esta dimensión se enmarca el concepto de *Volatilidad* el cual se refiere a la cantidad de tiempo que el dato se mantiene válido. Por ejemplo la dirección de una persona es un dato volátil ya que puede cambiar en cualquier momento, mientras que el número de documento de una persona es un dato no volátil.

Teniendo en cuenta estas cuatro dimensiones, es importante resaltar que una mala calidad de datos realmente daña el negocio ya que inevitablemente va a interferir tanto en los procesos que suceden dentro de una empresa como en los procesos que suceden entre la empresa y sus clientes. Así, puede malgastar los recursos asignados al marketing dañando la imagen y reputación de la empresa. En la literatura [5] se han detallado impactos negativos de una mala calidad de datos. Algunos de ellos son:

- *Clientes insatisfechos*: sus datos personales, sus pedidos o sus facturas no son correctas.

- *Empleados insatisfechos*: cometen errores o no conocen cierta información, lo que los hace cometer a su vez más errores.

- *Toma de Decisiones erróneas*: los datos usados por los gerentes también pueden tener errores y es sabido que las decisiones no van a ser mejores que los datos en los que están basadas.

La empresa donde desarrollamos el presente trabajo se enmarca en el mercado energético operando campos petroleros y gasíferos de Argentina y con actividad en otros 29 países. Por cuestiones de confidencialidad no divul-

garemos su nombre. La misma, adopta la definición de calidad del modelo FUNDIBQ (Fundación Iberoamericana de la Calidad) que establece: *"La calidad es el conjunto de características propias de un producto, servicio, sistema o proceso imprescindibles para cumplir las necesidades o expectativas de partes interesadas"*, es decir que la calidad es un punto de acuerdo entre las partes interesadas.

2.2 Beneficios para la Empresa

Los principales beneficios de una buena calidad de datos se ven reflejados en la empresa o empresas que hacen uso de esos datos. A continuación se detallan algunos de estos beneficios junto con los impactos que generan.

- **Mejora en el soporte a la toma de decisiones** [11]: Para tomar decisiones hay que basarse en la frialdad y objetividad de los datos, mas que intuiciones, deseos y/o esperanzas. Las decisiones acertadas, se basan en datos objetivos y fiables. Se plantea encontrar un método que mejore la calidad de los datos y así mejorar la calidad de la información obtenida. Con buena información, se pueden hacer estudios a futuro, y obtener mejoras a corto plazo.
- **Reducción del tiempo necesario para obtener un informe**: Al poseer datos confiables es posible reducir el costo hora/hombre invertido en la elaboración de los mismos.
- **Sustitución de actividades de bajo valor por otras de mayor valor**: Teniendo datos confiables se puede destinar más recursos al análisis de la información y la toma de decisiones ya que se reduce el esfuerzo en la recolección de los mismos y su comprobación.
- **Ley Sarbanes-Oxley** [14]: Esta ley, cuyo cumplimiento es exigencia para las empresas que cotizan en la bolsa de Nueva York, determina que los datos deben ser traceables (La trazabilidad es la capacidad de recorrer el camino de los datos que generan información). Está ley fue implementada para asegurar la transparencia de la información que afecta los estados contables de la empresa, abriéndose así una oportunidad para invertir en calidad de los datos.
- **Mejora de la imagen de la empresa** [5]: Una empresa que toma decisiones basada en datos erróneos indefectiblemente tendrá un

impacto negativo en la imagen de quienes se relacionen con ella. Por lo tanto esta iniciativa va asociada al incremento en la satisfacción del cliente tanto interno como externo.

2.3 El Ciclo de Vida del Dato

En Redman [6] se describe como el ciclo de vida del dato el cual se compone por cuatro etapas fundamentales: *modelado del dato*, *captura del valor*, *almacenamiento* y *visualización*. La Figura 2.1 muestra gráficamente estas cuatro etapas.

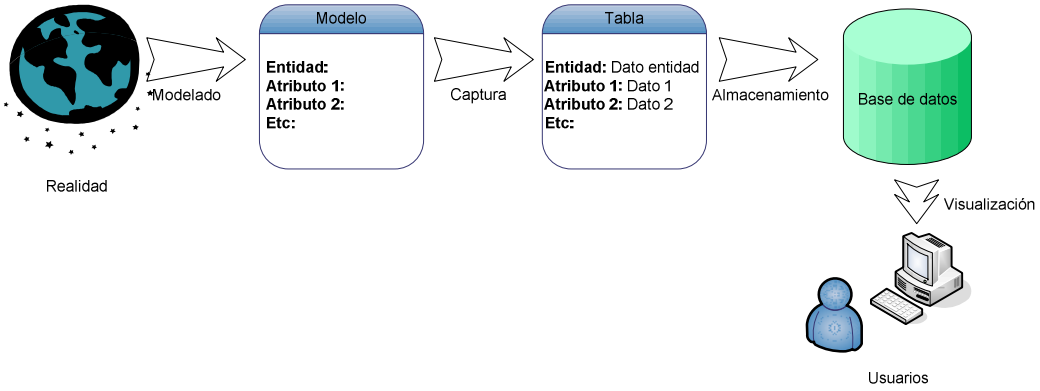

Figura 2.1 Ciclo de vida del dato.

2.3.1 Modelado del Dato

Una vez que han sido relevados los requerimientos o necesidades del cliente, se elabora una abstracción que represente la realidad. Esta constituye un modelo lógico donde se establecen que datos se tomaran y como fluirá la información por la aplicación y los roles de los actores que interactúan con ella.

En [16] se define un modelo de datos como un *sistema formal* y *abstracto* que permite describir los datos de acuerdo a reglas y convenios predefinidos. Es *formal* pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar. Y es *abstracto* porque existe una operación mental por la que las propiedades de los objetos se aíslan conceptualmente, a efectos de reflexionar sobre ellas sin tomar en consideración otros rasgos que momentáneamente se desea ignorar.

Esta etapa del ciclo de vida consiste en tomar la realidad y darle un marco lógico que permita describirla como un modelo de datos. Se define el contenedor de datos, así como los métodos para almacenar y recuperar información de este contenedor. Un modelo de datos consiste en los objetos, o entidades que existen y se manipulan; en los atributos que son características básicas de estos objetos y relaciones o forma en que enlazan los distintos objetos entre si.

2.3.2 Captura del Valor

La información se referencia a través de interfaces con otros sistemas o se recopila a través de una interface de usuario. En esta etapa el dato es tomado de la realidad y representado dentro del contenedor que se haya diseñado.

En este punto el dato está dentro del sistema, representando la realidad de acuerdo a su diseño. Es decir, es cuando los objetos poseen los datos y los atributos toman valores. Es importante aclarar que las distintas etapas del ciclo de vida del dato van heredando los defectos de las etapas anteriores, si hemos hecho un modelado del dato pobre, la captura sufrirá sin duda de falencias y el dato almacenado será distinto de la realidad.

En esta etapa, se deben contemplar decisiones del tipo de interface de carga de datos como qué datos se cargarán y cuáles se adquirirán de otras fuentes, controles referidos a la carga (máscaras, fechas, validaciones o reglas), se debe decidir si la carga será manual o automatizada, etc.

2.3.3 Almacenamiento

En esta etapa del ciclo de vida, el dato pasa de la interface de captura al repositorio de datos donde quedará almacenado, normalmente una base de datos.

Una base de datos es un conjunto de datos que pertenecen al mismo contexto y que se encuentran almacenados para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y gracias al desarrollo tecnológico la mayoría de las ba-

ses de datos tienen formato electrónico, esto ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen en la actualidad varias maneras de clasificar las bases de datos, dependiendo del criterio elegido para su agrupación:

Según la variabilidad de los datos almacenados:

- *Bases de datos estáticas*: Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- *Bases de datos dinámicas*: Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos; además de las operaciones fundamentales de consulta.

Según el contenido:

- *Bases de datos bibliográficas*: Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo.

- *Bases de datos de texto completo*: Almacenan la información completa de un registro fuente, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

Las bases de datos están basadas en modelos de datos definidos especialmente para la implementación de las mismas. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema

eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos utilizados con frecuencia en las bases de datos:

- *Bases de datos jerárquicas*: Son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en forma similar a un árbol, en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz y los nodos que no tienen hijos se los conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos compartidos, ya que permiten crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- *Bases de datos de red*: Es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Las bases de datos de red son una gran mejora con respecto al modelo jerárquico, ya que ofrecen una solución eficiente al problema de redundancia de datos. Pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- *Base de datos relacional*: Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. El modelo fue enunciado en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California) y no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de *"relaciones"*. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas. La mayoría de las veces se conceptualiza de una manera más sencilla, pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas

de una tabla), que representarían las tuplas, y campos (las columnas de una tabla). En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la ventaja de que es más fácil de entender y de utilizar para un usuario de la base de datos. La información es recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas en bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas. Es un estándar implementado por los principales o sistemas de gestión de bases de datos relacionales (SGBD). Por ejemplo el modelo SQL:1999 o SQL3 establecido por ANSI e ISO, agrega al modelo anterior del estándar expresiones regulares, consultas recursivas para relaciones jerárquicas, triggers y algunas características orientadas a objetos.

- *Bases de datos orientadas a objetos*: Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma orientado a objetos:

    - *Encapsulación* - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
    - *Herencia* - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
    - *Polimorfismo* - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

  En las bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación se especifica mediante dos partes: la *interface* (o signatura) de una operación que incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros); y la *implementación* de la operación que se especifica en forma separada y puede modificarse sin afectar a los demás objetos que hagan uso de la operación. Esto se denomina

encapsulación de la información. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

- *Bases de datos documentales*: Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.

- *Base de datos deductivas*: Un sistema de base de datos deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que se almacenan en la base de datos. También las bases de datos deductivas son llamadas base de datos lógicas, a raíz de que se basan en la lógica matemática.

2.3.4 Visualización:

Esta etapa se refiere a la presentación de los datos al usuario. Es decir, los datos se presentan al usuario asistiendo así a la compresión e identificación de errores o inconsistencias. De acuerdo a recomendaciones existentes en el proceso de desarrollo de la empresa en donde se desarrolló este trabajo, la visualización debe basarse en cinco atributos:

- *Navegación*: Es uno de los aspectos más importantes en el diseño de interfaces de visualización, ya que es el que nos permite movernos a través de las diferentes pantallas. *Tiempo de respuesta*: Se define generalmente como el tiempo que necesita el sistema para expresar los cambios de estado del usuario. Esta característica es difícil de parametrizar debido a la enorme diversidad de velocidades computacionales de los distintos dispositivos y velocidades de transmisión de datos. A pesar de estas connotaciones tecnológicas, es importante hacer consideraciones acerca de intentar que los tiempos de respuesta sean soportables para el usuario.
- *Contenido*: Es la pieza fundamental de una interface de visualización. Se refiere al conjunto de información que el usuario puede encontrar en la misma. *Interactividad*: Característica que incrementa, cualitativa y cuantitativamente, la capacidad de los usua-

rios de intervenir en el desarrollo de las posibilidades que ofrecen las aplicaciones, mejorando así su trabajo y aprendizaje.

- *Facilidad de comprensión*: Se refiere a que el lenguaje de la interface de visualización está optimizado para hacer más sencillo que el usuario sea capaz de interpretar, retener, organizar y valorar la información que recibe.

La construcción de la interface de visualización del dato debe tener en cuenta los siguientes parámetros:

- *Facilidad de Aprendizaje*: se refiere a que nuevos usuarios pueden tener una interacción efectiva. Está relacionada con la predicibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.

- *Flexibilidad***:** hace referencia a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, la similitud con tareas anteriores y la optimización entre el usuario y el sistema.

- *Robustez***:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de la forma en que la tarea se adapta al usuario.

Las buenas prácticas en esta etapa del ciclo de vida del dato reducen los costos de aprendizaje, asistencia al usuario, optimizan los costos de rediseño y mantenimiento de las aplicaciones, mejoran la imagen y el prestigio de la aplicación, mejoran la calidad de vida de los usuarios ya que reduce su estrés, incrementa la satisfacción y la productividad. Todos estos beneficios implican una reducción y optimización general de los costos de producción, así como un aumento en la productividad de las aplicaciones.

Una perspectiva centrada en el usuario se basa en contenidos claros y transparentes, dejando al usuario el control de la navegación y facilitándosela en lo posible.

2.4 Trabajos Relacionados

Para la elaboración este trabajo, hemos abordado previamente tres de las propuestas [3, 4, 8] mas significativas que se basan en la evaluación y análisis de la Calidad del Dato en de los Sistemas de Información.

2.4.1 Calidad de datos basada en su uso

En la propuesta de Ken Orr [3] el concepto de calidad de datos se basa en el uso del mismo y se establecen seis reglas para la calidad del dato:

1. Los datos que no son usados no se mantienen correctos por mucho tiempo.
2. La calidad de los datos en un sistema de información esta en función de su uso, no de su obtención.
3. La calidad de los datos no será mejor que su uso más riguroso.
4. La calidad de los datos tienden a volverse peores con el paso del tiempo.
5. Cuanto menos probable sea que un atributo del dato cambie, más traumático será cuando finalmente cambie.
6. Las reglas de calidad de datos se aplican tanto a los datos como a los metadatos (datos sobre los datos).

Según Orr, en la práctica estas reglas no se aplican en la mayoría de los sistemas. Es normal que las aplicaciones guarden gran cantidad de datos sólo teniendo en cuenta que en algún momento alguien los podría necesitar. Está claro que los datos almacenados que no se usen no se mantendrán actualizados cuando se produzca un cambio en el mundo real. Y el cambio en el mundo real, en algún momento, se producirá.

Por lo tanto, Orr propone programas de calidad de datos basados en el uso, donde la forma de mejorar la calidad es mejorando el uso. Así establece preguntas clave como:

- ¿Qué datos nos interesan?
- ¿Cuál es el modelo de datos?
- ¿Cuales son los metadatos?
- ¿Como se usan los datos?
- ¿Quien los usa?
- ¿Para que propósitos se usan?

- ¿Que tan seguido se usan?
- ¿Cual será el criterio de calidad en los datos almacenados y como se compararán con los datos en el mundo real?

Basadas en estas preguntas, esta propuesta distingue cuatro actividades:

1. *Auditar*: Esta actividad consiste en determinar que tan buenos son los datos hoy. Se deben realizar las siguientes preguntas: ¿En que datos estamos interesados? ¿Quién usa los datos? ¿Con qué propósito son usados? ¿Son usados regularmente? ¿Cuán actuales son los datos?.

2. *Rediseñar*: Se refiere a volver sobre las aplicaciones que están productivas, enfocándose sobre todo en aquellos datos que puedan resultar más críticos para los procesos de negocio soportados por la aplicación y analizar cuidadosamente el uso que se le está dando a estos datos. Esto implica el re planteo de elementos de diseño tales como el número de datos; si los datos no se están usando correctamente nos debemos preguntar si estos proporcionan algún valor a la empresa; desarrollar un conjunto sofisticado de metadata; promover su demanda, para que sean más usados y compartidos; estas prácticas tienen el efecto a largo plazo de mejorar dramáticamente la calidad de datos.

3. *Entrenar*: Los usuarios deben entender la importancia de la calidad de los datos, así de esta manera se dedica tiempo a educación y entrenamiento. Cuando los datos se requieren, deben pensar, ¿Para que los voy a usar?.

4. *Medir*: Se debe medir constantemente la calidad de los datos. Todos los pasos anteriores deben repetirse en el tiempo. Se comprueba si lo que la base de datos dice es verdad, es decir, se audita físicamente.

2.4.2 Matrices de Control

La propuesta [4] plantea medir la calidad de los productos de información emanados de los sistemas a través de múltiples dimensiones tales como la certeza, la accesibilidad, la consistencia, etc. Un producto de información se define como el resultado de transformar la información mediante un proceso computarizado para uso de consumidores de la información, que son aquellos que utilizan los productos de información para su trabajo.

Las matrices de control proveen una forma de combinar problemas con los controles de calidad para así evaluar los productos de información en términos de cómo ellos responden a las necesidades de los clientes, cómo se arman los productos de información, y cómo se maneja el ciclo de vida de los datos después de que se produce.

Las matrices de control se han usado desde los años 70 para evaluar problemas en los sistemas de información. En la propuesta [4] se explica la aplicación de estas matrices para evaluar los problemas de la calidad de los datos en los productos de información.

En la propuesta, las columnas de la matriz enumeran los problemas de la calidad de los datos que pueden afectar el producto de información. Las filas de la matriz son los controles de la calidad ejercitados durante el proceso de fabricación de la información para prevenir, detectar, o corregir estos problemas de la calidad del dato.

Así, estos controles ayudan a evitar que cierto error aparezca en el producto de información. El control puede ser del tipo Si/No o evaluar diferentes grados de cumplimiento de acuerdo a una escala predeterminada. Una vez que la matriz de control del producto se completa, se examina cada columna de error de los datos de la matriz para determinar si la calidad del producto de información se encuentra en un nivel aceptable.

2.4.3 Administración de Calidad del Dato Total

En la propuesta [8] se propone una metodología denominada Administración de la Calidad del Dato (TDQM - Total Data Quality Management) cuyo objetivo es generar productos de información de alta calidad para los consumidores de información. La metodología propone, luego de analizar y conceptualizar el producto de información, la construcción de sistemas que fabrican o

manufacturan la información (SMI). Estos SMI detallan las funcionalidades del sistema con los controles de calidad que debería poseer. Es justamente aquí donde se identifican posibles problemas de calidad analizando cómo se producen los datos.

Se establecen cuatro roles:

- *Proveedores de información*: aquellos que crean y coleccionan datos para los productos de información.

- *Fabricantes de información*: aquellos que diseñan, desarrollan o mantienen los datos para los productos de información.

- *Consumidores de información*: aquellos que usan los productos de información para realizar su trabajo.

- *Administradores de información*: aquellos que son responsables de manejar todo el proceso de producción de los productos de información a través su ciclo de vida.

A su vez, basado en estos cuatro roles, se define un proceso con cinco tareas principales:

- *Definir características del producto de información*: En esta etapa, se conceptualiza el producto de información de acuerdo a la funcionalidad para el consumidor de la información, es decir, la información de los clientes que se necesita para realizar las tareas. Así también se identifican los consumidores y funcionalidades del sistema capturando toda la información que estas personas crean necesarias para realizar las mismas.

- *Definir requerimientos del producto de información*: Aquí se deben identificar los requerimientos desde la perspectiva de los proveedores, fabricantes, consumidores y administradores de información. Luego de obtener estos requerimientos se ingresan en una herramienta (o a mano) para evaluar las dimensiones de la cali-

dad de información. Cada una de las personas que participó en los requerimientos debe evaluar la información según su punto de vista. Por último se debe definir el Sistema que Fabrica la Información (SMI). Este describe como se produce el producto de información y las interacciones entre el proveedor, fabricante, consumidor y administrador del producto de información, es decir entre los cuatro roles definidos.

- *Medir el producto de información*: Esta actividad consiste en desarrollar y aplicar métricas basadas en las dimensiones de la calidad de la información.

- *Analizar el producto de información*: Se trata de la búsqueda de los problemas en la calidad de la información. Los métodos y técnicas para esto varían en complejidad.

- *Mejorar el producto de información*: Sobre los resultados de los puntos anteriores, se pueden implementar mejoras que fueron detectadas en este proceso. Basándose en las características del producto de información de acuerdo a la funcionalidad para el consumidor de la información y con las definiciones de requerimientos del producto de información desde la perspectiva de los proveedores, fabricantes, consumidores y administradores de información.

A diferencia de la propuesta [3], nuestra propuesta intenta definir una guía práctica, aplicable a todos los sistemas que se van a construir en la empresa de aplicación. Las recomendaciones serán calificadas y no se basarán en su uso, sino en como afectan a las cuatro dimensiones de calidad del dato. Sin embargo, muchas de las prácticas que recogimos e incluimos en el marco de trabajo, las recopilamos pensando en las reglas definidas en [3].

La propuesta [4] nos ha servido de disparador para la idea de basar nuestro concepto de calidad en el ciclo de vida del dato. También tomamos la idea de realizar la evaluación de las aplicaciones utilizando una matriz. Para esto clasificamos las distintas recomendaciones de calidad del dato en-

cuadrándolas en el ciclo de vida del mismo y realizando el análisis para evaluar si se respetaban los parámetros que se establecieron.

2.5 Resumen

En este capítulo se vieron los conceptos que guiaron la elaboración de nuestra nueva propuesta.

Se describieron cuatro dimensiones de la calidad del dato y se hizo énfasis en los beneficios que la calidad del dato provee en una empresa, es decir, por qué resulta importante tener en cuenta la calidad del dato durante el proceso de desarrollo.

También se describieron cada una de las etapas del ciclo de vida del dato. Finalmente se presentaron tres de las propuestas más significativas en la literatura y se compararon las mismas con nuestra propuesta mostrando las ideas en las cuales se basa la misma.

Capítulo 3

En este capítulo se presentan las prácticas que surgen como recomendaciones para los sistemas de información. La aplicación de las mismas determina el nivel de calidad de los datos en un sistema.

Estas prácticas se ponderan con dos criterios, *Objetivos* y *Subjetivos* según su independencia del sujeto que realiza el análisis y se explica como afectan a la calidad del dato.

También se define un nuevo proceso de desarrollo que contempla el marco de trabajo; se explica como éste es utilizado para realizar las pruebas que determinan si la aplicación cumple con las mismas y como se registra y documenta dicha prueba.

3.1 Clasificación de las Prácticas

La clasificación de las prácticas fue un proceso que se llevo a cabo para dar un orden a las mismas, de acuerdo a la etapa del ciclo de vida del dato que afecten.

Para esto nos basamos en las cuatro dimensiones de la calidad del dato: Temporalidad, Consistencia, Completitud, y Exactitud.

Por ejemplo, a la práctica de *documentar las interfaces del sistema con otras aplicaciones*, la clasificamos dentro del ciclo de vida *modelado del dato* por considerar que en ese momento es cuando se debe realizar la actividad de documentar las mismas. Las dimensiones de *Temporalidad, Completitud y Exactitud* en esta práctica no son afectadas, en cambio la *Consistencia* si, ya que estamos analizando el origen, destino, compatibilidad de formato con otras aplicaciones y el hecho de documentarlas, implica un análisis y revisión del mismo en el momento del diseño.

La clasificación, está afectada por dos criterios, *Objetivo* y *Subjetivo* según su independencia del sujeto que realiza el análisis. Ambas determinan mediante una tabla de puntaje un *valor de error* si la práctica no se cumple. Esto se explica en detalle en el capítulo 3.

La *Clasificación Objetiva* se enumera a continuación.

- *Práctica Estándar (E)*: La aplicación de este punto es considerado un estándar en la industria y su aplicación debería masificarse. Se aplicó esta clasificación a prácticas que debido a su importancia para cierta dimensión resultaba de aplicación necesaria para garantizar la calidad de los datos.

- *Buena práctica (B)*: Los puntos calificados de esta forma deberían aplicarse siempre que sea posible. Se considera que son muy importantes para garantizar que los datos sean de calidad en las dimensiones que afectan, pero se contempla que a veces resultan difíciles de aplicar por su costo, siendo antieconómicas. En estos casos, se debe documentar la decisión de no aplicarlas como parte del diseño.

- *Recomendación (R)*: La aplicación de estos puntos se considera favorable, quedando a criterio del líder del proyecto y su evaluación costo/beneficio para su aplicación.

Las mismas se consideran objetivas ya que no varían ni de acuerdo al ámbito de la aplicación ni al criterio del evaluador de la aplicación.

El *soporte funcional* quién realizará las pruebas funcionales a la aplicación y la prueba de calidad de datos y una vez en producción dará soporte a las consultas de usuario y estará a cargo de la capacitación de usuario, utilizando este marco de trabajo para determinar una recomendación antes de poner en producción una aplicación; al momento de hacer el test de calidad, tendrá a su disposición una *Calificación Subjetiva*. En la misma se aplica el criterio del soporte y puede variar de acuerdo a su conocimiento del dominio y a su experiencia previa. Finalmente el valor de error de la práctica será la combinación de las calificaciones Objetivas y Subjetivas. Estas últimas se describen a continuación:

- *Sin Error*: El soporte funcional observa que la práctica en cuestión está aplicada de forma correcta en la aplicación.

- *No Aplica*: La práctica recomendada no se observa, pero no se considera error ya que esta decisión de no aplicarla fue tomada en tiempo de diseño y la misma está documentada.

- *Leves*: Situaciones que pueden disminuir la calidad del dato. Cuando se permite el ingreso de datos de baja calidad pero no

afecta a los datos que son críticos ni al éxito de la tarea que esté realizando el usuario.

- *Graves*: Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea. Cuando la aplicación permite que se ingrese un dato que al ser erróneo pueda comprometer la tarea que se está realizando.

- *Fatales*: Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente. Son los más peligrosos ya que permiten que se ingresen datos que impiden terminar la tarea para la cual se los está capturando.

A continuación se describe como cada una de ellas afecta a la calidad en las diferentes etapas del ciclo de vida del dato.

3.1.1 Ciclo de Vida de Dato: Modelado del dato

Esta etapa del ciclo de vida se refiere a la conceptualización del dato en un modelo matemático que se pueda representar en un sistema informático. A continuación se clasifican como marco de trabajo la lista de prácticas a tener en cuenta en esta etapa y como afecta a las diferentes dimensiones del dato:

- *La administración de usuarios debe estar integrada al Active Directory*: Active Directory, es la tecnología de administración de usuarios estándar en la compañía perteneciente al software de base (ya que se integra al sistema operativo). Si un usuario deja de pertenecer a la compañía o cambia de roles, este es el sistema fuente de todos los datos de usuarios y aquí se verán reflejados los cambios. Por lo tanto consideramos que las aplicaciones no debieran administrar la seguridad independientemente de lo reflejado en este sistema evitando así la duplicidad de datos. En la Figura 3.1 se muestra gráficamente la forma en que Active Directory se relaciona con el resto de los servicios de red de la compañía. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (E) Esta práctica es de bajo costo ya que la tecnología existe en la empresa. Cuando esta práctica se aplica el dato tiende a mantenerse valido en el tiempo, ya que si cambia, se reflejará primero en el sistema fuente y el resto de las aplicaciones tomaran los cambios de él.
    - *Consistencia*: (E) Al tomar todos los datos referidos a seguridad de una única fuente, nos aseguramos que el formato de los datos se corresponda en las aplicaciones. Así, si aparecen nuevos da-

tos requeridos que no hubieran sido tenidos en cuenta en el diseño original se actualiza esta misma fuente.

- *Completitud*: (E) Al ser considerada la base fuente de los datos de usuario, la misma se mantiene completa, con todos lo atributos que afectan a los usuarios.
- *Exactitud*: (E) Al evitar la duplicidad y mantener una única fuente de datos, la correspondencia entre los mismos y la realidad se verá beneficiada.

En la Figura 3.2 se muestra como se encuentra implementado Active Directory en la empresa de aplicación. En dicho gráfico se puede ver que a las capas de Datos, Aplicación y Usuarios, se agrega el servicio de Active Directory para que podamos obtener la información que está disponible allí.

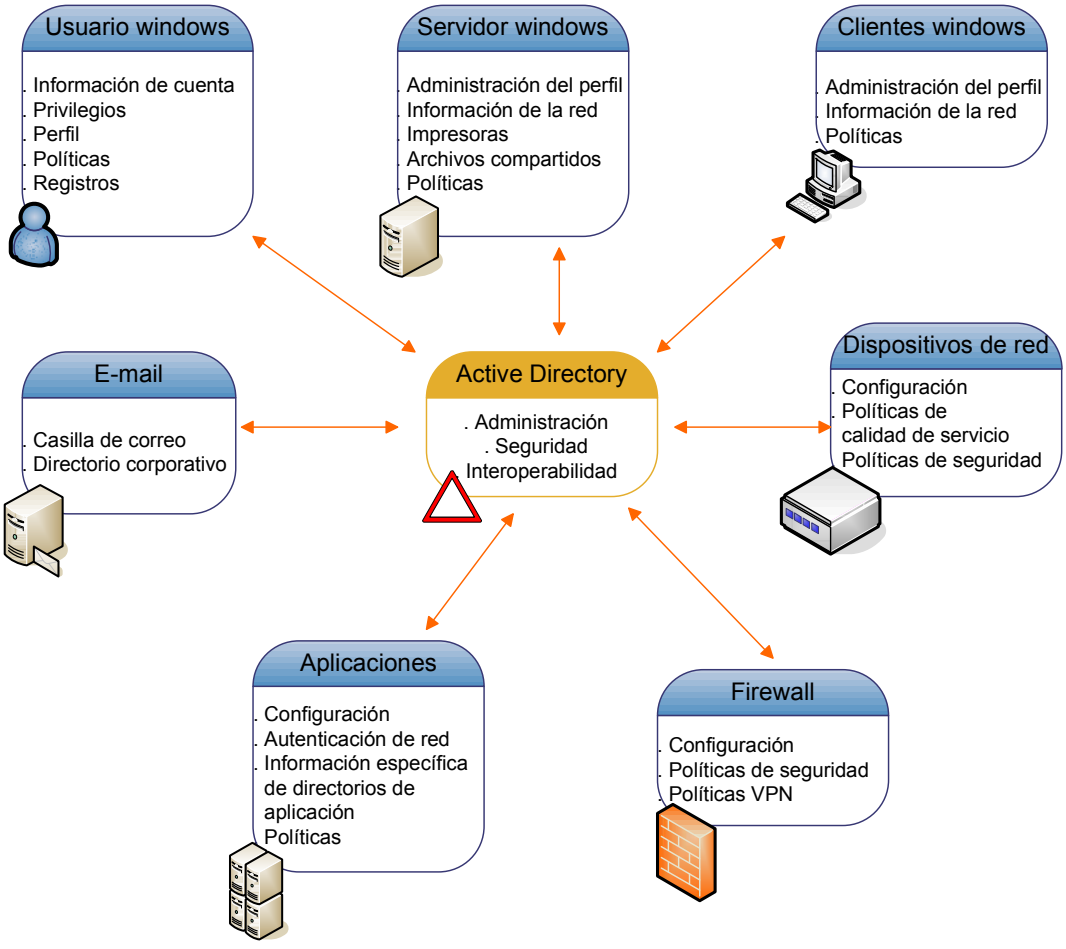

Figura 3.1 Esquema de Active Directory.

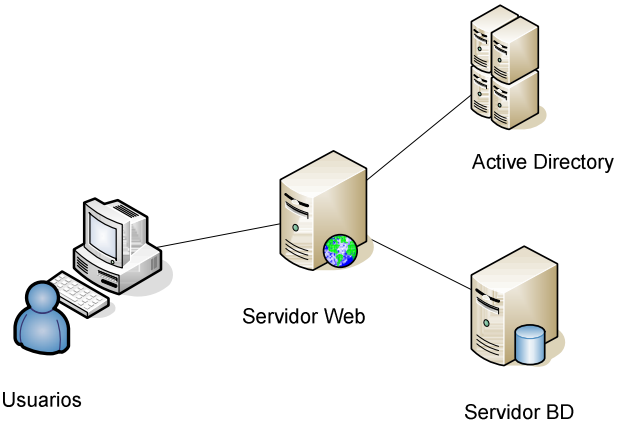

Figura 3.2 Implementación de Active Directory.

- *Debe existir una regla que desactive un usuario que ya no está en el Active Directory*: hace referencia a que si el usuario deja de existir en el sistema fuente o su estado cambia de algún modo, esto se debe reflejar en las aplicaciones satélites (aquellas que usan datos que están definidos en algún sistema fuente de datos). Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (E) El dato se mantendrá valido en el tiempo ya que si cambia, se reflejará primero en el sistema fuente y el resto de las aplicaciones recibirá los cambios de él.
    - *Consistencia*: No aplica
    - *Completitud*: No aplica
    - *Exactitud*: (BP) Al evitar la duplicidad y mantener una única fuente de datos, la correspondencia entre los mismos y la realidad se ve otra vez beneficiada. Además en este caso, se asegura que los cambios en la fuente se repliquen en los satélites, manteniendo actualizada dicha correspondencia.

- *Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas*: De esta forma se mantiene actualizado el flujo de negocio ya que a veces serán necesarias acciones derivadas de este cambio. En la Figura 3.3, observamos que el Administrador De Seguridad Informática da de baja a un usuario en Active Directory, el servidor Web de la aplicación monitorea este cambio y utiliza un servicio de Correo para advertir al Usuario Administrador de dicha aplicación para que tome las acciones derivadas

de este cambio de rol. Esto afecta a las cuatro dimensiones de la siguiente manera:

- *Temporalidad*: (E) El dato se mantendrá valido en el tiempo ya que si cambia, se notifica a los responsables del dato (los consumidores de información definidos como referentes) detectando tempranamente diferencias con la realidad.
- *Consistencia*: No aplica
- *Completitud*: No aplica
- *Exactitud*: (BP) Permite detectar de forma temprana diferencias entre el dato almacenado y la realidad ya que establece los mecanismos para actualizar el dato almacenado cuando se produce un cambio en la realidad, minimizando el impacto negativo de la duplicidad de datos.

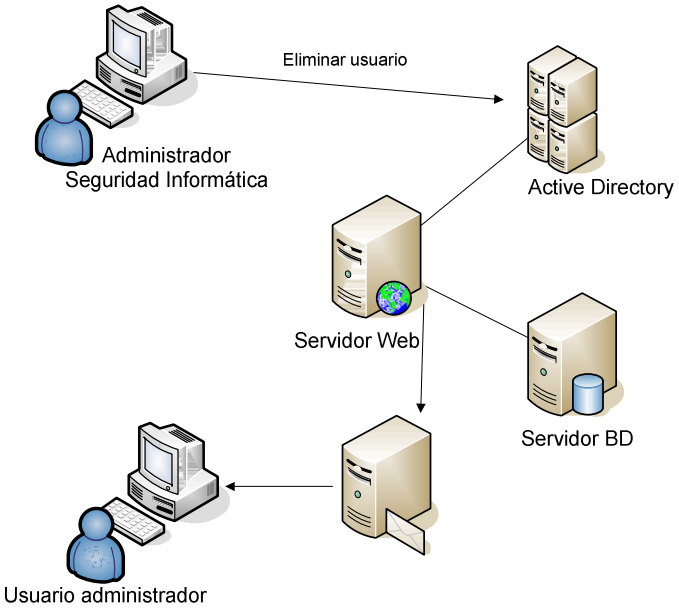

Figura 3.3 Notificación de bajas de un usuario.

- *Se debe proveer una interface de actualización de datos personales de los usuarios (los ajenos a Active Directory)*: Este ítem contempla a los usuarios que se encuentra fuera de Active Directory. Para la, administración de los mismos debe ser provista una interface de actualización modificable por el mismo usuario, el cual es el primero que va a conocer los cambios referidos a sus datos. Esto afecta a las cuatro dimensiones de la siguiente manera:

  - *Temporalidad*: (E) El dato se mantendrá valido en el tiempo ya que si cambia, el mismo usuario es responsable de actualizar sus datos personales.
  - *Consistencia*: No aplica
  - *Completitud*: No aplica
  - *Exactitud*: (E) Se favorece la correspondencia entre los datos almacenados y la realidad ya que el responsable de cambiar los datos es quien los posee.

- *Prohibir el borrado de padres cuando aun existen hijos*: La integridad referencial evita que queden segmentos de información almacenada que luego será inutilizable ya que al eliminar un padre sin haber eliminado a sus hijos, estos no son referenciables y por lo tanto quedan "huérfanos". Gracias a la integridad referencial se garantiza que una entidad (fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos. Por ejemplo en la Figura 3.4, observamos que si se elimina un registro Padre que aún tenga hijos, estos no podrán ser accedidos por la relación existente. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (E) El dato se mantendrá valido en el tiempo ya que si cambia, existen reglas que evitan la perdida de validez del mismo.
  - *Consistencia*: No aplica
  - *Completitud*: (E) Se evita que queden datos dispersos en la base que no son atómicos, es decir interpretables y utilizables por si mismos.
  - *Exactitud*: (E) Se favorece que la correspondencia entre los datos almacenados y la realidad ya que se evita el almacenaje de datos que no se utilicen.

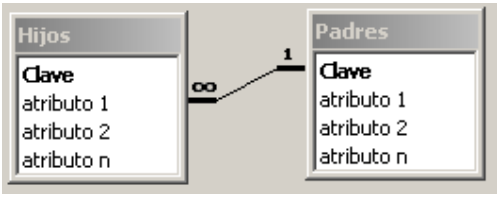

Figura 3.4 Integridad referencial.

- *Las propiedades de configuración regional deben tomarse de la configuración de sistema operativo*: Los sistemas desarrollados en diferentes

áreas de la compañía donde se aplica este trabajo, suelen ser exportados a otras áreas geográficamente dispersas. En estas otras áreas, los husos horarios como el formato del dato fecha y usos de convenciones numéricas, suelen cambiar. Al dejar estas diferencias sujetas a la configuración regional del Sistema Operativo, nos aseguramos que la correlación entre el formato del dato y el dato almacenado a nivel global se corresponda. Esto afecta a las cuatro dimensiones de la siguiente manera:

- *Temporalidad*: No aplica
- *Consistencia*: (E) La práctica debe ser estándar para garantizar que los datos estén almacenados de forma consistente más allá de las diferencias culturales de las distintas áreas geográficas de aplicación.
- *Completitud*: No aplica
- *Exactitud*: (R) Esta recomendación tiende a mejorar la correspondencia entre la realidad y el dato almacenado, ya que se minimiza la posibilidad de errores por diferencias en la convención del formato. Por ejemplo: ¿6-7 es 6 de julio o 7 de junio?

- *Si el dato existe en un sistema fuente, tomarlo de la misma*: Hay aplicaciones que por su dominio son consideradas fuentes de datos ya que son las que capturan el dato lo más cerca posible de su generación. Una vez que un sistema está definido como tal, si el dato es necesario para otra aplicación, no debe duplicarse, sino que debe existir una interface entre este y el sistema fuente. De esta forma no solo se garantiza unicidad sino que se mejora el uso e implementación del sistema fuente. En la Figura 3.5 se grafica la captura de un dato por el sistema fuente. Cuando un sistema satélite necesita este dato, en lugar de volver a capturarlo de la realidad con la posibilidad de diferencias que esto representa, lo toma de la fuente, mejorando así la implementación de este último. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (E) Con esta práctica se favorece a que el dato se mantenga válido por más tiempo y que se detecten cambios tempranamente.
  - *Consistencia*: (E) Al relacionar los sistemas, se mantiene la definición del modelado del dato y se mejora la misma que pudo haber sido buena para una problemática y al verse afectado por la problemática de otro ámbito, se amplía.

- *Completitud*: (E) Se favorece que todos los elementos necesarios del dato estén presentes al interrelacionar los sistemas que hacen uso del mismo.
- *Exactitud*: (E) Al mejorar el uso e implementación del dato, la relación del mismo con la realidad se ve favorecida. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de errores.

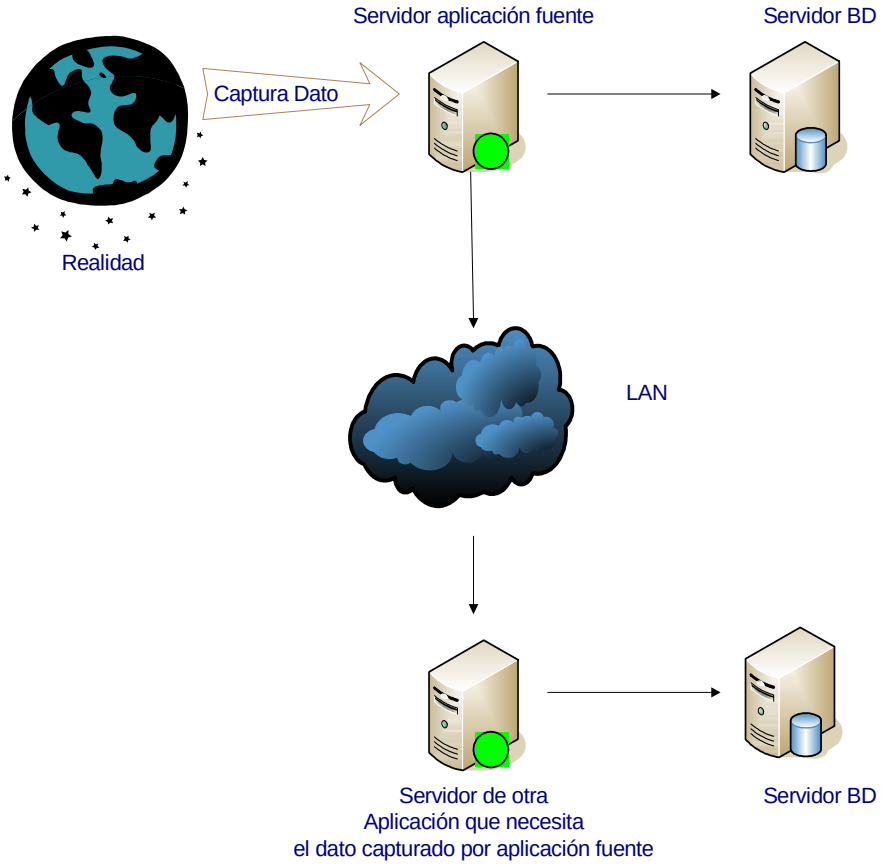

Figura 3.5 Captura de datos que existen en sistemas fuente.

- *Definir nivel de completitud de los datos*: Se exige que la completitud del dato sea un factor a tener en cuenta en el diseño del modelo de datos del diseño de la aplicación. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: (E) Al tener en cuenta esta práctica en el momento de diseño del modelo de datos, se disminuye el riesgo de no tener

un modelo apto para almacenar el dato en todos sus valores. Los cuales pueden no ser iguales para todos los dominios de aplicación.
    - *Exactitud*: No aplica.

- *Definir nivel de granularidad de los datos*: Analizar el nivel de granularidad de la información en tiempo de diseño pone al equipo de desarrollo en una perspectiva analítica del dato y su función que tiende a mejorar el uso de los mismos. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (BP) Al analizar el dato en su mínimo nivel de división, el analista tiende a hacer un reconocimiento de estándares y protocolos para los datos. También se analizará el formato de representación verificando que además sea el más adecuado para la tarea que se está desarrollando.
    - *Completitud*: (E) El tener en cuenta este concepto en momento de diseño del modelo de datos, disminuye el riesgo de tener un modelo no apto para almacenar el dato en todos sus valores, que pueden no ser iguales para todos los dominios de aplicación.
    - *Exactitud*: (R) Esta definición en tiempo de diseño ayudará a mejorar la relación entre el dato y el mundo real al cual está representando el modelo ya que garantiza que se realizó un análisis detallado de la información que será tratada en el sistema.

- *Evitar, siempre que se pueda, las claves de tipo autonumérico*: Basados en la experiencia dentro de este grupo de desarrollo, hemos observado que el uso de este tipo de claves suele ser un ahorro de tiempo de diseño pero perjudica la calidad del modelo de datos. En su lugar deben utilizarse claves naturales que identifiquen al registro y que sean parte de sus atributos, como números de documento de identidad, pasaportes, etc. Las claves generadas por el sistema, como son las autonuméricas, no tienen relación con el registro que representan. La clave debe estar compuesta por atributos propios del dato. En la Figura 3.6 se ve una comparación entre tablas, una indexada por un atributo univoco y la otra con una clave auto numérica. Esto afecta a las cuatro dimensiones de la siguiente manera:

    - *Temporalidad*: No aplica.

- *Consistencia*: (BP) Contribuye a que el analista piense más el modelo y a que la calidad del mismo suba, bajando el riesgo de dediciones apresuradas.
- *Completitud*: No aplica.
- *Exactitud*: No aplica.

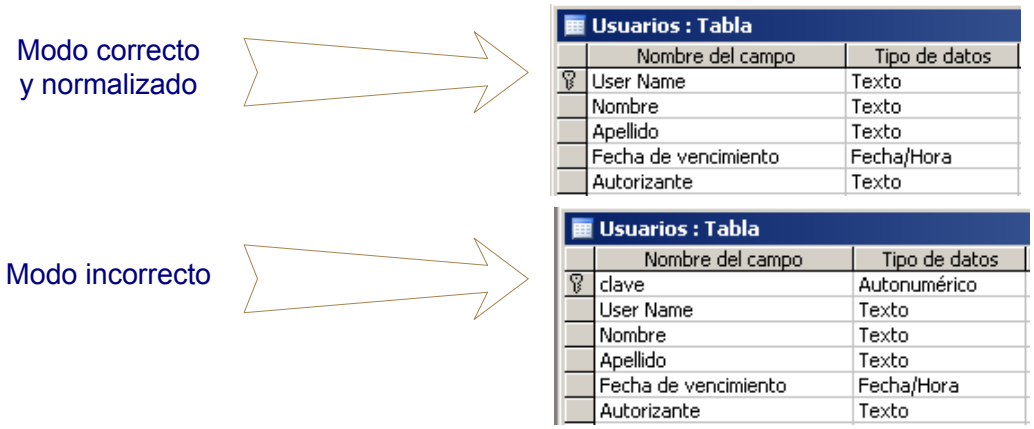

Figura 3.6 Forma correcta de seleccionar un atributo clave.

- *Evaluar la conveniencia que los datos sean escritos todos en mayúscula*: Esta práctica se aplica para unificar el contenido de los datos, evitando fallas de consistencia. También ayuda en la implementación de las búsquedas de los datos ya que no se deben hacer cambios de formato, etc. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Ayuda, en algunos casos, a mantener una unicidad en la representación de los datos.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *Evitar caracteres especiales en los nombres de las tablas de Bases de datos*: Esto podría dificultar la comprensión del modelo y su abstracción en una solución. La Figura 3.7 muestra un ejemplo de la utilización de alguno de estos caracteres. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Ayuda, en algunos casos, a mantener una unicidad en la representación de los datos.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

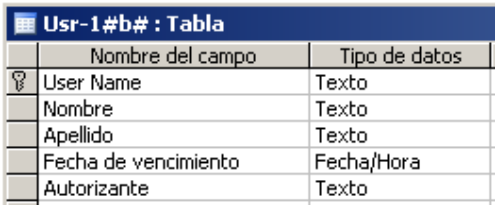

Figura 3.7 Tabla con caracteres especiales en el nombre de la misma.

- *Acotar en lo posible el dominio como reglas dentro de la base de datos*: Las reglas de negocio deberían estar modeladas en la solución, de forma tal que si existen reglas para formar o inferir datos en base a otros datos debiera reflejarse en el modelo. En la Figura 3.8 se observa la captura de dos datos y el almacenamiento de un tercero, que en lugar de ser capturado fue inferido por una regla de negocio relevada. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (E) Las reglas dentro del modelo ayudan a que los datos hereden propiedades de los datos que les dan origen.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) Evitar la captura de datos que se conforman de otros datos ayuda a disminuir las probabilidades de error en la captura.

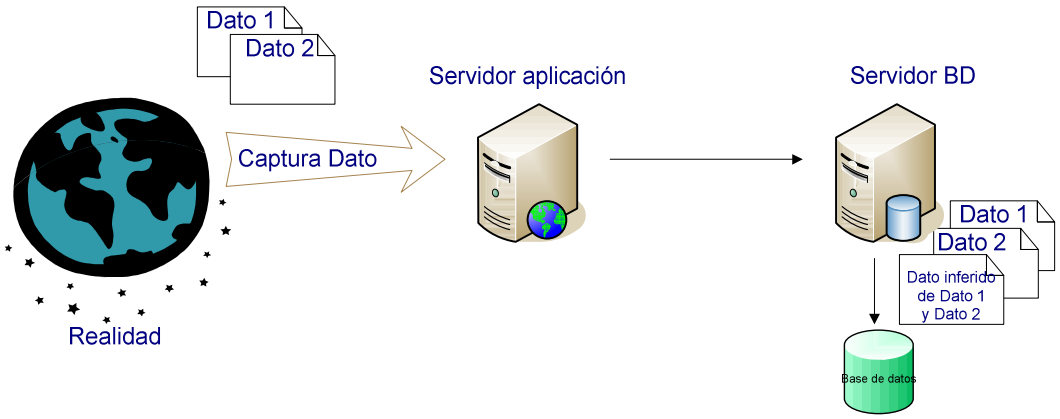

Figura 3.8 Almacenamiento de datos inferidos por reglas de negocio.

- *Almacenar la fecha de las relaciones que pueden cambiar (fecha de alta, fecha de baja)*: De esta forma se puede tener control del tiempo transcurrido desde la última acción para poder establecer controles periódicos.

Por ejemplo, si han pasado 5 años desde el último cambio de domicilio, contemplar la posibilidad de validar esa información. En la Figura 3.9 se muestra una tabla de este tipo. Esto afecta a las cuatro dimensiones de la siguiente manera:

- *Temporalidad*: (E) Esta práctica sencilla y de bajo costo contribuye a que los datos se mantengan válidos por más tiempo ya que se pueden realizar acciones pro activas para evitar su volatilidad.
- *Consistencia*: No aplica.
- *Completitud*: No aplica.
- *Exactitud*: (E) Las acciones preventivas permiten detectar cambios y corregir el registro para mantener su correspondencia con la realidad.

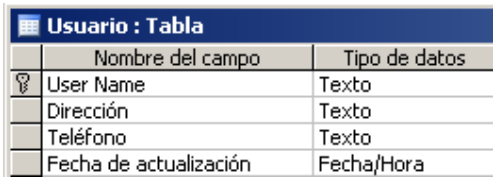

Figura 3.9 Tabla con atributo fecha de actualización.

- *Deben existir análisis y clasificaciones de datos en función de su criticidad, para enfocar el esfuerzo en los más críticos*: De acuerdo al Diagrama de Pareto [15] hay muchos problemas sin importancia frente a solo unos pocos graves ya que por lo general el 80% de los resultados totales se originan en el 20% de los elementos. La Figura 3.10 muestra esta correspondencia. Esta distribución suele asociarse a todos los órdenes. El calificar los datos en función de su criticidad asegura que se pueda enfocar en el esfuerzo de mantener los mismos sin desviar recursos al mantenimiento de datos poco críticos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (BP) Está práctica permite enfocar los esfuerzos en tiempo de mantenimiento de los datos.
  - *Consistencia*: (BP) De esta forma se puede pensar en dedicar más atención a mantener los datos más críticos una vez detectada una anomalía.
  - *Completitud*: (BP) Teniendo esta clasificación se espera atacar los problemas que afectan más al servicio de forma prioritaria.
  - *Exactitud*: (BP) Con este análisis, en el momento de realizar un mantenimiento, podemos pensar en primero solucionar los problemas más críticos.

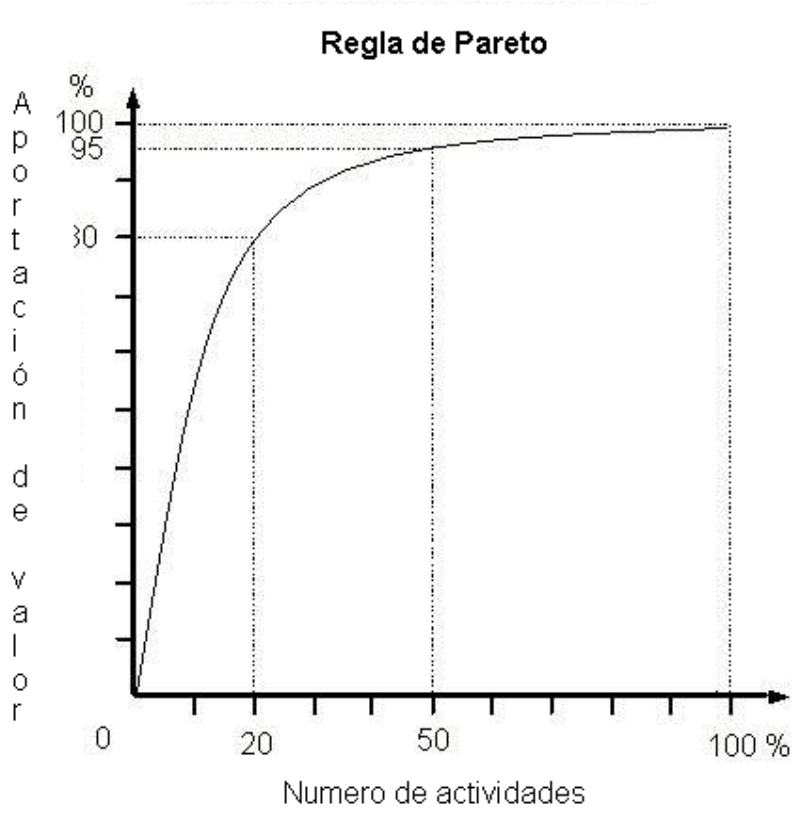

Figura 3.10 Regla de Pareto.

- *Deben existir mecanismos de registro de uso del dato (haciendo hincapié en los críticos - métricas de uso)*: En capítulos anteriores hemos definido el concepto de calidad de datos de acuerdo a varios autores en la literatura [7,2,6]. Todas estas definiciones convergen en que el concepto calidad del dato es *relativo al uso del dato* [7]. También que la calidad del dato no será mayor que su uso más riguroso. Por lo tanto, es necesario medir los resultados de lo que se realiza para monitorear cuales son los datos que no se usan y analizarlos, ya que son los mas propensos a perder calidad. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (BP) Está práctica permite determinar datos que sean propensos a perder calidad.
    - *Consistencia*: No aplica.
    - *Completitud*: No aplica.

  - *Exactitud*: (BP) Al identificar los datos menos usados podemos hacer esfuerzos por determinar si los datos de la realidad y el dato modelado no coinciden.

- *Las interfaces del sistema con otros, deben estar documentadas*: De esta forma puede hacerse un mejor análisis del uso y origen de los datos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica
  - *Consistencia*: (R) Permite que en tiempo de diseño tengamos más noción del dato en si, su origen y / o destino.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *El modelo de datos debe estar disponible*: sea en un desarrollo interno de la compañía o un producto comprado a un proveedor externo. El modelo de datos es necesario para realizar cualquier análisis del diseño y determinar posibilidades de mejora en los esquemas de los datos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (R) Se asegura la disponibilidad de esta herramienta para la interpretación del modelo.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *El diseño de la base de datos debe responder a un proceso de negocio para determinar el alcance de los datos almacenados*: Se establece que para modelar un dato se debió haber analizado previamente el proceso de negocio que lo generó. En la Figura 3.11 se muestra que el proceso más grande está formado por sub-procesos que deben ser parte de la solución total. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (BP) De esta forma el dato estará mejor definido y tendrá menos volatilidad ya que responde a un análisis de proceso y no solo a un análisis del dato por si mismo.
  - *Consistencia*: (BP) Por lo mismo, estará definido de forma más consistente.
  - *Completitud*: (BP) Así también de forma más completa.
  - *Exactitud*: (BP) Por lo anterior, debiera tender a corresponderse con el mundo real.

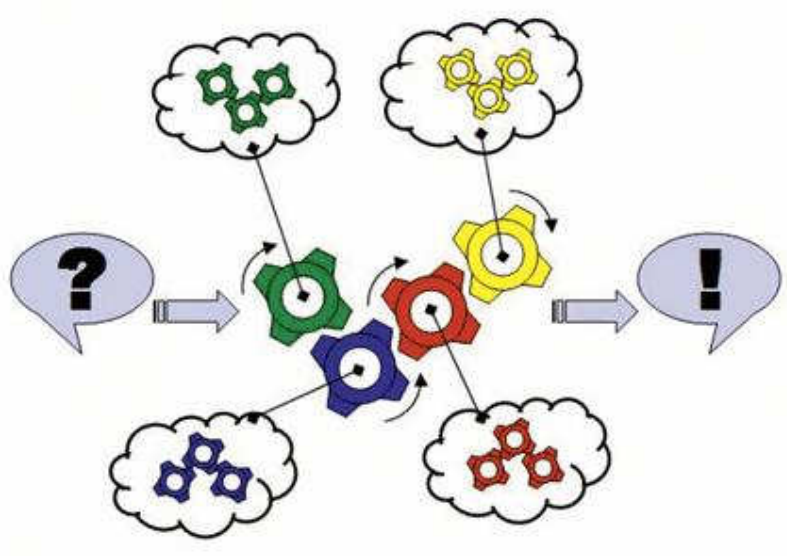

Figura 3.11 Diseño orientado a procesos.

- *La etiqueta que describe al dato debe ser comprensible*: Se deben evitar las abreviaciones, siglas y otros que en el momento de diseño ahorran muy poco tiempo y suelen dificultar luego la comprensibilidad del modelo. En la Figura 3.12 se muestra la diferencia entre un dato comprensible y otro que no lo es. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (BP) La mejor definición del dato, reflejada en la expresión del modelo, colabora con su análisis posterior a la hora del mantenimiento.
    - *Completitud*: No aplica.
    - *Exactitud*: No aplica.

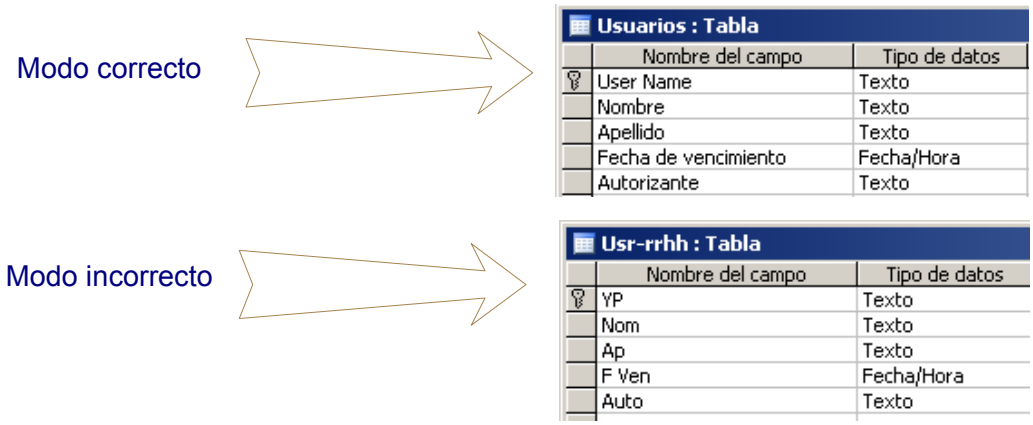

Figura 3.12 Ejemplo de etiquetas de atributos incomprensibles.

- *Se debe conocer como llegar al dato fuente (cuando el dato es obtenido de otro sistema)*: De esta forma, se podrá analizar al mismo cuando surjan dudas o diferencias. Esto se logra a través de una clara documentación de las consultas efectuadas a la fuente y sus interfaces. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Esta práctica tiende a mantener de forma más estrecha la relación entre el dato y su origen, por lo cual mejora su consistencia.
  - *Completitud*: No aplica.
  - *Exactitud*: (BP) Al mantener la relación entre el dato y su origen, tiende a disminuir riesgo de perdida de la correspondencia con la realidad.

- *El diseño de la base de datos debe respetar estándares para nomenclaturas de atributos*: Nos referimos a los nombres que se le asignan a las propiedades de un objeto modelado en una tabla, los estándares me aseguran que se mantenga coherencia entre los distintos sistemas, ayuda a su interrelación y mejora la comprensibilidad. Además evita el estar re-inventando soluciones a los mismos problemas. Por ejemplo: en todos los sistemas de la empresa, el nombre del pozo tiene la misma estructura: Empresa. Provincia. UnidadEconómica-NúmeroTipo. De Forma tal que un poso se llama YPF.NQ.EPN-102A y de esta forma lo puedo localizar en cualquier sistema en el que se encuentre. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.

  - *Consistencia*: (BP) Al re-utilizar soluciones exitosas que a su vez se consideran estándares, se asegura que se mantiene consistencia.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *La definición de los datos debe ser compatible con los estándares de la compañía*: De esta forma, además de lo dicho anteriormente respecto a estándares, nos aseguramos respetar lo definido por la compañía de aplicación. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Al re-utilizar soluciones exitosas consideradas estándares, se asegura que se mantiene consistencia.
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *Consensuar el modelo de datos a nivel esquemático con el usuario referente*: De esta forma validamos la abstracción elaborada por el analista con el negocio, sin entrar en detalles ni tecnicismos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Al validar el modelo los errores conceptuales de interpretación se verán minimizados
  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *Consensuar la presentación de los datos con los consumidores del dato*: Esta tarea se realiza en momento de diseño pero afecta a la visualización. Tiende a detectar errores de interpretación en las reglas de negocio y conceptualización del modelo. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) Al validar el modelo los errores conceptuales de interpretación deberían minimizarse.
  - *Completitud*: (BP) Al validar el modelo se espera que salgan a la luz los puntos que se hayan omitido en el relevamiento inicial.
  - *Exactitud*: No aplica.

- *Evitar que un dato esté duplicado en más de un sistema*: Con esta práctica no solo se garantiza unicidad sino que se mejora el uso e im-

plementación de los sistemas participantes. Esto afecta a las cuatro dimensiones de la siguiente manera:

- *Temporalidad*: (R) Con esta práctica el dato se mantendrá válido por más tiempo y se podrán detectar cambios tempranamente.
- *Consistencia*: No aplica.
- *Completitud*: No aplica.
- *Exactitud*: (R) Al mejorar el uso e implementación del dato, se favorece la relación del mismo con la realidad. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de error.

- *Las interfaces y las aplicaciones satélites deben usar la misma terminología que la fuente*: Evitando así diferencias de interpretación al analizar los datos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (R) Se mantiene lo definido para la fuente como válido para los satélites o se mejora la implementación de la fuente.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) Se ayuda a mejorar la correspondencia entre la realidad y la fuente, mejorando su implementación y uso.

3.1.2 Ciclo de Vida del Dato: Captura del valor

En esta etapa, el dato, es tomado de la realidad y representado en la abstracción modelada. A continuación se clasifican la lista de prácticas a tener en cuenta y como afectan a las diferentes dimensiones del dato:

- *Se deben ofrecer ejemplos del tipo de datos con el cual completar los formularios*: Se espera que a través del ejemplo en cada dato capturado en forma manual, el usuario tenga una idea de lo que se espera que se introduzca. En la Figura 3.13 se muestra como en la pantalla de carga de una aplicación donde debe transferirse información de una planilla impresa a una pantalla de carga, se ha graficado la planilla y señalado donde está cada dato a incorporar en la aplicación a modo de ayuda. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (R) Esta práctica tenderá a mantener el formato de los datos capturados con uniformidad.
  - *Completitud*: (R) El usuario tendrá esta ayuda para saber que se espera que el dato contenga a un cierto nivel de desagregación.

- *Exactitud*: No aplica.

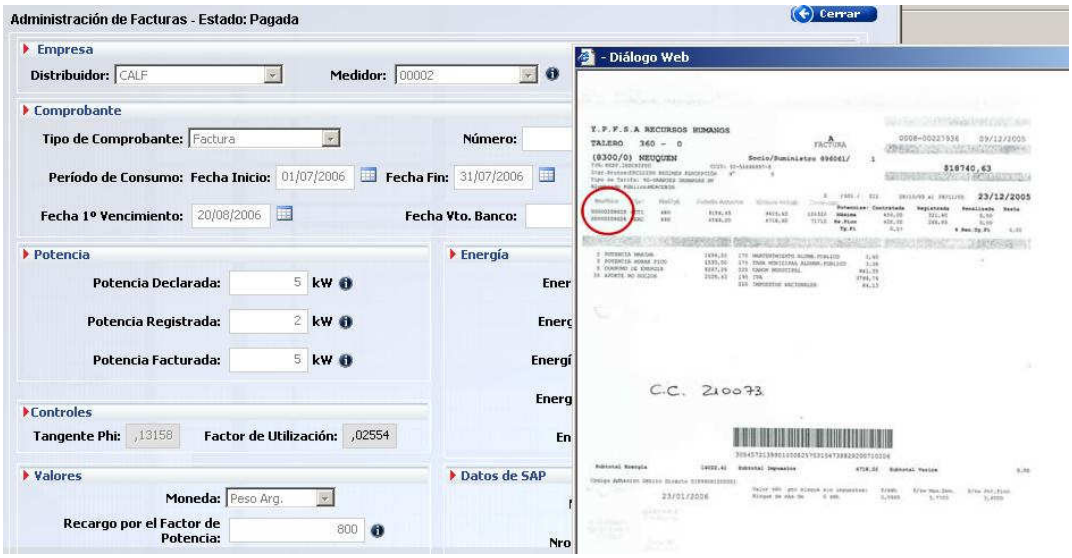

Figura 3.13 Ejemplo para orientar al usuario que va a cargar los datos.

- *Los Campos Fecha deben tener su calendario asociado*: Esto se refiere a que el campo de ingreso de fechas no debe ser texto, sino un objeto calendario como se ve en la Figura 3.14. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (R) Esta práctica impide que las fechas se ingresen con un formato inválido.
  - *Completitud*: (R) La fecha ingresada de esta manera no podrá estar incompleta.
  - *Exactitud*: No aplica.

Figura 3.14 Calendario asociado a un campo fecha.

- *Los Campos con formato particular (guiones, barras intermedias) deben tener una máscara*: Esto se refiere a que el campo de ingreso no deje ingresar un formato que no se corresponda con la definición del mismo. Este flujo se ve reflejado en la Figura 3.15. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (R) Esta práctica impide que los datos se ingresen con un formato inválido.
    - *Completitud*: (R) El dato ingresado de esta manera no podrá estar incompleto porque la máscara no lo permitirá.
    - *Exactitud*: De la forma antes descripta se favorece a que no se almacenen datos inexactos.

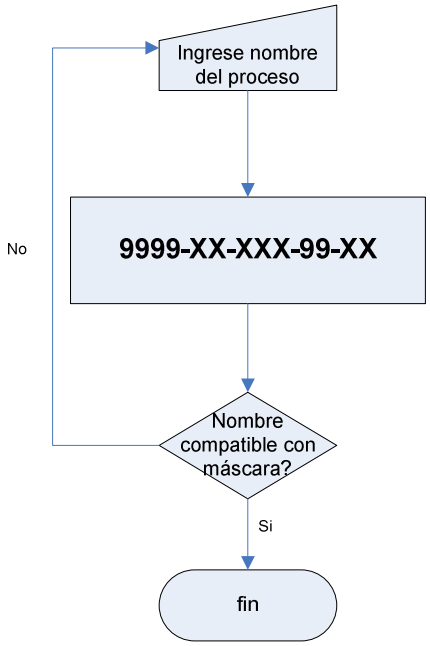

Figura 3.15 mascara para validar formato de campo ingresado.

- *Los Campos Fecha Nacimiento no pueden ser posteriores a la fecha del sistema operativo*: De esta forma se controla la inconsistencia del dato. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Esta práctica impide que el dato sea inconsistente.
    - *Completitud*: No aplica.

  - *Exactitud*: (R) De la forma antes descripta se favorece a que no se almacenen datos inexactos.

- *Fechas relacionadas a una persona: no pueden superar un valor definido*: Dicho valor debe haber sido definido en el momento del relevamiento, por ejemplo si al calcular la edad de una persona me da un valor superior a lo definido como expectativa de vida de un ser humano, puedo sospechar errores en los datos. De esta forma se controla que el dato sea consistente con las reglas impuestas por esta práctica. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (E) Esta práctica se usa para que el dato no pierda validez.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (BP) De la forma antes descripta se favorece a que no se almacenen datos inexactos.

- *Luego de una modificación, debe existir un mensaje que advierta que las modificaciones fueron realizadas*: De esta forma se espera que el usuario que cargo los datos tenga una confirmación de la tarea realizada y así evitar que la misma se haya hecho por error. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) De la forma antes descripta se favorece a que no se almacenen datos inexactos.

- *Validar la repetición de datos (Ej.:111111111111111111111) de acuerdo a reglas*: De esta forma se espera minimizar la posibilidad de que el usuario que ingresó los datos haya incurrido en un error de tipeo. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (E) Se espera mantener el dato en relación a los formatos definidos.
  - *Completitud*: No aplica.
  - *Exactitud*: (E) Se espera evitar que los datos ingresados no se correspondan con la realidad.

- *Validar la existencia de caracteres "especiales" en campos donde no debieran existir*: De esta forma se espera minimizar la posibilidad de que el usuario que cargo los datos haya incurrido en un error de tipeo. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Se espera mantener el dato en relación a los formatos definidos.
    - *Completitud*: No aplica.
    - *Exactitud*: (E) Se espera evitar que los datos ingresados no se correspondan con la realidad.

- *Evitar en lo posible la carga manual*: De esta forma se minimiza el error ya que se elimina de la cadena de captura del dato a la fuente más grande de error, al eslabón humano. Para esto es necesario un análisis de costo/beneficio de implementar un sistema de tele medición. En la Figura 3.16 se observa un esquema topológico típico de una red de tele medición. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (BP) Al minimizar el error humano reemplazando la carga por un instrumento de tele medición, se asegura que se mantiene el dato en el formato pre definido.
    - *Completitud*: (BP) El dato capturado de forma automática se almacenará en el nivel de granularidad definido.
    - *Exactitud*: (E) De esta forma se maximiza la correspondencia con la realidad dejando sólo un mínimo riesgo de falla en el instrumento.

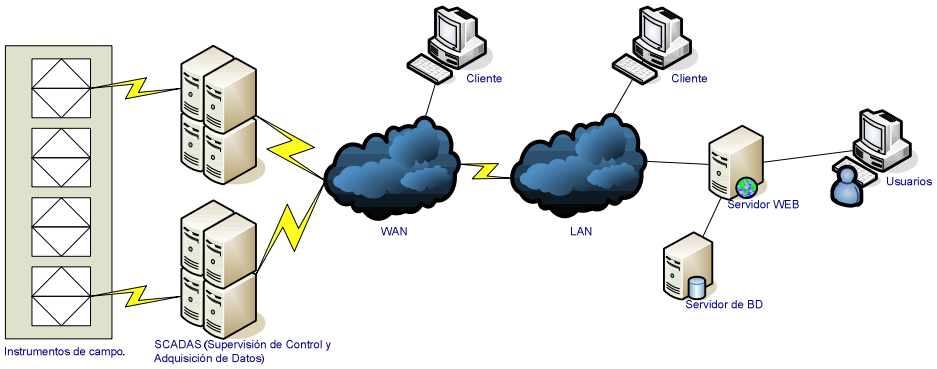

Figura 3.16 Ejemplo de topología de tele supervisión y captura automática de datos.

- *Solicitar confirmación de la operación de Eliminación*: Para evitar así las eliminaciones de datos por error. Una pantalla de confirmación se muestra en la Figura 3.17. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: No aplica.
    - *Completitud*: No aplica.
    - *Exactitud*: (R) Se espera evitar que los datos no se correspondan con la realidad por eliminaciones erróneas.

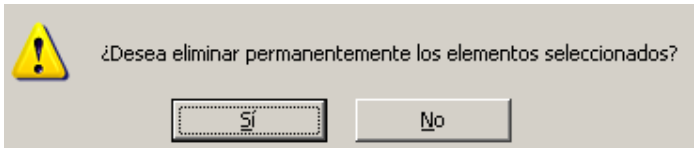

Figura3.17 Ejemplo de confirmación de eliminación de datos.

- *El sistema debe informar al cerrar que se deben grabar los datos o de lo contrario se perderán los cambios*: Mediante una pantalla como la que se observa en la Figura 3.18 el usuario debe saber que debe almacenar la información con la que esta trabajando actualmente. Así se espera evitar que el usuario crea haber cargado los datos cuando en realidad los mismos no quedaron almacenados. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (R) Se evita que el dato se haga inválido pese a que el cargador de datos crea haber actualizado el valor.
    - *Consistencia*: No aplica.
    - *Completitud*: (E) Se evita que falte información que el cargador de datos cree haber introducido al sistema.
    - *Exactitud*: No aplica.

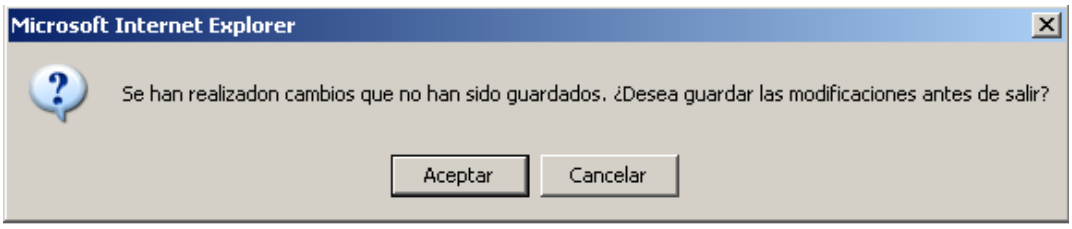

Figura 3.18 Ejemplo de advertencia de modificación no guardada.

- *Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción*: Este punto se refiere a permitir ingresar un dato luego de haber hecho una comprobación de si el mismo existe en la base de datos. Por ejemplo, si se ingresa una calle y se coloca como dato “Rivadavia” el sistema debiera consultar en la base de calles y comprobar que existen dos datos coincidentes. Por lo tanto preguntarle al usuario si se refiere a “Comodoro Rivadavia” o a “Bernardino Rivadavia”. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Asegura que el dato se mantiene consistente ya que no se guarda el ingreso sino la coincidencia con la tabla tipificada.
    - *Completitud*: (E) Por lo mismo, el dato se guarda completo y no solo lo que se ingresa.
    - *Exactitud*: (E) Se disminuye la posibilidad de error de carga y se elimina la posibilidad de error de tipeo.

- *El dato debe ser capturado lo mas cercano a la fuente*: Así se minimizan las probabilidades de error, por ejemplo la actividad de un equipo perforador podría ser volcada en una planilla que una vez al día se envía a una oficina para su carga en una herramienta o la misma se puede realizar directamente en el equipo perforador. La Figura 3.19 muestra esta implementación la cual se corresponde a una implementación real de la empresa de aplicación. En la primer opción la probabilidad de error es más alta porque hay posibilidad de error en la comunicación y porque la persona que carga no conoce lo que pasó en dicho equipo. Mientras en el segundo caso, la persona que carga esta en el lugar de los hechos. El caso ideal de esta cercanía es la tele supervisión de ese dato. En la Figura 3.20 se ve otra implementación real de la empresa donde se implementó la carga de datos de instalaciones de campo en agendas electrónicas en lugar de planillas. Luego la subida a la aplicación SAP se hace sincronizando la información de las pocket. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (E) Esta práctica ayuda a que el dato se mantenga válido porque la cadena de incorporación del dato al sistema es más corta.
    - *Consistencia*: (E) Por la misma razón, el dato tenderá a estar en un formato idóneo ya que quienes realizan la carga conocen el dominio de aplicación.

- *Completitud*: (E) Esto ayuda a que no se pierdan trozos de información por errores de interpretación.
- *Exactitud*: (E) También se ve beneficiada ya que la realidad está más cerca de la captura.

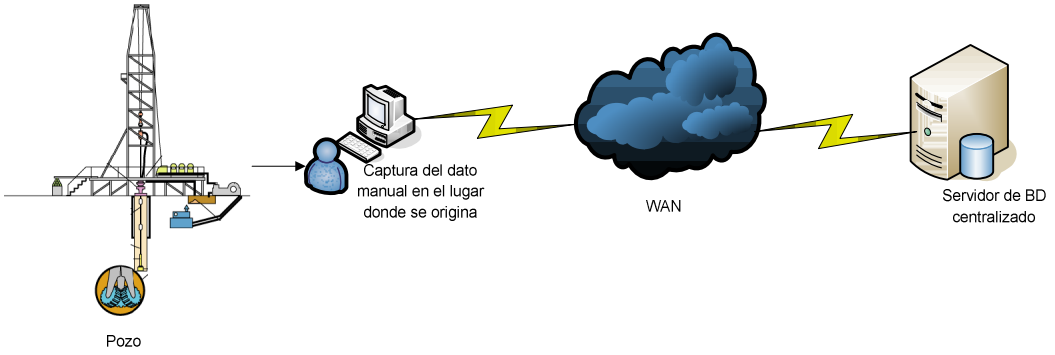

Figura 3.19 Esquema real de captura de datos en aplicación de registro de actividades de perforación.

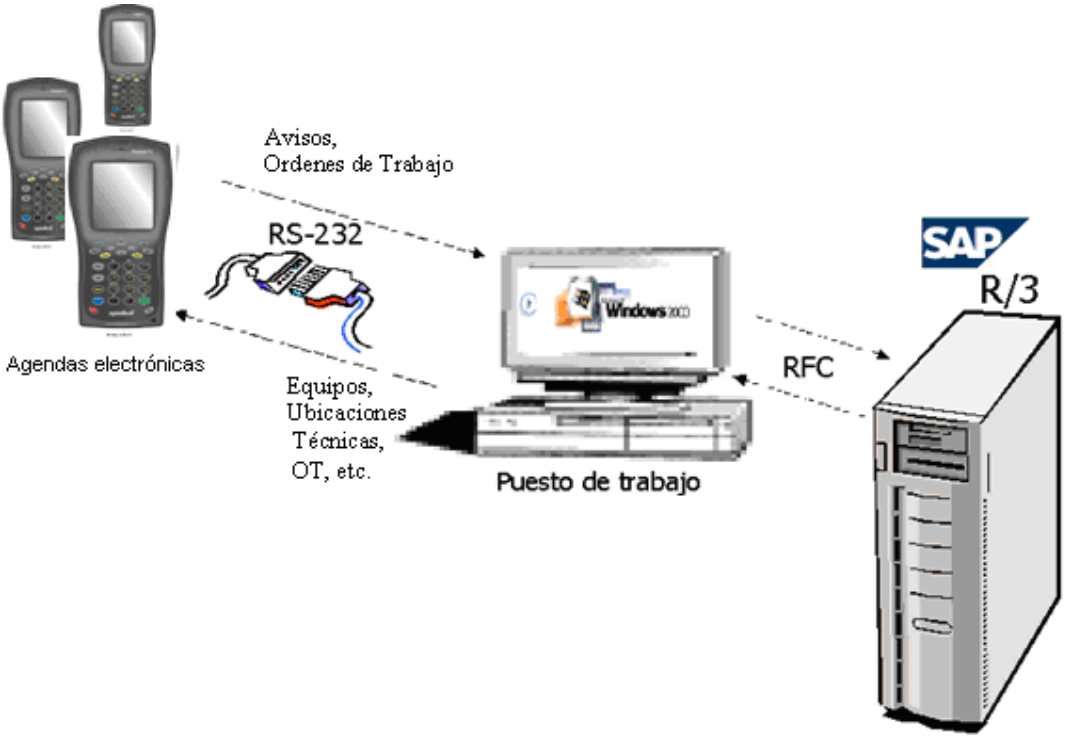

Figura 3.20 Esquema real de captura de datos de mantenimiento en el campo a través de agendas electrónicas.

- *¿Existen datos tipados? Si es así, deben poder ser desplegados de un campo de selección*: De esta manera se impide que se ingresen datos que no pertenecen al dominio. Para esto el relevamiento debe haber tipificado las posibilidades por extensión, ejemplo de esto es el grupo sanguíneo. En lugar de dejar ingresar cualquier carácter, debe aparecer un menú con las únicas posibilidades (A,B,AB,0). La Figura 3.21 muestra gráficamente este ejemplo. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Al acotar las posibilidades al único conjunto válido de respuesta, nos aseguramos que el formato de ingreso sea el correcto.
    - *Completitud*: (E) Por lo mismo, que será completo.
    - *Exactitud*: (E) Se incrementan las posibilidades de una correspondencia con la realidad reduciendo sólo al caso de error en la carga del valor las probabilidades de inexactitud.

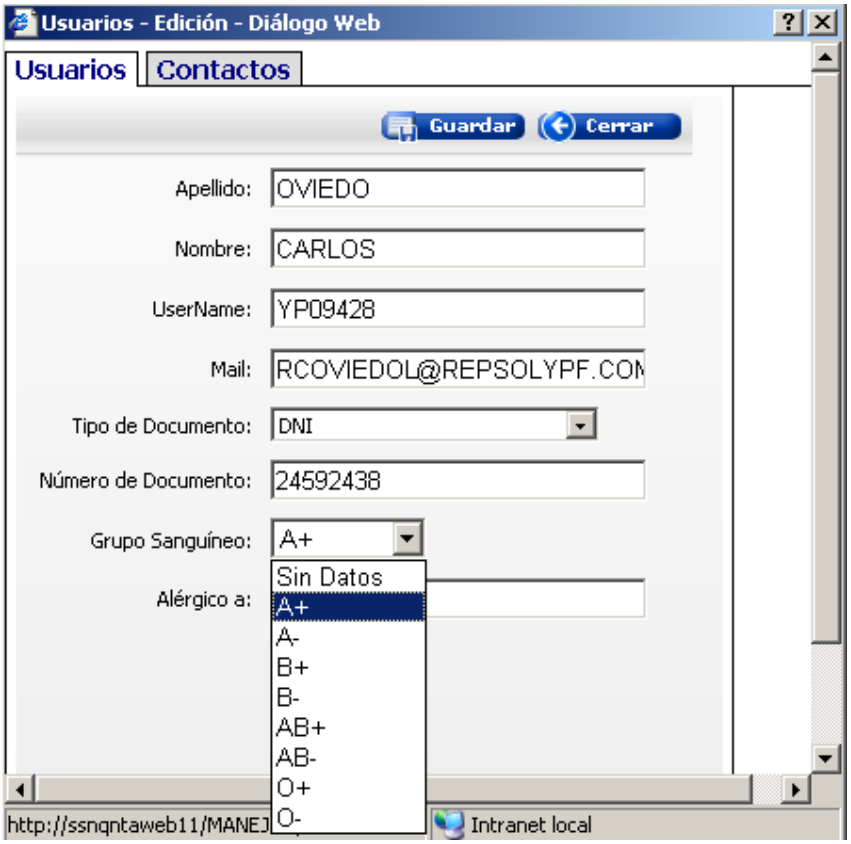

Figura 3.21 Ejemplo de tatos tipo tabulados cuyo ingreso se ha acotado.

- *Los Campos TIPO deben tener validación para no ingresar nombres duplicados mediante índice UNICO*: De esta manera se espera evitar que ingrese basura en la base de datos tipificada. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Esto impedirá que al capturar los datos, los mismos tengan una definición inconsistente.
    - *Completitud*: No aplica.
    - *Exactitud*: No aplica.

- *Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez*: Esto debe ser evaluado con el usuario referente para evitar que la carga sea tediosa. Por otro lado minimiza el error al combinar las probabilidades. Un ejemplo, si se introduce una clave de acceso nueva, como se considera un dato crítico, obligar a la persona que carga los datos a que el dato sea ingresado dos veces en la pantalla de carga, compararlos y advertir si hay diferencias. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: No aplica.
    - *Completitud*: No aplica.
    - *Exactitud*: (E) Minimiza la probabilidad de error de tipeo.

- *El diccionario de datos debe estar disponible con la descripción del dominio de cada dato y sus excepciones*: De esta forma se despejan dudas en el momento de la carga del dato. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (E) Ayuda en línea para que el cargador de datos sepa el formato en que es requerida la información.
    - *Completitud*: (E) Ayuda en línea para entender la completitud de la información requerida.
    - *Exactitud*: No aplica.

- *Evitar el uso de siglas en la interface de usuario*: Ayuda a clarificar la interface y a confundir menos al usuario. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: (BP) Con una interface más limpia, el usuario será menos propenso a errores.

  - *Completitud*: No aplica.
  - *Exactitud*: No aplica.

- *Si el ingreso de un dato es erróneo, se debe informar en el momento y de forma visible y fácil de corregir*: El hecho de avisar en el momento ayuda a que el usuario todavía tenga posibilidad de verificar el dato en la fuente. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: (BP) Se pueden detectar tempranamente y se da la oportunidad de corregir problemas de información faltante.
  - *Exactitud*: (BP) De igual manera con información inexacta.

- *Minimizar la trascripción del dato*: Esta práctica sirve para evitar la probabilidad de error y de mala comunicación. Es una manera de acercar más la carga al origen del dato. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (BP) Esta práctica disminuye la probabilidad de error y de falta de correspondencia entre el dato y el mundo real.

- *El dato lo debe ingresar quién lo genera y de no ser posible, alguien que esté muy cerca (físicamente) de él*: Al igual que el punto de la cercanía con la fuente, esto se minimiza las probabilidades de error. Quién generó el dato es quién mejor conoce el dominio de aplicación del mismo. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (BP) Por dicho anteriormente, esta práctica favorece a que el dato se corresponda con la realidad.

- *Optimizar la información mostrada en la pantalla de captura de datos*: Ayuda a clarificar la interface y a no confundir al cargador de datos. Evitando que la información en la pantalla de carga esté sobre cargada o confusa. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.

- *Completitud*: No aplica.
- *Exactitud*: (R) Con una interface más limpia, el usuario tendrá menos posibilidades de cometer errores.

3.1.3 Ciclo de Vida del Dato: Almacenamiento

En esta etapa, el dato ya representado en un modelo de abstracción, es almacenado en la estructura de datos que se haya considerado más apropiada para la solución. A continuación se clasifican como marco de trabajo la lista de prácticas a tener en cuenta en esta etapa y como afecta a las diferentes dimensiones del dato:

- *Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar*: esta regla de inferencia debe estar modelada en la aplicación para evitar así el error de ingreso de datos. En las Figuras 3.22 y 3.23 se observan diferentes ejemplos de cómo un dato que se puede inferir, no se ingresa. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (BP) Ayuda a mantener el dato válido porque al cambiar los datos que le dieron origen, estos se actualizarán.
  - *Consistencia*: (BP) Tendrá el formato esperado porque se define dentro de la aplicación.
  - *Completitud*: (BP) Será un dato completo porque no hay ingreso humano y la regla deberá validar que los datos que le dan origen le dan en toda su completitud.
  - *Exactitud*: (E) La correspondencia con la realidad se mantendrá mientras la regla de inferencia esté bien modelada.

Figura 3.22 Ejemplo de datos inferidos mediante reglas matemáticas.

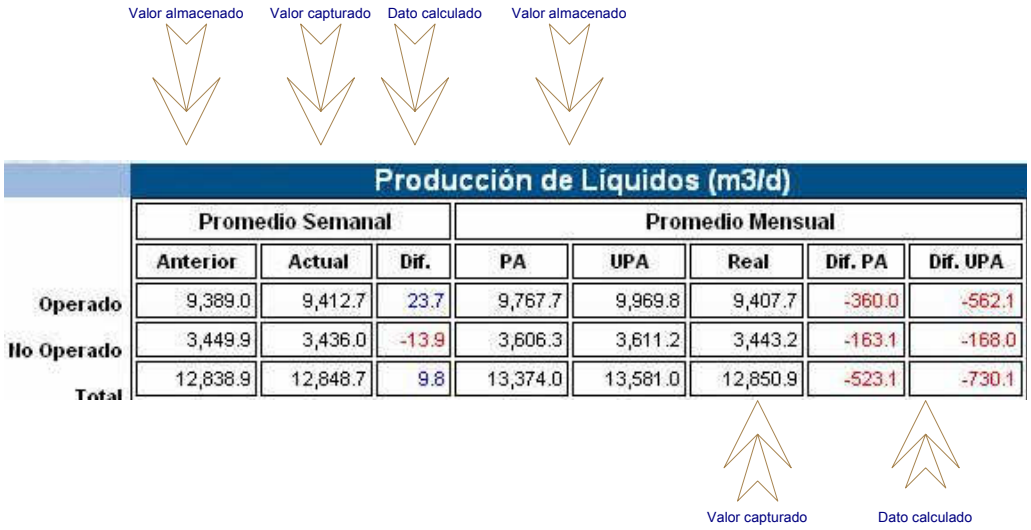

Figura 3.23 Ejemplo de datos inferidos mediante reglas matemáticas.

- *Si existe una regla matemática de verificación del campo, la misma se debe utilizar*: De esta forma se ayuda a eliminar errores de carga en el momento de realizarse la misma. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.

  - *Consistencia*: (E) Esto ayuda a que el dato se mantenga dentro del formato esperado y esta comprobación se puede hacer en el momento de carga y almacenado.
  - *Completitud*: (E) Se puede mejorar haciendo esta comprobación.
  - *Exactitud*: (BP) Se reduce la probabilidad de error de carga.

- *Los datos almacenados deben ser relevantes para el proceso de negocio que soportan*: Esto implica, no almacenar datos que no se van a usar. Este análisis debe ser efectuado en tiempo de diseño. Como hemos mencionado anteriormente, el uso mejora la calidad de los datos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (R) Si el dato no se usa tiende a no permanecer válido por mucho tiempo.
  - *Consistencia*: No aplica.
  - *Completitud*: (BP) Si no se usa, no se puede detectar falta de información.
  - *Exactitud*: (BP) Si no se usa no se puede detectar falta de correspondencia con el mundo real.

- *Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato almacenado este filtrado por estas reglas*: Esto se relaciona con lo dicho anteriormente ya que si se filtra la información por las reglas de negocio se puede detectar tempranamente falta de correspondencia entre el dato ingresado y lo esperado, evitando así el ingreso de basura. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (E) Esta aplicación de reglas favorece a que el dato ingrese más estable y tienda a no quedar desactualizado.
  - *Consistencia*: (E) De la misma manera ayuda a que el dato ingrese en el formato esperado.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) Esta práctica tiende a minimizar errores de ingreso de datos.

3.1.4 Ciclo de Vida del Dato: Visualización

En esta etapa, el dato es combinado para transformarse en información y representado de acuerdo al análisis de requerimientos para que el usuario final pueda interpretar la realidad y su situación. A continuación se clasifican como marco de trabajo la lista de prácticas a tener en cuenta en esta etapa y como afecta a las diferentes dimensiones del dato.

- *El sistema debe alertar sobre vencimientos*: De esta forma el responsable de los datos, que es el usuario referente de la aplicación o quién se haya designado, es avisado cuando de acuerdo a la lógica de la aplicación algún dato está por perder validez. En la Figura 3.24 se ve el proceso que valida el vencimiento de fechas y a través de esto se avisa al responsable de los datos para que tome las medidas que se establezcan dependiendo del dominio. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (E) Esta práctica alerta sobre la perdida de validez de un dato y ayuda al responsable a saber de esta situación y tomar acciones preventivas.
    - *Consistencia*: No aplica.
    - *Completitud*: No aplica.
    - *Exactitud*: (E) Al perder validez el dato se vuelve inexacto, por lo cual si esto se sabe con anticipación y se re valida, el riesgo de pérdida de correspondencia con la realidad disminuye.

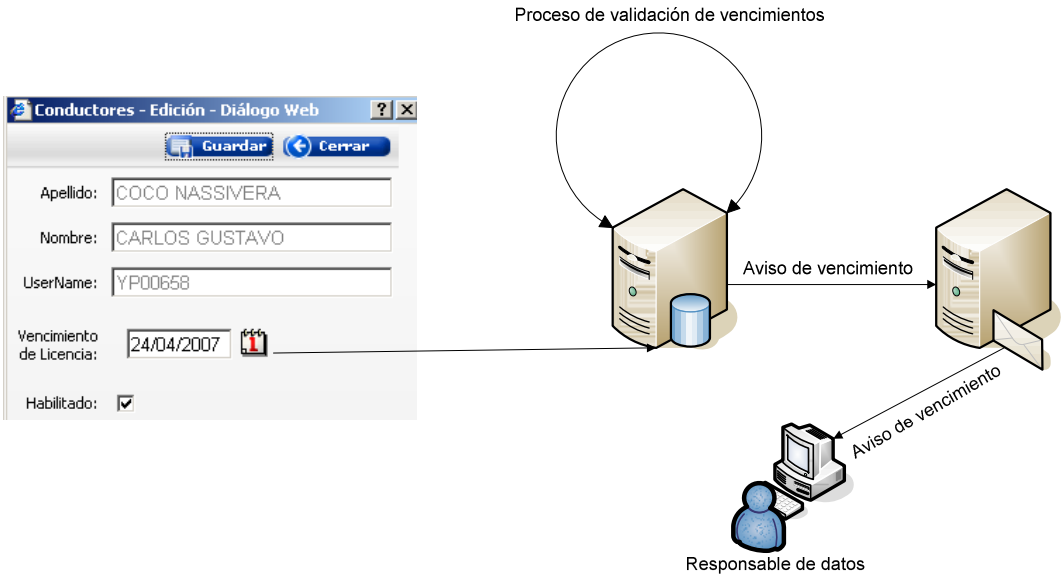

Figura 3.24 Proceso de control y alerta sobre vencimientos.

- *El sistema debe verificar y advertir cambios en la tendencia de los datos*: De esta forma, se pueden advertir en modo preventivo un cambio de tendencia. Para determinar si se trata de un error o de un cambio efectivo en la tendencia de la realidad, se requerirá un análisis funcional que

deberá realizar el dueño de los datos. Puede servir para detectar tempranamente errores en el registro del dato. En la Figura 3.25 se observa como el proceso controla y advierte al referente de estos desvíos para que los analice y tome las acciones correspondientes. Por ejemplo, si estamos midiendo el nivel de un tanque y se mantiene siempre en el orden de los 3.000 m^3 y un día se ve que el valor almacenado bajó a 1.000 m^3 y luego vuelve al orden de los 3.000 m^3, debe ser advertido por la aplicación y avisar al referente de los datos. Él sabrá si se debe a un proceso de negocio como por ejemplo un paro de la producción, o a un error en la captura del dato. Esto afecta a las cuatro dimensiones de la siguiente manera:

- *Temporalidad*: No aplica.
- *Consistencia*: No aplica.
- *Completitud*: No aplica.
- *Exactitud*: (E) Se alerta de desvíos en la tendencia del dato para tomar acciones preventivas destinadas a localizar falta de calidad en el registro del dato.

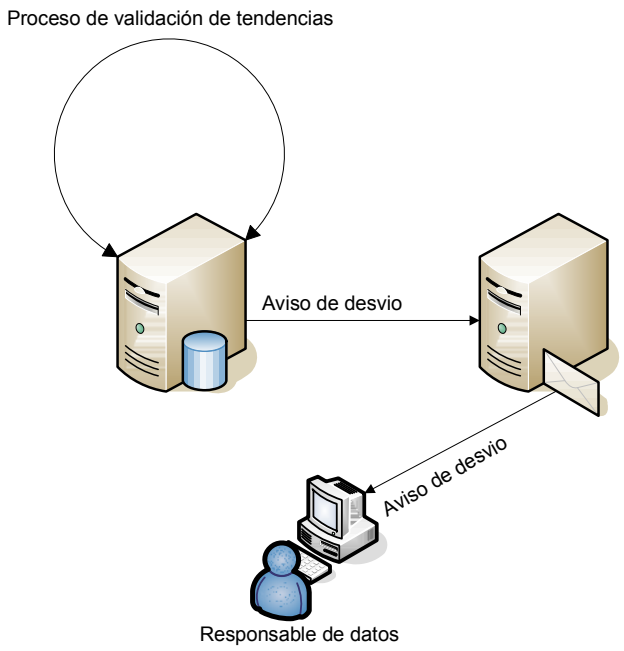

Figura 3.25 Proceso de control y alerta sobre tendencias en los datos.

- *El lenguaje de interface de usuario debe ser claro*: De esta forma se evitan confusiones que pueden llevar a malas interpretaciones de información. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: (BP) Se minimizan las confusiones que puedan tender a que el dato no se ingrese de forma completa.
  - *Exactitud*: (E) Minimizando la posibilidad de errores de interpretación, se favorece a que el dato mantenga su correspondencia con la realidad.

- *Los datos deben ser accesibles por herramientas de consulta de bases de datos*: De esta manera se facilita su uso y permite su masificación sin restricciones de interface. La Figura 3.26 muestra un ejemplo de esta práctica. Aquí se observa que se ha agregado al esquema habitual de la aplicación un servidor con herramientas de administración de datos. Así, un administrador de datos podrá acceder a ellos y publicar reportes. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (R) Permitirá que surjan más fácilmente las perdidas de validez de la información.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) Al permitir la masificación de los datos, surgirán más fácilmente las perdidas de validez de la información.

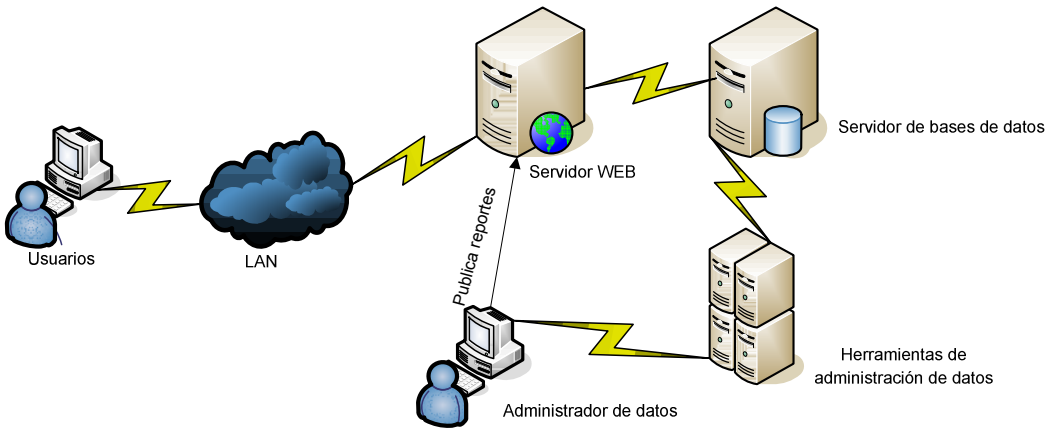

Figura 3.26 Los datos son accedidos por fuera de la aplicación web para mejorar su explotación.

- *Tienen que existir canales de retroalimentación para capturar propuestas de mejora*: Estos deben estar claros en la aplicación y se debe dejar a los usuarios finales conocer los mismos. Esto hará que se puedan recabar estas propuestas, algunas de las cuales pueden facilitar la tarea de mantener calidad en los datos. En la Figura 3.27 se ve un ejemplo de esto con un icono que permite enviar sugerencias y errores. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: (R) La comunicación con los actores del sistema en su ciclo de vida productivo ayuda a mejorar muchos aspectos del mismo, entre ellos esta dimensión de la calidad del dato.
    - *Consistencia*: (R) Se espera que de esta manera, se mejore también los aspectos que hacen a esta dimensión.
    - Completitud: (R) Al agregar canales de comunicación, se mejoran los aspectos de calidad que afectan a la completitud.

  Exactitud: (R) esta dimensión, se ve mejorada al implementar mejoras entre el usuario y el equipo de sistemas.

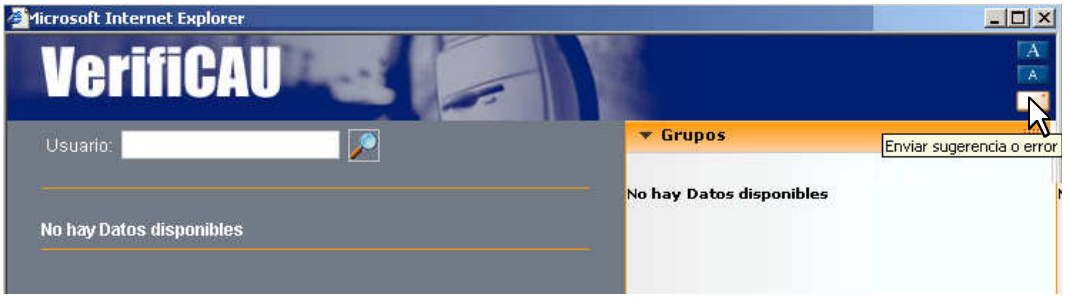

Figura 3.27 Canal de retroalimentación.

- *La aplicación debe permitir un análisis estadístico (grafico) para detectar anomalías*: De esta manera será más evidente la existencia de excepciones que luego de un análisis funcional se determinará si se está reflejando algún acontecimiento particular de la realidad del ámbito aplicativo o si se trata de un error en el dato. La Figura 3.28 muestra un ejemplo de esta práctica. Esto afecta a las cuatro dimensiones de la siguiente manera:
    - *Temporalidad*: No aplica.
    - *Consistencia*: No aplica.
    - *Completitud*: No aplica.
    - *Exactitud*: (BP) Permite hacer un seguimiento gráfico de la evolución y excepciones. Permitiendo detectar de forma rápida anomal-

ías en el registro del dato y tomar acciones correctivas rápidas si se trata de un error.

Figura 3.28 seguimiento gráfico para detectar desvíos en los datos.

- *Permitir opciones de visualización personales*: La personalización de reportes ayuda a los usuarios a extraer la información de la forma que les resulta más fácil su interpretación. De esta manera acortan la brecha entre el modelo conceptual diseñado y lo que les resulta práctico. Se espera que de esta forma se favorezca la calidad ya que el dato será más fácilmente interpretable. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.
  - *Completitud*: No aplica.
  - *Exactitud*: (R) El usuario podrá interpretar mejor la información y descubrir los errores en la misma más fácilmente.

- *Cuando el dato este codificado, debe ser interpretable a través de una regla*: De esta manera el dato y su metadato se hace visible, favoreciendo la interpretación del mismo. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) A partir de la correcta interpretación, el dato tenderá a permanecer consistente.
  - *Completitud*: No aplica.

  - *Exactitud*: (BP) Como se favorece la interpretación correcta del dato, este tiende a mantener su correspondencia con la realidad.

- *Las reglas de codificación de los datos codificados deben estar identificados en la interface de usuario*: Para hacer más fácil su interpretación y soportar la práctica anterior. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: (BP) A partir de la correcta interpretación, el dato tenderá a permanecer consistente.
  - *Completitud*: No aplica.
  - *Exactitud*: (BP) Como se favorece la interpretación correcta del dato, este tiende a mantener su correspondencia con la realidad.

- *El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)*: Esta práctica es más de negocio que de sistemas. Pero es una recomendación que debemos realizar al negocio cuando estamos observando un proceso que debiera alimentar o alimentarse de otros procesos de la compañía pero no lo hace. Se espera que de esta forma mejore la calidad, intensificando el uso de los datos. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: (R) Al intensificar el uso del dato, se espera que la volatilidad del dato sea rápidamente detectada.
  - *Consistencia*: (R) Se espera que al usar el más el dato, los problemas de consistencia se detecten más que si el dato no tiene uso.
  - *Completitud*: (R) De igual manera, este tipo de problema se mejorará también.
  - *Exactitud*: (R) Al igual que los puntos anteriores, se espera mejorar en este ítem a través de intensificar el uso del dato.

- *Las opciones de búsqueda deben considerar que el valor introducido este "contenido" en el campo de interés*: Esto ayuda a encontrar la información más rápidamente. En la Figura 3.29 se puede ver que al introducir un dato en la búsqueda de una dirección, la herramienta muestra dos coincidencias con el dato ingresado para que el usuario elija la que se adecua a su necesidad. Esto afecta a las cuatro dimensiones de la siguiente manera:
  - *Temporalidad*: No aplica.
  - *Consistencia*: No aplica.

- *Completitud*: No aplica.
- *Exactitud*: (R) Con esta práctica los resultados de la búsqueda tenderán a ser más adecuados, disminuyendo la posibilidad de error en la interpretación de los datos.

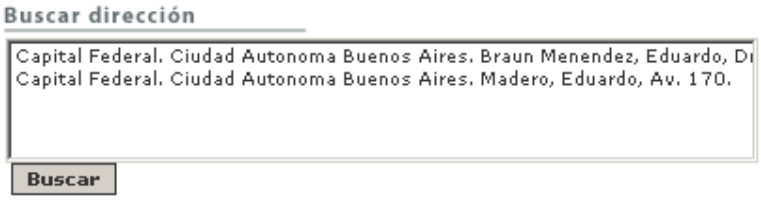

Figura 3.29 Antes de hacer la búsqueda, se corrobora la existencia del criterio en la base.

3.2 Mejora al Proceso de Desarrollo

Previamente el desarrollo de sistemas en la compañía donde realizamos este trabajo, desde la toma de requerimientos hasta la puesta en producción del mismo, estaba guiado por el proceso de desarrollo que se muestra en la Figura 3.30.

En el gráfico se observa como los actores que intervienen en el desarrollo de una aplicación interactúan. Estos actores son: Usuario, Proyectos, Gerencia de Proyectos y Programación.

El proceso se divide en 4 etapas que en el gráfico se han identificado con colores de la siguiente forma: en amarillo la etapa de análisis de factibilidad y relevamiento inicial. El color celeste se corresponde con la etapa de relevamiento detallado y diseño. En verde la etapa de construcción y desarrollo. El rosa es el color que identifica la etapa de implementación.

El proceso comienza con entrevistas de relevamiento donde el usuario plantea su problemática de negocio. Los integrantes del área de proyectos identifican en estas entrevistas las necesidades reales y proponen diferentes soluciones de tecnología de la información que se pueden aplicar. De estas reuniones, opcionalmente, se pueden generar documentos de *minutas de reunión*.

El equipo de proyectos se pregunta si conoce una solución corporativa que aplique a la necesidad. Si es así, se implementa. Con consenso de la Gerencia de proyectos. Si no se conoce una solución preestablecida, se elabora el *documento de visión* que enumera las necesidades del negocio, el objetivo de una solución informatizada y su alcance. Se consulta a dicha gerencia si existe una solución corporativa, si esta gerencia contesta que si, se implementa dicha

solución, si no se elabora un preproyecto de donde surge el análisis de costo. Se toman decisiones como por ejemplo si el desarrollo será realizado con la metodología local o la del departamento centralizado de desarrollo. De esto depende que se siga esta metodología o la definida por dicho departamento.

Si el desarrollo será local, se empieza con la etapa de relevamiento detallado y diseño. Para esto se identifican las historias que son un equivalente a los requisitos, casos de uso, y otras herramientas de análisis de requerimientos. La historia de usuario es una frase corta que representa alguna función que realizara el sistema. Estas se detallan en el *documento de historias* y también se diseña y documenta el *modelo de datos*.

El *documento de historias* se pasa al grupo de programación para que estime esfuerzo de desarrollo en tiempo por historia. Con esto el grupo de proyectos arma el cronograma y el plan de entregables, el mismo se valida con el usuario para ajustar prioridades y negociar tiempos de entrega teniendo como variable de ajuste el alcance de cada release. De estas entrevistas surgen como documentos el *cronograma* y el *release plan* que detalla cada entregable con su alcance.

Estos documentos se pasan a programación donde el grupo de desarrollo luego de ir avanzando con el cronograma devuelve el *código fuente* de la aplicación, la aplicación funcionando sin errores unitarios ni de integración (En programación, una prueba unitaria es una forma de probar la corrección de un módulo de código, esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego con las Pruebas de Integración se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.), el *manual de instalación* y continúa con la programación de la siguiente entrega.

El grupo de proyectos recibe dicha entrega y la instala en un servidor de prueba donde realiza las pruebas funcionales, en base a estas pruebas se generan *documentos de errores y mejoras detectadas* los que se utilizan para determinar si el sistema cumple con las funcionalidades mínimas requeridas para su puesta en producción, de no ser así se devuelve a desarrollo y se re planifica la entrega.

Si cumple, se instala en un ambiente de preproducción donde puede acceder el usuario referente a realizar pruebas de usuario, si aparecen correcciones o ajuste se re planifica y se devuelve a desarrollo. Si el usuario referente da su aprobación, pasa a un ambiente de producción donde lo pueden acceder todos los usuarios.

Como podemos observar en el mismo no existen actividades ni prácticas relacionadas a la calidad del dato.

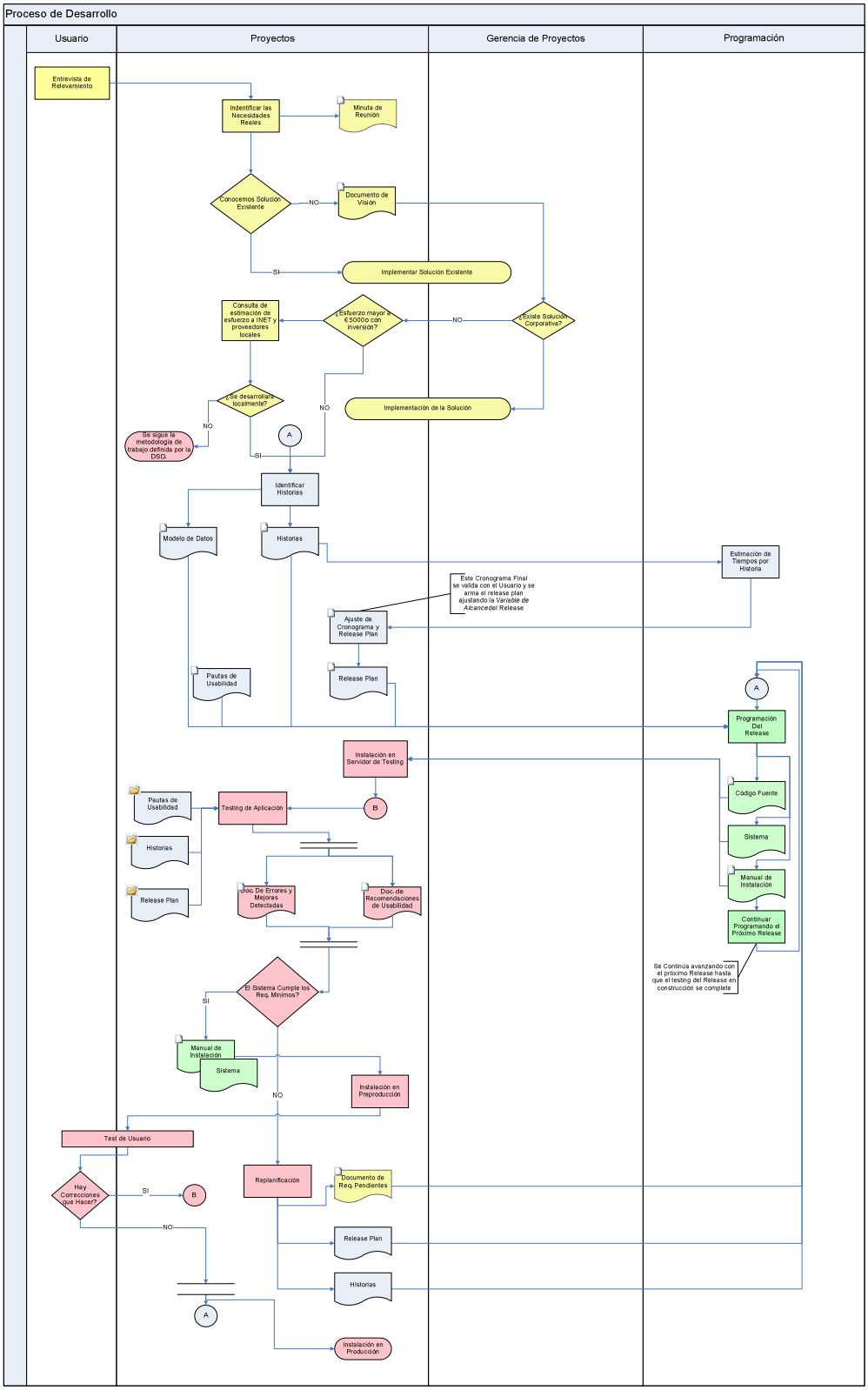

Figura 3.30 Proceso de desarrollo original.

En este trabajo, se modificó dicho proceso para que asegure que los sistemas consideren actividades y prácticas necesarias para mejorar la calidad de datos.

Estos cambios fueron motivados por la necesidad de formalizar los roles que se iban a desempeñar en un proceso de desarrollo orientado a lograr que las aplicaciones resultantes tengan un análisis orientado a la calidad del dato.

Para esto, se creo un grupo de trabajo para revisar, modificar e implementar el proceso, mejorándolo y haciendo la solución presentada en este trabajo exportable a todos los sistemas que se desarrollen en la empresa. Luego de aplicar los cambios, el proceso quedó representado como muestra la Figura 3.31. Los cambios están resaltados con un círculo rojo y se explican a continuación:

Se agrega a la documentación a pasar al grupo de programación las *pautas de calidad de datos* que son un compendio de las 64 prácticas antes explicadas excepto que alguna se desestime por una decisión de diseño.

Cuando la aplicación es entregada por el grupo de programación al grupo de proyectos, dentro de las pruebas funcionales que están establecidas, el Soporte Funcional, quién realizará las pruebas funcionales a la aplicación y la prueba de calidad de datos y una vez en producción dará soporte a las consultas de usuario y estará a cargo de la capacitación de usuario, toma estas pautas y las usa como guía para realizar las pruebas de calidad de datos. Utiliza el marco de trabajo para obtener los valores de error y la recomendación de aprobación o no del test.

En base a los resultados obtenidos, el Soporte Funcional, elaborará también el documento de *recomendaciones de calidad del dato*, donde dejará explicitada la descripción de las dimensiones que presentan errores, comentando todos los puntos en los que considere que no se respetaron las pautas o no se cumplen las recomendaciones y el motivo por el cual no encuadra con lo esperado. También se detallan las acciones para acceder o visualizar el dato en cuestión.

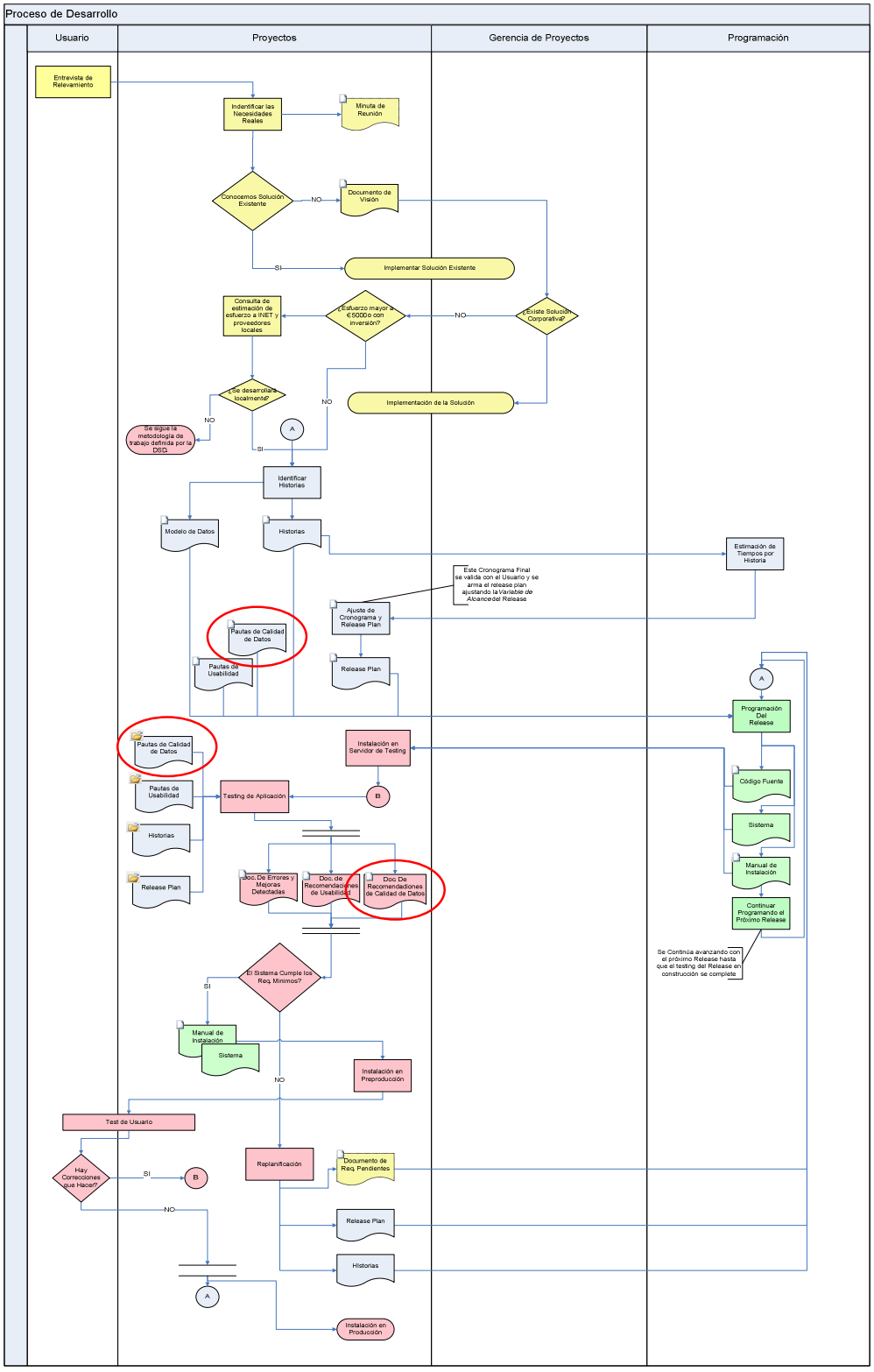

Figura 3.31 Proceso de desarrollo modificado para tener en cuenta la calidad del dato en la construcción de productos informáticos.

3.2.1 Procedimiento

Como derivado directo de este trabajo, se implementó en la empresa un procedimiento, cuyo objeto es promover y estandarizar las prácticas referentes a mejorar la calidad de los datos que soportan la toma de decisiones. El mismo establece partes y responsabilidades que se describen a continuación:

- **Usuario Referente**
    - Ofrecer información referente al proceso del cual los datos formarán parte.
    - Validar el modelo de datos propuesto para la solución informática.
    - Validar el informe del Testing de Calidad de Datos.

- **Proveedor de Aplicación**
    - Conocer las pautas de Calidad de Datos definidas.
    - Proveer la solución teniendo en cuenta los lineamientos definidos de Calidad de Datos.

- **Analista funcional**
    - Relevar los requerimientos del usuario teniendo en cuenta los lineamientos definidos de Calidad de Datos.
    - Diseñar la solución informática teniendo en cuenta todas las recomendaciones y consideraciones definidas.
    - Promover y proponer soluciones integradas a los procesos de negocio existentes, maximizando de esta forma la utilización de datos.
    - Seguimiento del cumplimiento de las normas definidas de Calidad de Datos en las diferentes entregas de los Proveedores de Aplicaciones.

- **Soporte funcional**
    - Conocer los distintos puntos con los que se testeará la Calidad de Datos de la aplicación.
    - Recomendar soluciones o buenas prácticas de acuerdo a su experiencia.
    - Realizar el test de Calidad de Datos según el marco de trabajo que se explica en la Sección 3.3.

El flujo de interacción entre los actores de este proceso se muestra gráficamente en la Figura 3.32.

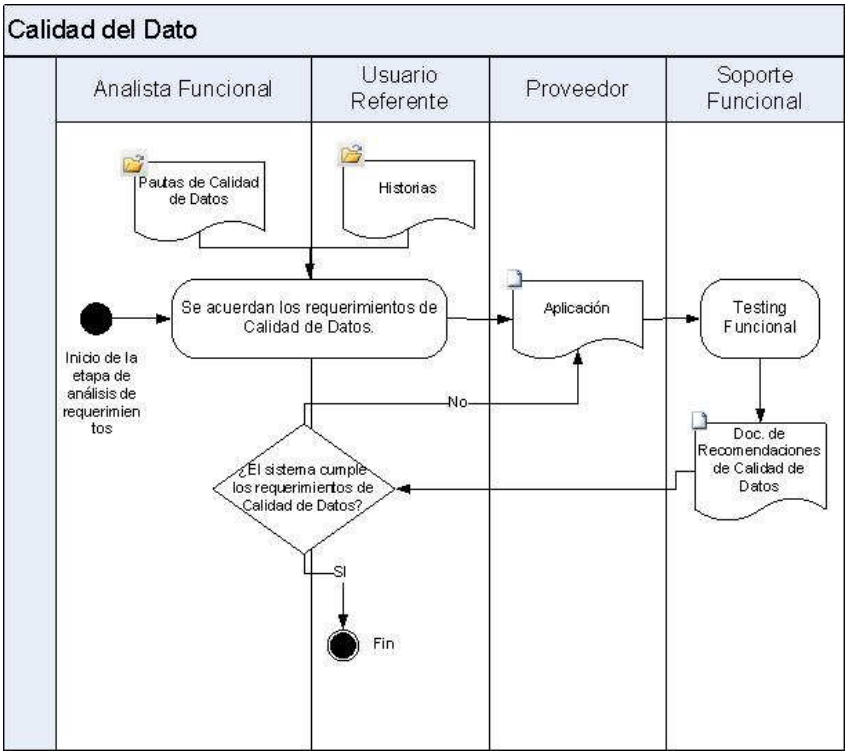

Figura 3.32 Flujo del proceso de calidad del dato en la construcción de aplicaciones.

Al comienzo del proceso de relevamiento y elicitación de requerimientos, el analista de la aplicación aporta las pautas de calidad que se describieron en el Capítulo 2 de este trabajo. El usuario referente aportará por su lado las historias que son su forma de representar los requerimientos del sistema, de acuerdo con la metodología de desarrollo existente en la empresa y cuyo análisis excede el alcance de este trabajo. Las historias de usuario son una descripción de las necesidades funcionales. Las mismas no deben ocupar muchas líneas, deben ser sencillas para comunicar que la necesidad se cumple.

Entre el analista funcional y el usuario referente, y de acuerdo a la definición de calidad adoptada por la empresa, *"La calidad es el conjunto de características propias de un producto, servicio, sistema o proceso imprescindibles para cumplir las necesidades o expectativas de partes interesadas"*, acuerdan cuales de estas prácticas (evaluando el peso y costo/beneficio de cada una) serán puestas en práctica en la aplicación en cuestión. Esto genera un documento con las pautas de calidad acordadas que se entrega al proveedor de aplicación, sea este el equipo de desarrollo o el proveedor de aplicación en ca-

so de de productos que ya están desarrollados. Cuando la aplicación está lista, el proveedor entrega la aplicación la cual se somete a análisis funcional. En este momento se aplica el testing de acuerdo a lo explicado en el punto 3.2 y como resultado, surge el documento de Recomendaciones de Calidad de Datos. Con este documento, entre el analista y el usuario referente, decidirán si la aplicación cumple con un mínimo de pautas para entrar a producción o si se deben re-enviar los resultados al proveedor para que mejore el producto antes de iniciar su ciclo productivo.

3.3 Testing de Aplicaciones

A continuación, explicaremos en detalle como se utiliza el marco de trabajo para realizar las pruebas de calidad de datos a las aplicaciones.

La forma de utilización consiste en recorrer la aplicación, teniendo como referencia las pautas de calidad definidas entre el analista y los usuarios referentes, además de las pautas de usabilidad, que excede el alcance de este trabajo. Además, las *historias* que son Casos de Usos más resumidos, definidos previamente como parte de la metodología de desarrollo y el *Release Plan* que es el detalle de los entregables con sus alcances.

En este marco de trabajo, cada práctica está ponderada de acuerdo a cada dimensión de la calidad del dato de acuerdo a la escala que se puede ver el la Figura 3.33.

Ponderación	
Estándar	3
Buena práctica	2
Recomendación	1
No aplica	0

Figura 3.33 Tabla de ponderación de las prácticas (Clasificación Objetiva).

Error	Puntos	Descripción
Sin Error	0	
No Aplica	0	
Leves	5	Situaciones que pueden disminuir la calidad del dato
Graves	15	Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea
Fatales	40	Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente.

Figura 3.34 Tabla de errores de las prácticas (Clasificación Subjetiva).

Esta ponderación objetiva se cruza con la apreciación del soporte funcional. Este, en el momento de testing, evalúa la aplicación de acuerdo al cumplimiento de cada práctica. Esta evaluación se efectúa de acuerdo a la escala de la Figura 3.34.

Luego, el valor de error se multiplica para cada práctica y cada dimensión por el de ponderación dando el valor de incumplimiento de cada práctica. Finalmente se suman los valores de todas las prácticas y este número se compara con el determinado por la Figura 3.35 obteniendo la recomendación.

	Mínimo	Máximo
Aprobado		45
Aprobado con Observaciones	46	119
Rechazado	120	

Figura 3.35 Tabla de rangos para la recomendación del marco de trabajo.

De esta manera, si el valor de error es menor a 46, el test se da por aprobado, si está entre 46 y 119 es aprobado con observaciones y si el valor de error es mayor a 120, la recomendación del test será rechazar la aplicación.

Estos valores fueron escogidos para que si una práctica estándar tiene un error fatal, (equivale a 40 x 3) de un valor de error de *rechazo*, y cualquier valor de error mayor dará una recomendación de rechazo también.

La *aprobación* se dará con cualquier valor de error menor o igual a 45, esto se estableció considerando que un error grave en una práctica estándar era lo máximo aceptable para aprobar.

La *aprobación con observaciones* será cualquier combinación de valores intermedios.

Estos resultados serán validados con el usuario referente ya que el rechazo implica volver a desarrollar y planificar las entregas del producto. Será una decisión consensuada ya que existe la posibilidad de que el usuario referente igualmente apruebe el pasaje a producción de la entrega y que los errores encontrados se solucionen en la siguiente etapa.

La plantilla del marco de trabajo para aplicar el test de calidad de datos a las aplicaciones se muestra en la Figura 3.36.

Aprobado

				Total Puntos:
Modelado del dato	**Temporal**	**Consister**	**Completit**	**Exactitud**
La administración de usuarios debe estar integrada al Active Directory	Sin Error	Sin Error	Sin Error	Sin Error
Debe existir una regla que desactive un usuario que ya no está en el active directory - (validar contra Vto en A.D.)	Sin Error			Sin Error
Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas	Sin Error			Sin Error
Se debe proveer una interfaz de actualización de datos personales de los usuarios (los ajenos a Active Directory).	Sin Error			Sin Error
Prohibir el borrado de padres cuando aun existen hijos	Sin Error		Sin Error	Sin Error
Las propiedades de configuración regional deben tomarse de la configuración de SO		Sin Error		Sin Error
Si el dato existe en un sistema fuente, tomarlo de la misma	Sin Error	Sin Error	Sin Error	Sin Error
Definir nivel de completitud de los datos			Sin Error	
Definir nivel de granularidad de los datos		Sin Error	Sin Error	Sin Error
Evitar, siempre que se pueda las claves auto numéricas.		Sin Error		
Evaluar la conveniencia que los datos sean escritos todos en mayúscula		Sin Error		
Evitar caracteres especiales en los nombres de las tablas de Bases de datos		Sin Error		
Acotar en lo posible el dominio como reglas dentro de la base de datos		Sin Error		Sin Error
Contemplar en diseño fechar las relaciones que pueden cambiar (fecha de alta, fecha de baja)	Sin Error			Sin Error
Debe existir análisis y clasificación de datos en función de su criticidad, para enfocar el esfuerzo en los más críticos	Sin Error	Sin Error	Sin Error	Sin Error
Deben existir mecanismos de registro de uso del dato (haciendo hincapié en los críticos) (métricas de uso)	Sin Error			Sin Error
Las interfases del sistema con otros, deben estar documentadas		Sin Error		
El modelo de datos debe estar disponible		Sin Error		
El diseño de la BD debe responder a un proceso de negocio para determinar el alcance de los datos almacenados	Sin Error	Sin Error	Sin Error	Sin Error
La etiqueta que describe al dato debe ser comprensible		Sin Error		
Se debe conocer como llegar al dato fuente (cuando el dato estraído de otro sistema)		Sin Error		Sin Error
El diseño de la BD debe respetar estándares para nomenclaturas de atributos		Sin Error		
La definición de los datos debe ser compatible con los estándares de la compañía		Sin Error		
Concensuar el modelo de datos a nivel esquemático con el usuario referente		Sin Error		
Concensuar la presentación de los datos con los consumidores del dato en cuestión		Sin Error	Sin Error	
Captura del valor	**Temporal**	**Consister**	**Completit**	**Exactitud**
Debe ofrecer ejemplos del tipo de datos que queremos que ponga en los formularios		Sin Error	Sin Error	
Los Campos Fecha deben tener su calendario asociado.		Sin Error	Sin Error	
Los Campos con formato particular (guiones, barras intermedias) deben tener una máscara.		Sin Error	Sin Error	Sin Error
Los Campos Fecha Nacimiento no pueden ser posteriores a la fecha del sistema operativo		Sin Error		Sin Error
Fechas relacionadas a una persona: no pueden superar un valor definido	Sin Error			Sin Error
Luego de la Modificación, debe existir un mensaje que advierta que las modificaciones fueron realizadas				Sin Error
Validar la repetición de datos (Ej.:11111111111111111111111) de acuerdo a reglas		Sin Error		Sin Error
Validar la existencia de caracteres "especiales" en campos donde no debiera		Sin Error		Sin Error
Evitar en lo posible la carga manual		Sin Error	Sin Error	Sin Error
Solicitar confirmación de Eliminación				Sin Error
El sistema debe informar al cerrar que se deben grabar los datos o de lo contrario se perderán los cambios	Sin Error		Sin Error	
Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción		Sin Error	Sin Error	Sin Error
El dato debe ser capturado lo mas cercano a la fuente	Sin Error	Sin Error	Sin Error	Sin Error
¿Existen datos tipados? Si es así, deben poder ser desplegados de un campo de selección		Sin Error	Sin Error	Sin Error
Los Campos TIPO deben tener validación para no ingresar nombres duplicados mediante índice UNICO		Sin Error		
Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez				Sin Error
El diccionario de datos debe estar disponible con descripción del dominio de cada dato y sus excepciones		Sin Error	Sin Error	
Evitar el uso de siglas en la interfaz de usuario		Sin Error		
Si el ingreso de un dato es erróneo, se debe informar en el momento y de forma visible y fácil de corregir.			Sin Error	Sin Error
Minimizar la trascripción del dato				Sin Error
El dato lo carga quién lo genera y de no ser posible, alguien que esté muy cerca (físicamente) de él				Sin Error
Optimizar la información mostrada en la pantalla de captura de datos				Sin Error
Almacenamiento	**Temporal**	**Consister**	**Completit**	**Exactitud**
Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar	Sin Error	Sin Error	Sin Error	Sin Error
Si existe una regla matemática de verificación del campo, la misma se debe utilizar		Sin Error	Sin Error	Sin Error
Los datos almacenados deben ser relevantes para el proceso de negocio que soportan	Sin Error		Sin Error	Sin Error
Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado filtrado por estas reglas. El mismo debe ser dinámico	Sin Error	Sin Error		Sin Error
Visualización	**Temporal**	**Consister**	**Completit**	**Exactitud**
El sistema debe alertar sobre vencimientos	Sin Error			Sin Error
El sistema debe verificar y advertir cambios en la tendencia de los datos				Sin Error
El lenguaje de interfaz de usuario debe ser claro			Sin Error	Sin Error
Los datos deben ser accesibles por fuera del sistema	Sin Error			Sin Error
Tienen que existir canales de retroalimentación para capturar propuestas de mejora	Sin Error	Sin Error	Sin Error	Sin Error
La aplicación debe permitir un análisis estadístico (grafico) para detectar anomalías				Sin Error
Permitir opciones de visualización personales				Sin Error
Cuando el dato este codificado, debe ser interpretable a través de una regla		Sin Error		Sin Error
La regla de codificación de los datos codificados deben estar identificados en la interfaz de usuario		Sin Error		Sin Error
El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)	Sin Error	Sin Error	Sin Error	Sin Error
Las opciones de búsqueda deben considerar que el valor introducido este "contenido" en el campo de interés				Sin Error

Figura 3.36 Marco de trabajo para aplicar el test de calidad de datos en las aplicaciones.

Todos los errores encontrados y las decisiones tomadas en base a estos quedarán documentados en el *Documento de Recomendaciones de Calidad de Datos*. En el mismo se detallan las recomendaciones que no se cumplen, como afectan a cada dimensión describiendo el error y el motivo por el cual no encuadra con la dimensión en cuestión.

3.4 Resumen

En este capítulo se presentaron las prácticas que surgen como recomendaciones para los sistemas de información. Y se mostró como la aplicación de las mismas determina el nivel de calidad de los datos en un sistema. Se definió una forma de clasificar a las prácticas en *"Estándar"*, *"Buena Práctica"* y *"Recomendación"*. Además, se definió una metodología de como valorizarlas dependiendo de cómo eran calificadas, es decir, sin error, no aplicar (por dedición del analista), error leve, grave o fatal.

Luego se describió cada práctica encuadrada en el ciclo de vida del dato y como afecta el cumplimiento de cada una a las cuatro dimensiones en las que se mide la calidad.

Finalmente analizamos el procedimiento de desarrollo de aplicaciones que quedó implementado, se explicaron los roles de los actores que intervienen y el flujo de trabajo establecido para elaborar productos de información que atienden a la problemática de la calidad de los datos. A su vez, se definió el marco de trabajo que se utiliza como herramienta para desarrollar el testing de calidad de datos a las aplicaciones antes de ponerlas en producción. Además, como de la misma deriva la recomendación de calidad de datos y los documentos que se generan luego de dicha pruebas.

Capítulo 4

En este capítulo se verá cómo abordar un proceso de puesta en marcha de un proyecto de software pensando en la calidad de los datos y se muestra un ejemplo del marco de trabajo puesto en práctica.

4.1 Caso de Estudio

Como caso de estudio se utilizó una entrega de una aplicación administra la electricidad de la compañía. Esta aplicación centraliza la información acerca de la electricidad generada por la compañía, la electricidad comprada y registra las ventas de energía eléctrica, como así también los datos de los equipos generadores de electricidad. La disponibilidad de esta información tiene, entre otros objetos, la generación de informes solicitados por la Secretaría de Energía de la Nación.

Antes de una solución informática, la información era mantenida en planillas, lo que dificultaba la generación de los informes a la Secretaría de Energía y otros entes reguladores. La generación manual de estos informes era compleja ya que había que corroborar grandes cantidades de datos para evitar inconsistencias. Los datos de la energía generada eran mantenidos localmente en cada planta y centralizados en forma manual por personal del área de energía eléctrica.

La solución informática, para evitar los problemas antes mencionados, se había comenzado a desarrollar cuando estábamos elaborando esta propuesta de mejora la calidad de los datos. Por lo que el procedimiento de calidad no fue aplicado desde el comienzo. Pese a lo cual, como nuestra metodología de trabajo contempla el desarrollo modular incremental, definimos que se iba a empezar a aplicar este método a partir de las entregas siguientes.

Para el desarrollo de este módulo y como era la primera vez que se aplicaba, se reunió a los desarrolladores, el analista, el soporte funcional de la aplicación y se los instruyo en los roles que cada uno asumiría para esta prueba del proceso modificado y orientado a la calidad del dato en las aplicaciones.

Antes de comenzar la programación de la entrega que se tomó para la prueba, se les entregó a los desarrolladores las 64 prácticas definidas, junto con la categorización de las mismas de acuerdo a las cuatro dimensiones del

dato y agrupadas de acuerdo al ciclo de vida del dato. Es decir, como están presentadas en el Capítulo anterior.

Cuando el equipo de desarrolló superó las pruebas unitarias, que en programación se refiere a una forma de probar la corrección de un módulo de código para asegurar que cada uno de los módulos funcione correctamente por separado, y las pruebas de integración para asegurar el correcto funcionamiento del sistema o subsistema en cuestión. Se le aplica al entregable un testing funcional. En este momento el soporte funcional toma la aplicación y ejecuta el test funcional, el test de usabilidad y el de calidad del dato, de acuerdo a la modificación implementada del proceso de desarrollo.

Para esto, el soporte funcional, recorre cada una de las recomendaciones y verifica su cumplimiento, registrando si detecta alguna falta o falla.

El marco de trabajo que se presenta en la Figura 4.1 es el del test que se le aplico a la entrega del modulo 1 del sistema de Gestión eléctrica. Este módulo atiende sólo la problemática de los distribuidores que identifica los gastos, la certificación de los mismos, los contratos con los distintos proveedores de energía eléctrica (Ente Provincial de Energía del Neuquén, Cooperativa eléctrica CALF, Cooperativa eléctrica COPELCO, Distribuidora de energía Mendoza EDEMSA), los consumos de energía eléctrica por unidades operativas, variación en porcentaje por período, desviación en el factor de potencia requerido por la distribuidora.

A continuación se explicará cada una de las observaciones encontradas tal y como se documentaron en el *“Documento de Recomendaciones de Calidad de Datos”.*

Aprobado con Observaciones

Total Puntos: 90

#	Modelado del dato	Tempora	Consiste	Completi	Exactitud	Puntos de Err
1	La administración de usuarios debe estar integrada al Active Directory	Sin Error	Sin Error	Sin Error	Sin Error	0
2	Debe existir una regla que desactive un usuario que ya no está en el active directory - (validar contra Vto en A.D.)	Sin Error			No Aplica	0
3	Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas	Leves			No Aplica	15
4	Se debe proveer una interfaz de actualización de datos personales de los usuarios (los ajenos a Active Directory).	No Aplica			No Aplica	0
5	Prohibir el borrado de padres cuando aun existen hijos	Sin Error		Sin Error	Sin Error	0
6	Las propiedades de configuración regional deben tomarse de la configuración de SO		No Aplica		No Aplica	0
7	Si el dato existe en un sistema fuente, tomarlo de la misma	Sin Error	Sin Error	Sin Error	Sin Error	0
8	Definir nivel de completitud de los datos			Sin Error		0
9	Definir nivel de granularidad de los datos		Sin Error	Sin Error	Sin Error	0
10	Evitar, siempre que se pueda las claves auto numéricas.		Sin Error			0
11	Evaluar la conveniencia que los datos sean escritos todos en mayúscula		No Aplica			0
12	Evitar caracteres especiales en los nombres de las tablas de Bases de datos		Sin Error			0
13	Acotar en lo posible el dominio como reglas dentro de la base de datos		Sin Error		Sin Error	0
14	Contemplar en diseño fechar las relaciones que pueden cambiar (fecha de alta, fecha de baja)	Sin Error			Sin Error	0
15	Debe existir análisis y clasificación de datos en función de su criticidad, para enfocar el esfuerzo en los más críticos	Sin Error	Sin Error	Sin Error	Sin Error	0
16	Deben existir mecanismos de registro de uso del dato (haciendo hincapié en los críticos) (métricas de	Leves			Sin Error	10
17	Las interfases del sistema con otros, deben estar documentadas		Sin Error			0
18	El modelo de datos debe estar disponible		Sin Error			0
19	El diseño de la BD debe responder a un proceso de negocio para determinar el alcance de los datos	Sin Error	Sin Error	Sin Error	Sin Error	0
20	La etiqueta que describe al dato debe ser comprensible		Sin Error			0
21	Se debe conocer como llegar al dato fuente (cuando el dato es traído de otro sistema)		Sin Error		Sin Error	0
22	El diseño de la BD debe respetar estándares para nomenclaturas de atributos		Sin Error			0
23	La definición de los datos debe ser compatible con los estándares de la compañía		Sin Error			0
24	Concensuar el modelo de datos a nivel esquemático con el usuario referente		Leves			10
25	Concensuar la presentación de los datos con los consumidores del dato en cuestión		Sin Error	Sin Error		0
26	Evitar que un dato esté duplicado en más de un sistema	Leves			Sin Error	5
27	Las interfases y las aplicaciones satélites deben usar la misma terminología que la fuente		Sin Error		Sin Error	0
#	**Captura del valor**	**Tempora**	**Consiste**	**Completi**	**Exactitud**	**Puntos de Err**
28	Debe ofrecer ejemplos del tipo de datos que queremos que ponga en los formularios		Sin Error	Sin Error		0
29	Los Campos Fecha deben tener su calendario asociado.		Sin Error	Sin Error		0
30	Los Campos con formato particular (guiones, barras intermedias) deben tener una máscara.		Sin Error	Sin Error	Sin Error	0
31	Los Campos Fecha Nacimiento no pueden ser posteriores a la fecha del sistema operativo		No Aplica		No Aplica	0
32	Fechas relacionadas a una persona: no pueden superar un valor definido	No Aplica			No Aplica	0
33	Luego de la Modificación, debe existir un mensaje que advierta que las modificaciones fueron realizadas				Sin Error	0
34	Validar la repetición de datos (Ej.:11111111111111111111111) de acuerdo a reglas		Sin Error		Sin Error	0
35	Validar la inexistencia de caracteres "especiales" en campos donde no debiera		Sin Error		Sin Error	0
36	Evitar en lo posible la carga manual		Sin Error	Sin Error	Sin Error	0
37	Solicitar confirmación de Eliminación				Sin Error	0
38	El sistema debe informar al cerrar que se deben grabar los datos o de lo contrario se perderán los cambios	Sin Error		Sin Error		0
39	Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción		No Aplica	No Aplica	Leves	15
40	El dato debe ser capturado lo mas cercano a la fuente	Sin Error	Sin Error	Sin Error	Sin Error	0
41	¿Existen datos tipados? Si es así, deben poder ser desplegados de un campo de selección		Sin Error	Sin Error	Sin Error	0
42	Los Campos TIPO deben tener validación para no ingresar nombres duplicados mediante índice UNICO		Sin Error			0
43	Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez				No Aplica	0
44	El diccionario de datos debe estar disponible con descripción del dominio de cada dato y sus excepci		Sin Error	Sin Error		0
45	Evitar el uso de siglas en la interfaz de usuario		Sin Error			0
46	Si el ingreso de un dato es erróneo, se debe informar en el momento y de forma visible y fácil de corr			Sin Error	Sin Error	0
47	Minimizar la trascripción del dato				Leves	10
48	El dato lo carga quién lo genera y de no ser posible, alguien que esté muy cerca (físicamente) de él				Sin Error	0
49	Optimizar la información mostrada en la pantalla de captura de datos				Sin Error	0
#	**Almacenamiento**	**Tempora**	**Consiste**	**Completi**	**Exactitud**	**Puntos de Err**
50	Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar	Sin Error	Sin Error	Sin Error	Sin Error	0
51	Si existe una regla matemática de verificación del campo, la misma se debe utilizar		Sin Error	Sin Error	Sin Error	0
52	Los datos almacenados deben ser relevantes para el proceso de negocio que soportan	Sin Error		Sin Error	Sin Error	0
53	Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado filtrado por estas reglas. El mismo debe ser dinámico	Sin Error	Sin Error		Sin Error	0
#	**Visualización**	**Tempora**	**Consiste**	**Completi**	**Exactitud**	**Puntos de Err**
54	El sistema debe alertar sobre vencimientos	Sin Error			Sin Error	0
55	El sistema debe verificar y advertir cambios en la tendencia de los datos				Leves	15
56	El lenguaje de interfaz de usuario debe ser claro			Sin Error	Sin Error	0
57	Los datos deben ser accesibles por fuera del sistema	Sin Error			Sin Error	0
58	Tienen que existir canales de retroalimentación para capturar propuestas de mejora	Sin Error	Sin Error	Sin Error	Sin Error	0
59	La aplicación debe permitir un análisis estadístico (grafico) para detectar anomalías				Leves	10
60	Permitir opciones de visualización personales				Sin Error	0
61	Cuando el dato este codificado, debe ser interpretable a través de una regla		Sin Error		Sin Error	0
62	La regla de codificación de los datos codificados deben estar identificados en la interfaz de usuario		Sin Error		Sin Error	0
63	El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)	Sin Error	Sin Error	Sin Error	Sin Error	0
64	Las opciones de búsqueda deben considerar que el valor introducido este "contenido" en el campo d				Sin Error	0

Figura 4.1 Marco de trabajo con el resultado del test de calidad de datos.

4.1.1 Ciclo de Vida del Dato: Modelado del Dato

1. *La administración de usuarios debe estar integrada al Active Directory*: Se verificó que la administración este realmente integrada con active directory e implementada con el grupo de seguridad lógica de la compañía.
2. *Debe existir una regla que desactive un usuario que ya no está en el Active Directory*: Se encontró que en esta etapa del desarrollo incremental de la aplicación, donde no se ha masificado su uso, implementar este control era antieconómico por la relación costo/beneficio. Por lo que se decidió que no aplicaba.
3. *Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas*: En la dimensión temporalidad se consideró un error “Leve”, ya que no se controlaba. Igualmente, la cantidad de usuarios en esta etapa del ciclo de vida de la aplicación no justificaba que se considere como un error de mayor envergadura. En Exactitud se colocó “No aplica” por la razón antes explicada. De esta forma quedó documentado para ser tenido en cuenta en las siguientes entregas del desarrollo.
4. *Se debe proveer una interface de actualización de datos personales de los usuarios (los ajenos a Active Directory)*: No aplica porque no hay datos ajenos al Active Directory que se traten dentro de la aplicación.
5. *Prohibir el borrado de padres cuando aun existen hijos*: Sin error, se comprobó que la integridad referencial se encontraba implementada en la base de datos.
6. *Las propiedades de configuración regional deben tomarse de la configuración del sistema operativo*: Se colocó “No aplica” en las dos dimensiones afectadas por esta práctica, consistencia y exactitud. Ya que fue requerimiento de los usuarios que siempre el *“.”* sea usado como separador decimal.
7. *Si el dato existe en un sistema fuente, tomarlo de la misma*: En la aplicación existen datos que están en sistemas fuentes como los contratos de comercialización que se llevan con SAP. El sistema tiene una interface con este por lo que se pudo verificar que no hay error.
8. *Definir nivel de completitud de los datos*: El mismo fue definido en el modelo de datos, por lo que se verifica sin error.
9. *Definir nivel de granularidad de los datos*: El mismo fue definido en el modelo de datos, por lo que se verifica sin error.

10. *Evitar, siempre que se pueda las claves auto numéricas*: Se encontró que las tablas se encontraban normalizadas y que se habían utilizado atributos unívocos propios de la entidad como índice.
11. *Evaluar la conveniencia que los datos sean escritos todos en mayúscula*: En este módulo los datos son numéricos y los que no, son importados de SAP por lo que se consideró que la práctica no aplicaba.
12. *Evitar caracteres especiales en los nombres de las tablas de Bases de datos*: Se verificó y se consideró sin error.
13. *Acotar en lo posible el dominio como reglas dentro de la base de datos*: Se verificó el cumplimiento de la práctica y se consideró sin error. Por ejemplo, el cálculo de distribución de gastos de consumo eléctrico se realiza en base a una regla de negocio que es la cantidad de pozos activos en el caso de los campos productores de petróleo. Esta regla está integrada a la aplicación.
14. *Con respecto a contemplar en diseño fechar las relaciones que pueden cambiar (fecha de alta, fecha de baja)*: Se corroboró que esto estuviera consensuado y relevado con el usuario. Por lo tanto se consideró sin error.
15. *Debe existir análisis y clasificación de datos en función de su criticidad, para enfocar el esfuerzo en los más críticos*: Se consideró sin error ya que el mismo está en etapa de desarrollo para ser implementado en un módulo posterior.
16. *Deben existir mecanismos de registro de uso del dato (haciendo hincapié en los críticos) (métricas de uso)*: Se le asignó un error leve en temporalidad y sin error en exactitud porque esta todavía en etapa de desarrollo. Aunque no estaba disponible en ese momento.
17. *Las interfases del sistema con otros, deben estar documentadas*: Se verificó y se consideró sin error.
18. *El modelo de datos debe estar disponible*: Se verificó y se consideró sin error.
19. *El diseño de la base de datos debe responder a un proceso de negocio para determinar el alcance de los datos almacenados*: Se verificó y se consideró sin error.
20. *La etiqueta que describe al dato debe ser comprensible*: Se verificó y se consideró sin error. Por ejemplo, el nombre de los entes que proveen energía eléctrica esta etiquetado como “RazonSocial”, el código del contrato de servicio en la aplicación fuente SAP está identificado en la aplicación como “codigoSAP”.
21. *Se debe conocer como llegar al dato fuente (cuando el dato es traído de otro sistema)*: Se verificó y se consideró sin error. Por ejemplo, en

el caso antes comentado del contrato de servicio que está en SAP, está documentado como se obtiene el contrato SAP mediante el código.

22. *El diseño de la base de datos debe respetar estándares para nomenclaturas de atributos*: Se verificó y se consideró sin error.
23. *La definición de los datos debe ser compatible con los estándares de la compañía*: Se verificó y se consideró sin error.
24. *Consensuar el modelo de datos a nivel esquemático con el usuario referente*: Se aplicó un error leve en consistencia ya que se pudo constatar que esta instancia no se había efectuado.
25. *Consensuar la presentación de los datos con los consumidores del dato en cuestión*: Se verificó y se consideró sin error. El registro de esto quedó en las minutas de reunión que se realizaron.
26. *Evitar que un dato esté duplicado en más de un sistema*: Se puso un valor de error leve en temporalidad y no aplica en exactitud. Ya que por disposiciones de seguridad informática los datos tomados del sistema fuente SAP, con el cual se conecta esta aplicación no pueden ser accedidos en línea sino que se exportan una vez al día a los sistemas satélites. Por lo cual la temporalidad puede hacer que el dato no sea valido como máximo por 24 horas. Pese a lo cual para este desarrollo se acordó que no afectaría a la exactitud ya que los datos importados son poco dinámicos.
27. *Las interfases y las aplicaciones satélites deben usar la misma terminología que la fuente*: Se verificó y se consideró sin error. Por ejemplo, en el caso de SAP, se utilizó la misma terminología para los datos que la aplicación toma de esta.

4.1.2 Ciclo de Vida del Dato: Captura del Valor

28. *Debe ofrecer ejemplos del tipo de datos que queremos que ponga en los formularios*: Se verificó y se consideró sin error. Por ejemplo en la pantalla de carga de la aplicación donde debe transferirse información de una planilla impresa, se ha graficado la planilla y señalado donde está cada dato a incorporar en la aplicación a modo de ayuda.
29. *Los Campos Fecha deben tener su calendario asociado*: Se verificó y se consideró sin error.
30. *Los Campos con formato particular (guiones, barras intermedias) deben tener una máscara*: Se verificó y se consideró sin error.
31. *Los Campos Fecha Nacimiento no pueden ser posteriores a la fecha del sistema operativo*: No existen campos de este tipo, se considera un no aplica.

32. *Fechas relacionadas a una persona: no pueden superar un valor definido*: No existen campos de este tipo, se considera un valor de no aplica.
33. *Luego de la Modificación, debe existir un mensaje que advierta que las modificaciones fueron realizadas*: Se verificó y se consideró sin error.
34. *Validar la repetición de datos (Ej.:111111111111111111111) de acuerdo a reglas realizadas*: Se verificó y se consideró sin error.
35. *Validar la inexistencia de caracteres "especiales" en campos donde no debiera*: Se verificó y se consideró sin error.
36. *Evitar en lo posible la carga manual*: Se verificó y se consideró sin error. Siempre que el dato existe en algún otro sistema de la compañía, el mismo se importa de la fuente.
37. *Solicitar confirmación de Eliminación*: Se verificó y se consideró sin error.
38. *El sistema debe informar al cerrar que se deben grabar los datos o de lo contrario se perderán los cambios*: Se verificó y se consideró sin error.
39. *Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción*: Se consideró un error leve para la exactitud y no aplica para consistencia y completitud ya que las mismas no se ven afectadas porque la redundancia está controlada por la base de datos.
40. *El dato debe ser capturado lo mas cercano a la fuente*: Se verificó y se consideró sin error. Así está definido en el proceso de negocio que el sistema soporta.
41. *¿Existen datos tipados?. Si es así, deben poder ser desplegados de un campo de selección*: Se verificó y se consideró sin error. Por ejemplo están tipados los distribuidores, el número de medidor, el tipo de tarifa, el tipo de comprobante, la moneda de los reportes, etc.
42. *Los Campos TIPO deben tener validación para no ingresar nombres duplicados mediante índice UNICO*: Se verificó y se consideró sin error.
43. *Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez*: En esta práctica se puso un valor de no aplica en la dimensión exactitud, ya que hay circuitos y cadenas de aprobación definidos para los datos más críticos. Como por ejemplo, cuando se carga el valor de una factura de distribuidor de un determinado proveedor de energía eléctrica, se dispara un mail a la gente de “Cuen-

tas a Pagar" quienes con una copia de la factura verifican que el valor sea correcto y aprueban el pago en la aplicación.

44. *El diccionario de datos debe estar disponible con descripción del dominio de cada dato y sus excepciones*: Se verificó y se consideró sin error.
45. *Evitar el uso de siglas en la interface de usuario*: Se verificó y se consideró sin error.
46. *Si el ingreso de un dato es erróneo, se debe informar en el momento y de forma visible y fácil de corregir*: Se verificó y se consideró sin error.
47. *Minimizar la trascripción del dato*: Se aplico un valor de error leve en la dimensión exactitud, ya que está en análisis si el costo/beneficio justifica una forma de recolección de datos más segura que la manual.
48. *El dato lo carga quién lo genera y de no ser posible, alguien que esté muy cerca (físicamente) de él*: Se verificó y se consideró sin error.
49. *Optimizar la información mostrada en la pantalla de captura de datos*: Se verificó y se consideró sin error.

4.1.3 Ciclo de vida del Dato: Almacenamiento

50. *Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar*: Se verificó y se consideró sin error.
51. *Si existe una regla matemática de verificación del campo, la misma se debe utilizar*: Se verificó y se consideró sin error.
52. *Los datos almacenados deben ser relevantes para el proceso de negocio que soportan*: Se verificó y se consideró sin error.
53. *Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado filtrado por estas reglas*. *El mismo debe ser dinámico*: Se verificó y se consideró sin error.

4.1.4 Ciclo de Vida del Dato: Visualización

54. *El sistema debe alertar sobre vencimientos*: Se verificó y se consideró sin error. Por ejemplo, una vez cargada la factura de distribución, la gente de "cuentas a pagar" tiene 5 días hábiles para pagar, la aplicación verifica este vencimiento y va alertando del mismo.
55. *El sistema debe verificar y advertir cambios en la tendencia de los datos*: Se colocó error leve en exactitud ya que está en desarrollo y se espera implementarlo en una etapa más avanzada de la implementación.

56. *El lenguaje de interface de usuario debe ser claro*: Se verificó y se consideró sin error.
57. *Los datos deben ser accesibles por fuera del sistema*: Se verificó y se consideró sin error. Los datos se pueden acceder por herramientas de consulta de datos ya que la base de datos es SQL Server y el modelo está documentado.
58. *Tienen que existir canales de retroalimentación para capturar propuestas de mejora*: Se verificó y se consideró sin error. Esto es cubierto por el rol del soporte funcional que acompaña a la aplicación en su ciclo de vida productivo.
59. *La aplicación debe permitir un análisis estadístico (grafico) para detectar anomalías*: Se colocó error leve en exactitud ya que está en desarrollo y se espera implementarlo en una etapa más avanzada de la implementación
60. *Permitir opciones de visualización personales*: Se verificó y se consideró sin error. La aplicación tiene la capacidad de personalizar reportes.
61. *Cuando el dato este codificado, debe ser interpretable a través de una regla*: Se verificó y se consideró sin error.
62. *La regla de codificación de los datos codificados deben estar identificados en la interface de usuario*: Se verificó y se consideró sin error.
63. *El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)*: Se verificó y se consideró sin error. De hecho el usuario referente es partidario de la sinergia entre áreas. Valoró que Sistemas de Información tenga entre sus recomendaciones el verificar que esto se cumpla. Esta aplicación que es del área “Ingeniería de Petróleo Gas y Electricidad” será también usada por el área “Cuentas a Pagar” gracias a esta cultura de procesos abiertos.
64. *Las opciones de búsqueda deben considerar que el valor introducido este "contenido" en el campo de interés*: Se verificó y se consideró sin error. De forma tal que si se buscan los proveedores y como parámetro se introduce por ejemplo “C” Aparecen como resultado de esta búsqueda “CALF” y “COPELCO”

Estos son los puntos que se analizaron y documentaron. La recomendación final fue: *“Aprobar con observaciones”*. Luego, se decidió pasar a producción la aplicación, documentar estas observaciones, analizarlas y mejorar en la siguiente entrega modular de la aplicación.

4.2 Resumen

En este capitulo se aplicó la metodología propuesta en el Capítulo 3 sobre el testing de aplicaciones a un caso de estudio real para encontrar posibles mejoras orientadas a la calidad del dato.

Se describieron cada una de las observaciones encontradas detallando las decisiones que se tomaron basados en el marco de trabajo. Finalmente y como conclusión se determinó la recomendación final que surgió del análisis mismo y que resultó en una *Aprobación con observaciones*. Estas observaciones estarán detalladas en el punto 4.1.

Con respecto a la experiencia de utilizar este marco de trabajo, encontramos que el mismo fue de gran utilidad. Tanto los desarrolladores que recibieron las recomendaciones al inicio del desarrollo como el Soporte Funcional que realizó el test, encontraron que estas prácticas iban a resultar de utilidad para que el sistema no deje entrar datos de baja calidad en la aplicación.

Conclusiones

Hemos definido a la calidad de los datos como un punto de acuerdo entre las partes interesadas, es decir, las características que un producto debe cumplir para satisfacer las expectativas de los interesados.

Los datos de las organizaciones, que estén en medios de tecnología de la información, son propensos a dejar de satisfacer las necesidades de dichas partes rápidamente.

A partir de la observación de este efecto, intercambio de opiniones con usuarios, analistas y responsables de los datos; concluimos en que las aplicaciones generalmente permiten que datos de baja calidad ingresen, vivan y se reproduzcan dentro de los recursos informáticos.

Hemos introducido varias definiciones de calidad de datos la cuales convergen en que la calidad del dato está en función de su uso y que existen reglas para que los datos se mantengan con calidad. De esta forma, hemos creado una metodología para utilizar como guía de recomendaciones a aplicar en los sistemas que se desarrollan, permitiendo evaluar los sistemas de acuerdo a la forma en que están construidos con respecto a la calidad e los datos que manejarán.

Esta metodología debe ser parte integral de una organización, del grupo de desarrollo y de la mentalidad de sus componentes. Para lo cual es necesario mejorar los procesos profesionales y la manera de trabajar; dar a la planificación el lugar que se merece y producir un cambio de cultura en aquellos que aún dicen: *“Estamos en tiempos de crisis y me vienen a hablar de lo que deberíamos hacer para mejorar o de perder tiempo planificando cuando lo que yo quiero es sacar soluciones YA”*. Nadie discute que en estos tiempos los vientos están soplando muy duro, pero en las tormentas están quienes se ocultan en refugios, poniéndose a resguardo y hay quienes montan molinos de viento y recogen éxitos permanentes. Es necesario persuadir a los empresarios de que los beneficios de medidas preventivas son tangibles.

La calidad de los datos no debe ser un agregado a las aplicaciones, sino algo que surja desde el propio diseño.

Durante la elaboración del presente trabajo observábamos que las aplicaciones que se utilizan como solución de problemáticas en el negocio, esta-

ban con la debilidad de ser propensos a permitir que los datos sean de baja calidad. Es decir, observábamos que nuestros sistemas permitían que los datos sean de mala calidad.

Basándonos en los trabajos relacionados desarrollados en el punto 2.4, además de las prácticas que recolectamos de la experiencia y del análisis de las aplicaciones que tenemos en la compañía, nos habíamos planteado el objetivo de definir un marco de trabajo general analizando el ciclo de vida del dato que permita la evaluación y mejora de la calidad del dato. Para esto, varias tareas fueron realizadas:

- relevamos las técnicas que favorecen la calidad de los datos durante su ciclo de vida,
- clasificamos las técnicas en recomendaciones, buenas prácticas o estándares,
- realizamos un marco conceptual que permite pensar, diseñar, desarrollar y probar los sistemas teniendo en cuenta prácticas, y estándares que favorezcan la calidad de los datos y
- aplicamos el marco conceptual definido a un sistema real dentro de una empresa a modo de prueba piloto para empezar el ciclo de mejora continua (ya que todo proceso debe ser constantemente revisado y mejorado).

La propuesta se desarrollo detallando la lista de pautas para mejorar los sistemas enmarcando las prácticas en las dimensiones en las que se evalúa la calidad del dato. Las mismas se agruparon de acuerdo a las diferentes etapas el ciclo de vida del dato. Se explicó la forma en que se clasificaron de acuerdo al criterio consensuado del grupo de trabajo junto con una descripción de cada una y a su afectación a las diferentes dimensiones de la calidad.

También se modificaron los procesos existentes para adaptarlos a una nueva forma de trabajo. Se volvieron a definir flujos formales para que se puedan implementar el nuevo proceso.

Finalmente se detalló la forma de utilizar este marco de trabajo, desarrollando un ejemplo de su aplicación.

Creemos que el modelo desarrollado es aplicable en las empresas del entorno que seguramente tienen, quizá con conciencia o no, los mismos problemas que nosotros habíamos detectado. Estos problemas se deben en su mayoría al poco cuidado de la calidad del dato desde su concepción. Se toma erróneamente, a la calidad preventiva como un costo y no como una inversión.

Es necesario que las empresas que desarrollan software cambien su cultura respecto a la calidad del dato, como se ha venido dando con la calidad de los procesos en los últimos años. Aquí jugamos nosotros, los profesionales de sistemas de información, un papel fundamental ya que debemos mostrar a las empresas los beneficios de tomar conciencia de estas prácticas en tiempo de diseño, eligiendo pagar ahora en lugar de pagar más adelante más.

Bibliografía

[1] G. Brackstone.
Managing data quality in a statistical agency.
Survey Methodology, (25):139-179, 1999.

[2] E. M. Burns, O. MacDonald, and A. Champaneri.
Data quality assesment methodology: A framework. In Joint Statistical Meetings Section on Government Statistics, pages 334-337, 2000.

[3] K. Orr.
Data quality and systems theory.
Communications of the ACM, 41(2):66-71, February 1998.

[4] E. Pierce.
Assesing data quality with control matrices.
Communications of the ACM, 47(2):82-86, February 2004.

[5] T. Redman.
The impact of poor data quality on the typical enterprise.
Communications of the ACM, 41(2):79-83, February 1998.

[6] T. Redman.
Data Quality: The Field Guide. Digital Press, January 15 2001.

[7] G. Tayi and D. Ballou.
Examining data quality.
Communications of the ACM, 41(2):54-57, February 1998.

[8] R.Wang.
A product perspective on total data quality managment.
Communications of the ACM, 41(2):58{65, February 1998.

[9] Manuel Serrano, Ismael Caballero, Coral Calero, Mario Piattini
Calidad de los Almacenes de Datos
Grupo de Investigación Alarcos, E.S. Informática Editorial: I+D Computación, Vol. 2, No. 2, Julio 2003

[10] G. Shankaranarayanan, Richard Y. Wang
Representing the Manufacture of an Information Product
Department of Management Information Systems.

[11] LA TOMA DE DECISIONES ESTA BASADA EN EL ANÁLISIS DE LOS DATOS Y LA INFORMACION
http://www.tuobra.unam.mx/publicadas/040921162449.html

[12] 7 Herramientas básicas para el control de calidad
http://www.monografias.com/trabajos7/herba/herba.shtml

[13] FireWall para los datos de mala calidad
http://javierdequiros.blogspot.com/2005_03_01_javierdequiros_archive.html

[14] Sarbanes-Oxley
http://www.deloitte.com/dtt/section_node/0,1042,sid%253D96325,00.html

[15] Diagrama de Pareto
http://www.gestiopolis.com/recursos/documentos/fulldocs/eco/diagramapareto.htm

[16] Modelo de datos
http://es.tldp.org/Tutoriales/NOTAS-CURSO-BBDD/notas-curso-BD/node18.html