# DQE: Una Herramienta para Evaluar la Calidad de los Datos en un Sistema de Integración

**Fabián Fajardo, Ignacio Crispino, Verónika Peralta**
Instituto de Computación, Universidad de la República, URUGUAY
ffajardo@adinet.com.uy, ignacioc@internet.com.uy, vperalta@fing.edu.uy

**Resumen:** Los sistemas de información actuales necesitan integrar grandes cantidades de información de múltiples fuentes de datos para resolver requerimientos complejos de los usuarios. Un desafío en este tipo de sistemas es proveer al usuario con información adaptada a sus requerimientos de calidad. La calidad se expresa como un conjunto de factores de calidad que miden ciertos aspectos relevantes de los resultados, como la frescura de los datos, la completitud o el tiempo de respuesta. Este artículo trata el problema de evaluar la calidad en un sistema de integración de datos. Concretamente, se presenta una herramienta que permite medir y comparar la calidad de la información devuelta al usuario. La herramienta permite modelar el sistema de integración, sus propiedades y ejecutar algoritmos de evaluación especializados en la medición de algunos factores de calidad. La herramienta es flexible y generalizable, permitiendo la selección de los factores de calidad más relevantes así como la incorporación dinámica de nuevos factores y nuevos algoritmos de evaluación. Actualmente está siendo utilizada para el "test" de diferentes algoritmos de evaluación y la comparación de los resultados producidos por éstos. Para ilustrar nuestro enfoque, presentamos un caso de estudio y mostramos la evaluación de un factor de calidad: la frescura de los datos.

**Palabras claves:** calidad de datos, evaluación de la calidad, algoritmos de evaluación

## 1. Introducción

Los avances tecnológicos de los últimos años en materia de comunicaciones han permitido el desarrollo de sistemas de información de gran porte que brindan acceso a grandes volúmenes de información. La necesidad de acceder en forma uniforme a la información disponible en múltiples fuentes de datos, ya sean internas a una organización o accesibles a través de Internet, es cada vez más fuerte y generalizada. Dichos requisitos de información son generalmente resueltos implementando complejos procesos de manipulación de datos que implican vistas o consultas sobre fuentes de datos heterogéneas y autónomas. A medida que aumenta la cantidad de datos potencialmente recuperados, los usuarios se interesan más y más en la calidad de los resultados. Debido a la heterogeneidad de las fuentes de datos resulta difícil evaluar la calidad de los datos para brindar a los usuarios respuestas uniformes y de alta calidad.

Este artículo trata el problema de evaluar la calidad de la información producida por un *Sistema de Integración de Datos* (SID). La calidad de la información devuelta al usuario depende principalmente de la calidad de las fuentes de datos y de las características del *proceso de cálculo* que construye dicha información a partir de las fuentes. Más concretamente, la calidad depende de la calidad interna de las fuentes (la coherencia, la completitud, la frescura, etc.), de la confianza sobre quién produce los datos de esas fuentes, y también de la forma de producir la información devuelta al usuario. En un contexto en donde la información es producida por algoritmos sofisticados de agregación, la evaluación de la calidad requiere un conocimiento fino del proceso de producción. Además, la heterogeneidad de las fuentes de datos (por ejemplo diferentes formatos o semántica de los datos) agrega complejidad a la evaluación.

La información devuelta al usuario puede ser diferente dependiendo de la forma de producirla, es decir, de las fuentes de donde se extraen los datos y de las operaciones realizadas para integrar dichos datos. Los diferentes procesos de cálculo pueden ser estudiados y comparados para seleccionar la mejor implementación del sistema de integración de acuerdo a las necesidades de los

usuarios. La calidad de los datos producidos por cada proceso es un elemento importante para realizar dicha comparación.
La calidad se expresa mediante un conjunto de *factores de calidad*, los cuales miden ciertos aspectos del resultado que son de importancia para los usuarios, como por ejemplo, la frescura de los datos, el tiempo de respuesta o la disponibilidad de las fuentes. Los factores más relevantes para un determinado sistema de integración dependen de las necesidades de los usuarios y de sus aplicaciones. La evaluación de la calidad se realiza mediante la ejecución de *algoritmos de evaluación*, cada uno especializado en la medición de un factor de calidad.
En este artículo presentamos una herramienta que permite medir y comparar la calidad de los datos devueltos por cada proceso. La herramienta propuesta es flexible y generalizable, permitiendo la selección de los factores de calidad más relevantes así como la incorporación dinámica de nuevos factores y nuevos algoritmos de evaluación. La herramienta puede ser usada para: (i) evaluar la calidad de los datos devueltos en un sistema existente, (ii) comparar diferentes procesos de cálculo para decidir cómo implementar un sistema o (iii) comparar el desempeño de diferentes algoritmos de evaluación. Además, la herramienta permite visualizar en forma gráfica el sistema de integración, los diferentes procesos de cálculo y los factores de calidad lo cual permite una mejor comprensión de los resultados por parte del usuario.
El resto del artículo se organiza de la siguiente manera: En la sección 2 se presenta el enfoque global y se motiva el estudio de la calidad en un SID por medio de un ejemplo. En la sección 3 se presenta un marco de trabajo para representar los procesos y propiedades del SID y modelar el cálculo de la calidad de los datos producidos por dichos procesos. En la sección 4 se describe la herramienta para evaluación de la calidad y en la sección 5 se discute su utilización. Finalmente, en la sección 6 se presentan las conclusiones y perspectivas.

## 2. Enfoque Global

Nuestro objetivo es evaluar la calidad de la información devuelta por un sistema de integración de datos. Concretamente, se propone una herramienta que permite:

- Seleccionar diferentes factores de calidad.
- Facilitar la implementación de algoritmos especializados en dichos factores.
- Modelar diferentes procesos de cálculo y sus propiedades.
- Evaluar los niveles de calidad de los datos devueltos por un proceso de cálculo.
- Presentar al usuario los resultados de dichas evaluaciones.

De esta forma, se da soporte para: (i) el *diagnóstico* de un sistema existente, mostrando los valores de calidad que un proceso de cálculo puede proveer y dando a los usuarios un valor agregado sobre la información devuelta por el sistema, (ii) el *diseño* de un nuevo sistema, evaluando y comparando la calidad de diferentes procesos de cálculo y comparándola con las preferencias de los usuarios, con el fin de seleccionar el proceso de cálculo más apropiado para implementar, y (iii) la *reingeniería* de un sistema, estudiando procesos de cálculo que optimicen a un proceso existente para mejorar la calidad de los resultados. En un contexto de investigación, se da soporte también para el testing y la ejecución de diferentes algoritmos de evaluación de calidad y la comparación del desempeño de los mismos y al testing de técnicas de optimización sobre los procesos de cálculo.
A continuación se describe el proceso de evaluación de la calidad y se presenta un ejemplo que ilustra nuestro enfoque.

### 2.1. *Proceso de Evaluación de la Calidad*

En el proceso de evaluación de la calidad se distinguen dos grandes fases: la personalización de los métodos de evaluación y su utilización para evaluar la calidad del sistema.
En la primera fase se seleccionan los factores de calidad que se quieren evaluar y se implementan

algoritmos de evaluación que calculen dichos factores. La herramienta provee una interfaz y una serie de primitivas para facilitar la implementación de los mismos.
En la segunda fase, se utilizan dichos algoritmos para evaluar y comparar la calidad de diferentes procesos de cálculo. El primer paso es crear una sesión, en la cual se definen las fuentes de datos a las que se accede y las clases de consultas de usuarios que obtienen datos del sistema. A continuación se agregan a la sesión los algoritmos que se quiere utilizar para evaluar la calidad, indicando qué implementación de los mismos utilizar, y se cargan de los procesos de cálculo. La herramienta brinda una representación gráfica de dichos procesos. Una vez configurada la sesión con información sobre las fuentes, las clases de consultas, los algoritmos de evaluación (y por lo tanto los factores de calidad) y los procesos de cálculo, se ejecutan los algoritmos. La herramienta permite ejecutar en paralelo varios algoritmos sobre un conjunto de procesos de cálculo. Una vez finalizada la ejecución se pueden visualizar en forma gráfica los resultados. Por último, la herramienta permite generar un informe para una primera comparación de los resultados, mostrando los valores de calidad alcanzados por cada proceso de cálculo en formato de matriz de doble entrada.
La Figura 1 muestra la utilización de la herramienta en el proceso de evaluación de la calidad.

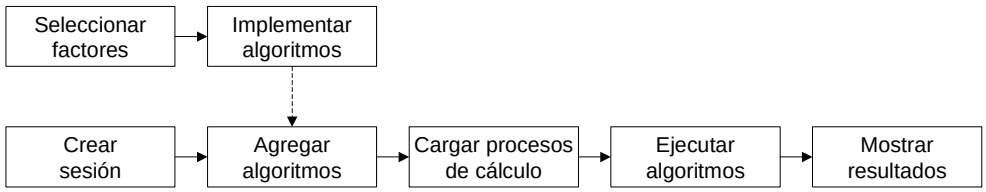


**Figura 1 - Pasos del proceso de evaluación de la calidad**

## 2.2. *Ejemplo Ilustrativo*

Consideremos un sistema de consultas sobre pasajes aéreos partiendo o llegando a la ciudad de Montevideo. Los usuarios están interesados en tres clases de consultas: ($Q_1$) obtención de precios de pasajes por destino y por aerolínea; ($Q_2$) listado de diferentes vuelos para llegar a un destino dentro de un periodo de tiempo; y ($Q_3$) obtención del mínimo precio de pasaje para llegar a un destino.
Para obtener la información se cuenta con tres posibles fuentes de datos: ($R_1$) con información de vuelos internacionales de las principales aerolíneas (pero no contiene información de aerolíneas pequeñas que realizan vuelos cortos); ($R_2$) con datos de una agencia de viajes local sobre aerolíneas que salen y llegan a Montevideo y las posibles combinaciones; y ($R_3$) con información sobre vuelos de Pluna (aerolínea local), actualizada en tiempo real.
Las consultas de usuarios pueden ser resueltas de diferentes formas, siguiendo diferentes procesos de cálculo. Para resolver las consultas de precios de pasajes se puede extraer datos de cualquiera de las fuentes o combinar datos de varias de ellas (promedio de precios, mínimo, máximo). Sucede lo mismo con las consultas de obtención de los vuelos hacia un destino, el resultado puede ser la unión de los vuelos encontrados en las fuentes, la intersección o sólo los que aparecen en alguna fuente en particular. Por lo tanto, se obtienen diferentes resultados dependiendo de la forma de resolver la consulta y dichos resultados tendrán diferentes valores de calidad. Si un vuelo de Pluna cambia de horario sería actualizado de inmediato en la fuente $R_3$. Las consultas resueltas a partir de $R_3$ devolverán datos actualizados, pero si se utilizan las otras fuentes los resultados podrían no estar al día. Sin embargo, los resultados sobre vuelos devueltos por $R_3$ pueden no ser completos, ya que $R_3$ no tiene información del resto de las aerolíneas.
Los valores de calidad más relevantes (o a los que se pretenda dar más peso) pueden ser muy diferentes dependiendo de los usuarios y sus aplicaciones. Por ejemplo, un usuario que está planeando un viaje y busca ideas de destinos (perfil tipo navegador) no se interesa tanto por la frescura o la precisión de los horarios de los vuelos, sino que el tiempo de respuesta es más

relevante para él. Por el contrario, para un agente de viajes que está reservando o vendiendo un boleto la frescura de la información sobre las plazas disponibles, promociones y precios es fundamental y puede permitirse esperar más por dichos datos.
En el resto del documento se utilizará este ejemplo para discutir como se evalúan los factores de calidad a partir de los procesos de cálculo. En particular se mostrará como ejemplo el cálculo del factor "*frescura de los datos*".

## 3. Modelando el Sistema de Integración de Datos

Para modelar los algoritmos de evaluación de la calidad es necesario estudiar y modelar las características del sistema, incluyendo propiedades de los datos fuentes utilizados como materia prima y las propiedades de los procesos de cálculo. Además es necesario tener en cuenta las preferencias de los usuarios, tanto en los factores de calidad que son más relevantes para sus aplicaciones como en los valores de calidad esperados. Una herramienta de evaluación de la calidad debe facilitar la parametrización de dichas preferencias.
En esta sección se presenta un marco de trabajo para la evaluación de la calidad de los datos producidos por un SID. El marco de trabajo modela los procesos de producción y las propiedades del SID que son necesarias para el cálculo de la calidad.

### 3.1. *Modelo de DAG*

Un sistema de integración de datos es un sistema de información que integra datos de diversas fuentes y provee a los usuarios de un acceso uniforme a dichos datos por medio de un modelo global. Las consultas de los usuarios se expresan en términos del modelo global. Algunos ejemplos de SID son los *Sistemas de la Mediación*, los cuales extraen e integran información de varias fuentes de datos para realizar consultas, los *Sistemas de Data Warehousing*, que extraen, transforman y resumen datos de varias fuentes (posiblemente heterogéneas) y la dejan disponible para análisis estratégico y toma de decisiones, las *Federaciones de Bases de Datos*, donde la autonomía de las fuentes es una característica clave y los *Portales Web* que proporcionan acceso a información temática, adquirida y sintetizada de fuentes Web, generalmente utilizando técnicas de Caching.
Un SID puede verse como un flujo de trabajo (workflow) compuesto de actividades que representan las diversas tareas de extracción, transformación y retorno de datos a los usuarios. Cada actividad toma como entrada datos de las fuentes u otras actividades y produce datos que pueden ser utilizados como entrada para otras actividades. De esta forma, los datos siguen un camino desde las fuentes hasta los usuarios, siendo transformados y procesados según la lógica del sistema. Los datos producidos por una actividad pueden ser inmediatamente consumidos por otra actividad o pueden materializarse para ser consultados más tarde. Observe que esta noción de actividad puede representar procesos de diversa complejidad; desde simples operaciones SQL hasta complejos procedimientos de transformación que pueden ejecutarse autónomamente.
La Figura 2 muestra la representación de un SID como un flujo de trabajo. En la parte inferior se muestran las fuentes ($R_i$). En la parte central se muestran las diversas actividades ($A_i$) que ejecutan a partir de los datos fuentes. Las flechas indican que el nodo de la salida utiliza los datos producidos por el nodo de la entrada. Las actividades que toman los datos directamente de relaciones fuente son los extractores de datos (wrappers). Las otras actividades toman datos de entrada directa o indirectamente de los wrappers. En la parte superior se muestran las clases de consultas de usuario ($Q_i$) que representan las familias de consultas que pueden responderse usando los datos producidos por las actividades.
Formalmente, representamos el flujo de trabajo del SID por medio de un grafo acíclico dirigido (DAG) que describe las actividades implicadas, sus entradas y salidas. El DAG muestra el flujo de datos desde las fuentes hasta las clases de consultas pasando por las diversas actividades.

**Definición 1.** Un *DAG de calculo* (CDAG) G, es un grafo acíclico dirigido definido de la siguiente manera: Los nodos de G son de tres tipos: *nodos fuente* (sin ninguna artista entrante) que representan las fuentes, *nodos destino* (sin ninguna arista saliente) que representan las clases de consultas, y *nodos de actividades* (con aristas entrantes y salientes) que representan las diferentes actividades que calculan el conjunto de nodos destino a partir de los nodos fuente. Las aristas de G indican que un nodo es calculado a partir de otro (los datos transitan en el sentido de la flecha). □

**Ejemplo 1.** En la Figura 2 se muestra un CDAG que representa un proceso de cálculo para resolver las consultas del ejemplo de la sección 2.2. Las actividades $A_1$, $A_2$ y $A_3$ extraen la información de las fuentes de datos. La actividad $A_4$ filtra los datos de las compañías que pueden ser operadas desde Uruguay. $A_5$ integra información de $A_2$ y $A_3$. $A_6$ realiza la unión de la información de $A_4$ y $A_5$. Por último, $A_7$ computa una función agregada sobre los datos de $A_6$ obteniendo el mínimo precio por cada destino. □

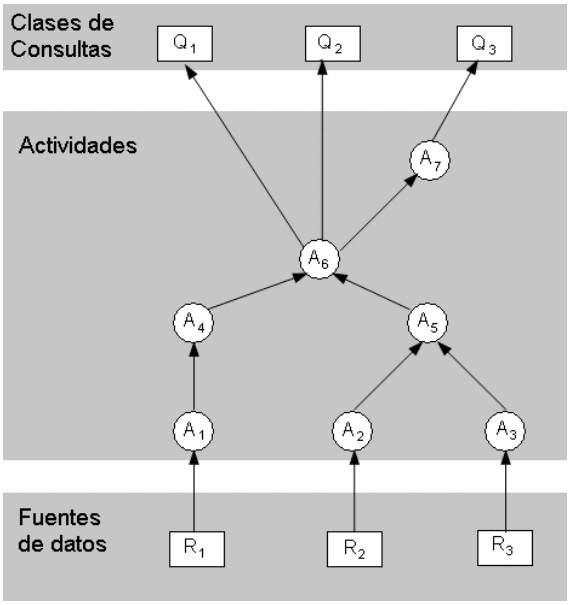


**Figura 2 - DAG de cálculo del ejemplo**

## 3.2. *Propiedades Asociadas al Cálculo de la Calidad*

En esta sección describimos las características y propiedades del SID necesarias para expresar la calidad del sistema. Para realizar la evaluación se necesita, en primer lugar, identificar qué factores de calidad se van a evaluar. La elección de los factores de calidad más apropiados para un determinado SID depende de los requerimientos del usuario y sus aplicaciones. Varios trabajos estudian los factores de calidad que son más relevantes para diversos tipos de sistemas, por ejemplo [7]. La selección de los factores de calidad implica la selección de las métricas apropiadas y la implementación de los algoritmos de la evaluación para esos factores.

Los algoritmos de evaluación necesitan como entrada cierta información que describe las características del sistema, por ejemplo, el tiempo que una actividad necesita para ejecutar o un descriptor que indica si una actividad materializa datos o no. Las propiedades pueden ser de dos tipos: (i) *descripciones*, indicando una cierta característica del sistema (costos, retardos, políticas, estrategias, restricciones, etc.), o (ii) *medidas*, indicando un valor que corresponde a un factor de calidad, que puede ser un *valor real* adquirido de una fuente, un *valor calculado* por un algoritmo de evaluación o un *valor esperado* por un usuario. La elección de las propiedades adecuadas depende de los factores de calidad y del tipo de sistema. A continuación se describe como ejemplo el factor de calidad *frescura* y tres propiedades necesarias para su evaluación:

**Frescura**: La *frescura de los datos* (data freshness) representa la "edad" de los datos, indicando qué tan "viejos" son dichos datos y si son "apropiados" para un determinado requerimiento [7]. Se mide como el tiempo transcurrido entre el momento en que los datos fueron producidos y el momento en que dichos datos son retornados al usuario en respuesta a una consulta [4]. Otras definiciones y métricas son estudiadas en [1].

En la medición de la frescura de los resultados son importantes: la frescura de los datos en las fuentes (frescura real), los tiempos de ejecución de las actividades (costo de procesamiento) y los tiempos de espera entre la ejecución de las actividades (demora de sincronización). También es importante conocer las expectativas de frescura de los usuarios (frescura esperada) para comparar con las mediciones. A continuación describimos dichas propiedades:

**Costo de procesamiento**: El costo de procesamiento de una actividad es el lapso de tiempo, en el peor caso, que la actividad necesita para leer los datos de entrada, ejecutar y producir el resultado. Hay varios retardos relacionados con el costo de procesamiento de la actividad. Para los wrappers incluye el tiempo necesario para comunicarse con la fuente (enviar de la consulta y recibir la respuesta), realizar la extracción y materializar los resultados (en caso necesario). Para las otras actividades incluye el tiempo necesario para leer los datos de entrada, procesar los datos y materializar los resultados (en caso necesario).

**Demora de sincronización**: Cuando dos actividades consecutivas en un camino del CDAG ejecutan con diferentes frecuencias (por ejemplo una ejecuta una vez al día y otra una vez por semana), los datos producidos por la primera se deben materializar para ser consultados más adelante por la segunda. En tal caso, hay una demora de sincronización. Dicha demora es la cantidad de tiempo transcurrido entre el final de la ejecución de una actividad y el comienzo de la otra. Las demoras de sincronización son muy importantes en la evaluación de la frescura porque introducen tiempos de espera suplementarios y por lo tanto disminuyen la frescura de los datos.

**Frescura real**: La frescura real es la medida de la frescura de datos en una fuente. La misma puede ser proporcionada por la fuente, puede ser estimada o acotada por el sistema o puede ser estimada por usuarios expertos.

**Frescura esperada**: La frescura esperada es el valor de frescura máximo esperado o tolerado por los usuarios. La misma puede ser proporcionada por los usuarios o estimada a través de su comportamiento (logs, históricos de requerimientos, etc.).

Una propiedad se relaciona con los ciertos nodos o aristas del CDAG. Por ejemplo, podemos asociar el costo de procesamiento a los nodos de actividades, la frescura real a los nodos fuente y la demora de sincronización a las aristas.

**Definición 2.** Un *DAG de calculo etiquetado* (LDAG), es un DAG de calculo cuyos nodos y aristas tienen propiedades asociadas. □
La formalización de un LDAG y las propiedades puede consultarse en [5].

**Ejemplo 2.** En la Figura 3 se muestra el CDAG del ejemplo anterior, etiquetado con algunas propiedades: frescura real (nodos fuente), frescura esperada (nodos destino), costo de procesamiento (nodos de actividades y nodos destino) y la demora de sincronización (aristas). Los nodos de actividades que ejecutan periódicamente se etiquetaron también con su período de ejecución para facilitar la lectura del ejemplo. Todos los valores están expresados en minutos.
El costo de procesamiento de los nodos $A_1$, $A_2$ y $A_3$ corresponde a la extracción de datos desde sus respectivas fuentes. El costo de procesamiento de $A_2$ es mayor porque la velocidad en la conexión con la fuente es menor. Los costos de procesamiento de las actividades $A_4$, $A_5$, $A_6$ y $A_7$ corresponden al tiempo necesario para leer los datos de entrada, realizar las operaciones y materializar los resultados (esto último si corresponde).

Las actividades $A_1$ y $A_4$ se ejecutan a demanda de las actividades que las suceden. Estas actividades no materializan sus resultados si no que son leídos directamente por la actividad sucesora. Por lo tanto, no hay demora de sincronización entre ellas. En cambio, las otras actividades ejecutan periódicamente materializando sus resultados. Como algunas de ellas ejecutan con diferentes períodos, existen demoras de sincronización. Por ejemplo, las actividades $A_2$ y $A_5$ pueden sincronizarse para que el tiempo que transcurre desde que $A_2$ materializa sus resultados hasta que $A_5$ los lee sea de 24 minutos en el peor caso. En cambio las actividades $A_6$ y $A_7$ pueden sincronizarse para que no haya demora. En [6] se presenta un análisis de las demoras de sincronización y las formas de estimarlas. □

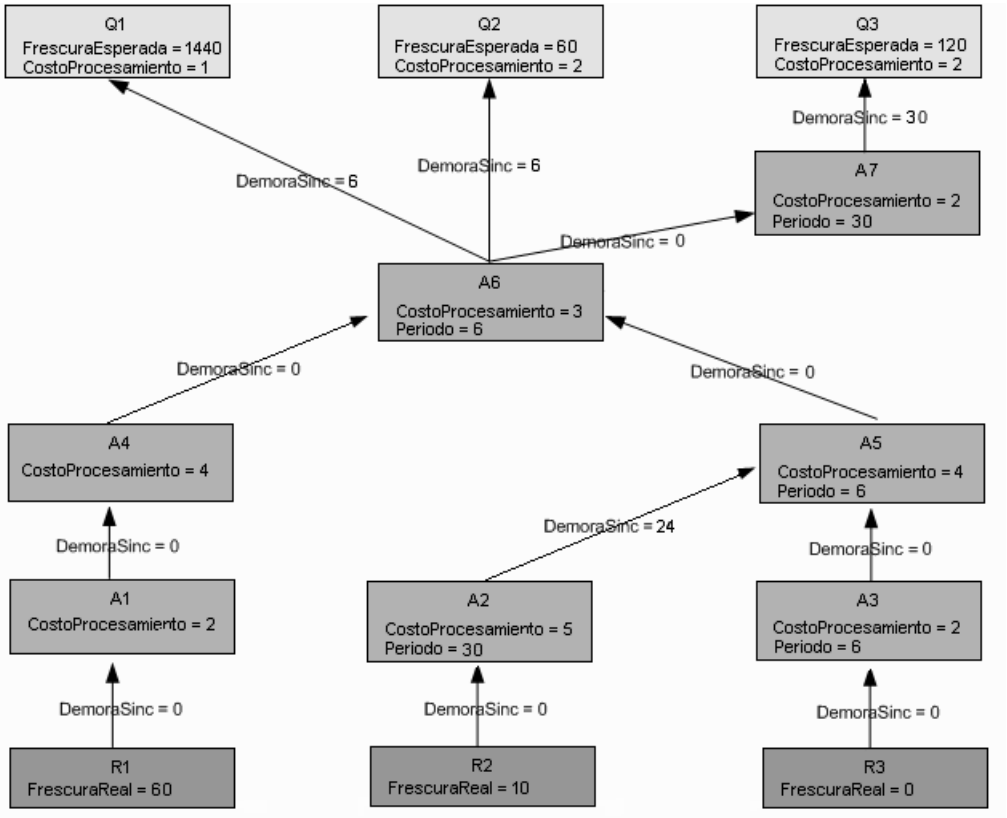


**Figura 3 – DAG de cálculo etiquetado con algunas propiedades**

## 3.3. *Algoritmos de Evaluación*

El cálculo de la calidad es realizado por algoritmos de evaluación. Cada algoritmo toma como entrada un LDAG y retorna un nuevo LDAG con una nueva propiedad asociada, correspondiente al factor de calidad calculado por el algoritmo.

Para la ejecución de un algoritmo el LDAG debe tener asociadas en los nodos y aristas las propiedades que el algoritmo espera. Por ejemplo, para el cálculo de la frescura las actividades deben tener asociada la propiedad *costo de procesamiento*.

**Algoritmo de evaluación de la frescura**

El algoritmo de evaluación de la frescura calcula la frescura que tendrán los datos devueltos a los usuarios por un proceso de cálculo. Para ello toma un LDAG cuyos nodos y aristas tienen asociada la siguiente información:

- Nodos fuentes con la propiedad “frescura real”.
- Nodos de actividades con la propiedad “costo de procesamiento”.
- Nodos destino con la propiedad “costo de procesamiento”.
- Aristas con la propiedad “demora de sincronización”.

Según [5] la frescura de los datos producidos por un nodo depende de la frescura de los datos de entrada (la frescura de los datos producidos por el nodo precedente más la demora de sincronización entre ambos nodos) más el tiempo que el nodo necesita para ejecutar (costo de procesamiento). Para calcular la frescura del nodo sumamos dichos valores. Cuando el nodo lee datos de diferentes nodos de entrada, los valores de frescura de las entradas deben ser combinados. Como estamos interesados en devolver los valores de frescura en el peor caso tomamos el máximo.

**Definición 3.** La frescura de un nodo de actividad o destino en un LDAG G es la máxima suma de la frescura de un nodo predecesor, más la demora de sincronización entre ambos nodos más el costo de procesamiento del nodo. La frescura de un nodo fuente es su frescura real.

- Para un nodo fuente A:

  Frescura(A) = FrescuraReal(A)

- Para un nodo de actividad o destino A:

  Frescura(A)= max{Frescura(B)+DemoraSinc(B,A,G) /B ∈ predecesores(A,G)}+Costo(A,G)

Las funciones *FrescuraReal*, *DemoraSinc* y *Costo* devuelven las propiedades respectivas. La función *predecesores* devuelve el conjunto de predecesores de un nodo en el DAG. □

**Ejemplo 3.** En la Figura 4 se muestra el resultado de la ejecución del algoritmo de frescura sobre el LDAG de la Figura 3, etiquetando los nodos con una propiedad adicional: la *frescura*, calculada por el algoritmo.

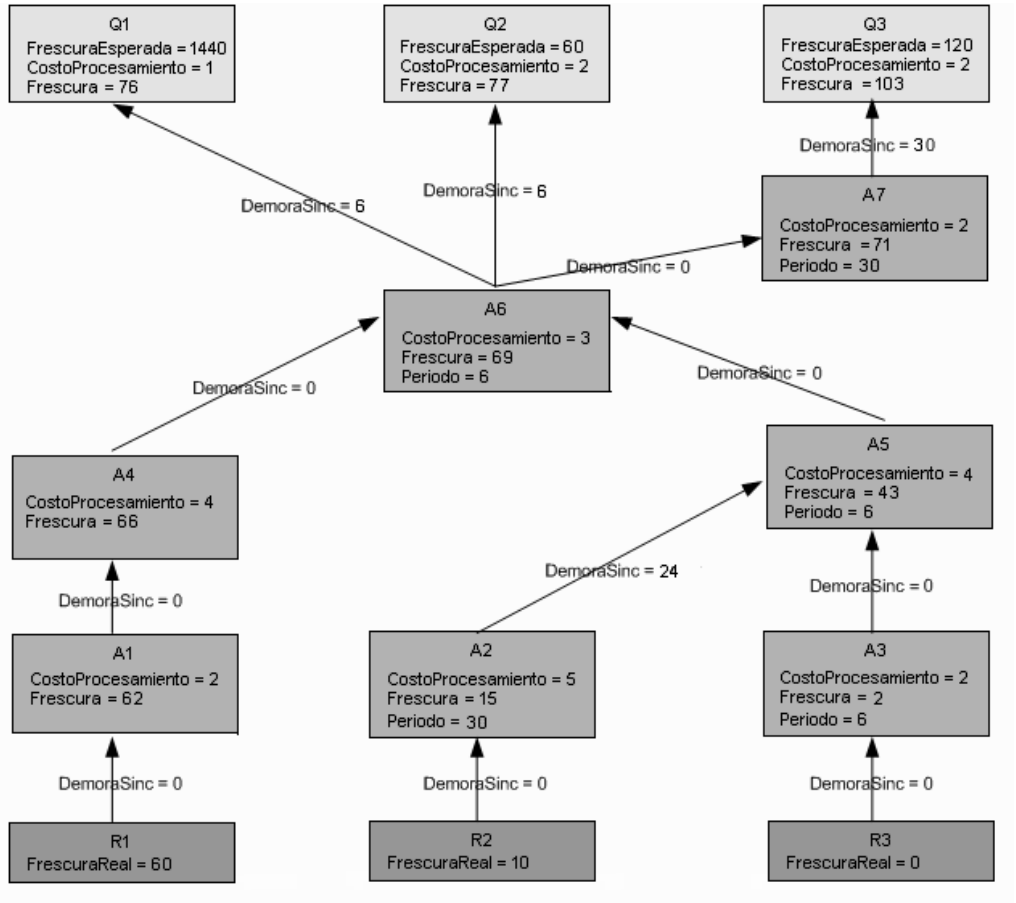


**Figura 4 - LDAG resultado de ejecutar el algoritmo de evaluación**

Por ejemplo, la frescura en el nodo $A_5$ se calcula como la suma del costo de $A_5$ y el mínimo entre: la frescura de $A_2$ más la demora de la sincronización entre ambos y la frescura de $A_3$ más la demora de sincronización. Por lo tanto, la frescura de $A_5$ se calcula como min{15+24, 2+0}+4, con lo que se

obtiene que la frescura de $A_5$ es 43. Para la actividad $A_7$ la frescura se calcula como la frescura de $A_6$ más la demora de sincronización más el costo de procesamiento de la actividad $A_7$. La frescura es entonces 69 + 0 + 2 = 71. □

# 4. Implementación de la Herramienta

En esta sección se describe la implementación de la herramienta propuesta para evaluar y comparar la calidad de los datos devueltos por un proceso de cálculo. Se presenta primero la arquitectura de la herramienta y luego se describe la implementación de las diferentes funcionalidades del sistema.
La herramienta se implementó en Java (JDK 1.4) accediendo a metadatos en formato XML a través de Castor [2].

## 4.1. *Arquitectura*

La herramienta posee una arquitectura en capas y la comunicación entre ellas se realiza por medio de interfaces de forma de estandarizar el acceso a los servicios de cada capa. El manejo de la representación gráfica sigue el patrón *Model-View-Controller*, el cual está implementado en 3 capas distintas (view, logic, model). La Figura 5 muestra un diagrama de las capas de la arquitectura y la interacción entre ellas, así como los paquetes más relevantes que las componen.

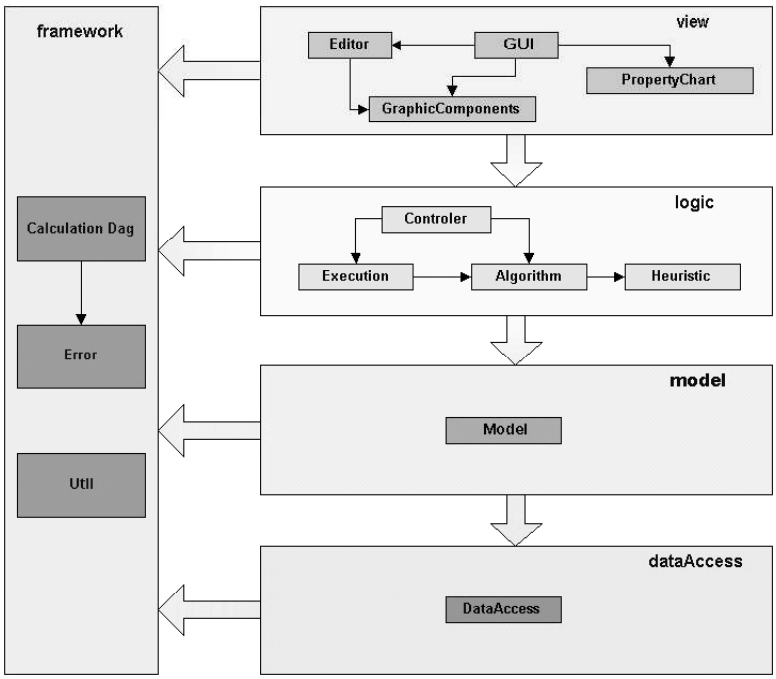


**Figura 5 – Diagrama de la arquitectura**

A continuación se presenta una breve descripción de cada capa:
La capa *View* es la que contiene la representación gráfica que se presenta al usuario. Los principales paquetes que la componen son "Editor" el cual permite la edición y creación de procesos de cálculo, "GUI" que posee la interfaz de ventanas, "GraphicComponents" el cual brinda los componentes para la representación de los procesos de cálculo, y "PropertyChart" el cual permite visualizar la tabla comparativa de resultados.
*Model* es la capa que contiene la información del modelo de datos que se muestra al usuario. El paquete principal es "Model" el cual mantiene la información de los procesos de cálculo, los algoritmos y el resultados de las ejecuciones
La capa *Logic* completa el MVC junto con las capas *View* y *Model*. Esta capa contiene la lógica de la aplicación proporcionando los métodos necesarios para ejecutar algoritmos y manejar sus

resultados. Posee varios paquetes importantes, por un lado “Heuristic” contiene implementaciones de diferentes heurísticas. “Algorithm” contiene la interfase que deben implementar los algoritmos y algunos algoritmos de ejemplo ya implementados. “Controller” contiene la lógica de la aplicación haciendo un puente entre el modelo de datos y su representación. Finalmente “Execution” es quien contiene la lógica que permite la ejecución concurrente de los algoritmos.
La arquitectura posee dos capas adicionales: DataAccess y Framework. *DataAccess* brinda las funcionalidades necesarias para la persistencia y recuperación de la información de sesiones, algoritmos y procesos de cálculo.
*Framework* es una capa transversal que brinda servicios básicos a todas las otras capas. Los principales paquetes son “CalculationDag” el cual posee la representación de un CDAG, “Error” quien brinda objetos para el manejo de errores, y “Util” la cual contiene funcionalidades varias.

## 4.2. *Implementación de las funcionalidades*

En la sección 2.1 se describió el proceso de evaluación de la calidad, cuyos pasos se muestran en la Figura 1. En esta sección se describe la implementación de dichas funcionalidades en la herramienta.

**Seleccionar factores de calidad**
Es una tarea subjetiva que depende de los intereses de los usuarios y de las aplicaciones que manejan. Por el momento es realizada en forma manual por los usuarios y administradores del sistema. Entre las posibles soluciones que ayuden a automatizar esta tarea, puede obtenerse información de perfiles de usuarios e históricos de su comportamiento y deducir los factores importantes a partir de dicha información.

**Implementación de algoritmos de evaluación**
La herramienta cuenta con un conjunto de algoritmos de ejemplo para evaluar algunos factores de calidad, pero fue diseñada para permitir la fácil incorporación de nuevos algoritmos. Para ello se definió una interfaz (IAlgorithm) a través de la cual la aplicación se comunica con los algoritmos. La interfaz cuenta únicamente con dos métodos: (i) *execute*, que permite a la herramienta invocar la ejecución de un algoritmo sobre un proceso de cálculo, y *getQualityProperties*, que permite consultar cuáles son los factores de calidad que el algoritmo calcula. Para implementar un nuevo algoritmo de evaluación (o para adaptar uno ya existente) simplemente se debe implementar dicha interfaz. El acceso a los procesos de cálculo es a través de una única interfaz. La misma incluye métodos para seleccionar los distintos componentes (nodos, aristas, propiedades, etc.) y navegar por la estructura del grafo (por ejemplo obtener los sucesores de un nodo). Además incluye métodos para agregar nuevos nodos/aristas y nuevas propiedades y asociar valores de propiedades a nodos/aristas.
También puede definirse un archivo de parametrización para el algoritmo que contenga información de configuración, por ejemplo el nombre de las propiedades de nodos y aristas a las que necesita acceder, el nombre de los factores de calidad que calcula o cualquier otra información de interés para el algoritmo.

**Creación de una sesión**
Una sesión define las fuentes de datos a las que se accede y las clases de consultas de usuarios que interrogan el sistema. La herramienta accede a dichos metadatos a través de archivos de parametrización en XML. Por esto se definieron esquemas XML (XSD) y se generaron las clases para la carga y descarga de información mediante Castor [2]. Esto permite utilizar diferentes archivos de entrada, con diferentes formatos, dejando la posibilidad a la aplicación (a nivel de la vista) de elegir cuál usar en una instancia dada del programa.

**Incorporación de un algoritmo a una sesión**
La herramienta posee una interfaz para agregar dinámicamente nuevos algoritmos. Básicamente debe indicarse el nombre del algoritmo, la clase que lo implementa, una descripción detallada y la URL del archivo de parametrización. La herramienta controla que la clase esté bien definida (cumpla con la interfaz).

**Carga de un proceso de cálculo**
Al igual que los metadatos que describen las fuentes y las clases de consulta de los usuarios, los metadatos que describen los LDAGs (que representan los procesos de cálculo y sus propiedades asociadas) se almacenan en archivos XML. La incorporación de un LDAG a una sesión puede realizarse dinámicamente, simplemente indicando la URL del archivo. También puede editarse un LDAG utilizando el editor gráfico de la herramienta, y pueden agregarse o modificarse propiedades a los LDAGs ya cargados. La herramienta brinda un visualizador gráfico de los LDAGs y sus propiedades. Las Figuras 3 y 4 son capturas de pantallas de la herramienta.
Los procesos de cálculo se diseñaron de forma de permitir definir tipos de nodos y aristas, asociar propiedades a los tipos y asignar valores a las propiedades. Por lo tanto, un proceso de cálculo se define como:

- Un identificador.
- Una colección de tipos de nodos. Para cada tipo de nodo se define qué propiedades tiene asociadas y qué tipo de dato tiene cada propiedad.
- Una colección de tipos de aristas. Análogamente, para cada tipo de arista se definen las propiedades asociadas y sus tipos de datos.
- Una colección de nodos. Cada nodo corresponde a un tipo de nodo y tiene asociados valores para las propiedades de su tipo de nodo.
- Una colección de aristas. Análogamente, cada arista corresponde a un tipo de arista y tiene asociados valores para las propiedades de su tipo de aristas.

Este diseño permite representar cualquier LDAG que cumpla con la Definición 1.

**Ejecución de un algoritmo**
Una vez cargada la información de una sesión: fuentes, clases de consultas, algoritmos (y por lo tanto factores de calidad) y procesos de cálculo, la herramienta permite la ejecución de un algoritmo de evaluación sobre un proceso de cálculo, o la ejecución en paralelo de varios algoritmos sobre un conjunto de procesos de cálculo. La aplicación ejecuta los algoritmos invocando al método *execute*.
Cada ejecución de un algoritmo se realiza en una hebra separada (thread). Cuando la ejecución finaliza dispara un evento, de forma que todos los componentes interesados sean informados que la ejecución finalizó y en qué estado. El manejo de eventos se hace a través de la interfaz IexecutionListener y la clase ExecutionObserver.

**Visualización de los resultados**
Como se dijo anteriormente, la herramienta posee un visualizador de LDAGs, por lo tanto, luego de ejecutar un algoritmo de evaluación se puede visualizar el LDAG resultado el cual contiene una nueva propiedad con los valores calculados para el factor de calidad correspondiente.
Además, se puede generar un informe comparativo de los valores de los factores de calidad calculados para los diferentes procesos de cálculo. El informe se muestra en forma de matriz de doble entrada donde las filas representan los procesos de cálculo y las columnas los algoritmos que evaluaron los factores de calidad. En las celdas aparece el valor calculado para el proceso de cálculo y el factor de calidad.

## 5. Utilización de la Herramienta

El sistema de integración debería proveer a los usuarios con información que satisfaga sus requerimientos de calidad. Los valores de calidad calculados por los algoritmos de evaluación pueden compararse con los valores de calidad esperados por los usuarios de manera de determinar si el sistema permite satisfacer sus expectativas. Entonces, un primer uso del sistema es como herramienta de *diagnóstico*, para informar de la calidad de un sistema existente.
Si los niveles de calidad no son suficientes para los usuarios se debería mejorar la implementación del sistema para asegurar la calidad o negociar con los proveedores de datos (fuentes) o los usuarios para relajar restricciones. La evaluación de la calidad también puede usarse para identificar las actividades que deterioran la calidad, por ejemplo, las actividades más costosas o las demoras de sincronización más grandes que deterioran la frescura de los datos. De esta forma, la evaluación de la calidad es de utilidad para la *optimización* o *re-ingeniería* del sistema.
Por otro lado, si el sistema aún no ha sido implementado, se puede estudiar la calidad de diferentes procesos de cálculo, de manera de compararlos e implementar el más beneficioso. En este caso, la evaluación se usa durante el *diseño* del sistema.
Otro uso es el de *testing*, es en un contexto de investigación, donde se quiere comparar el desempeño y la relevancia de diferentes algoritmos de evaluación y comparar los resultados obtenidos por cada uno. Por ejemplo, en [5] se discute la utilización del enfoque para estudiar la evaluación de la frescura de los datos.

## 6. Conclusiones

En este artículo se presentó una herramienta para evaluar la calidad en un sistema de integración de datos. La herramienta permite modelar diferentes elementos del sistema de integración que tienen impacto en la evaluación de la calidad, como son: las fuentes de datos, las clases de consultas de usuarios, los procesos que extraen, integran y devuelven información a los usuarios, las propiedades de éstos, los factores de calidad y los algoritmos de evaluación. La herramienta fue concebida para permitir la fácil incorporación de nuevos factores de calidad, propiedades y algoritmos de evaluación, brindando un contexto flexible y generalizable para el testing de algoritmos y técnicas.
Actualmente se la está utilizando para experimentar con la evaluación de la frescura de los datos, adaptando los algoritmos de evaluación a diferentes escenarios. Como trabajo futuro, se prevé aplicar el encare al análisis de otros factores de calidad.

## Referencias


[1] Bouzeghoub, M.; Peralta, V.: “A Framework for Analysis of Data Freshness”. En anales del International Workshop on Information Quality in Information Systems (IQIS’2004), Francia, 2004.

[2] Exolab Group: “The Castor Project”. URL: http://castor.exolab.org. Versión 0.9.5.3 disponible en ftp://ftp.exolab.org/pub/castor/castor_0.9.5.3/castor-0.9.5.3.jar.

[3] Fajardo, F.; Crispino I.: “Implementación de una Plataforma para Análisis de Calidad de Datos”. Informe de proyecto de grado, Facultad de Ingeniería, Universidad de la República, Uruguay, 2004.

[4] Naumann, F.; Leser, U.: “Quality-driven Integration of Heterogeneous Information Systems”. En anales de 25th International Conference on Very Large Databases (VLDB’99), Scotland, 1999.

[5] Peralta, V.; Ruggia, R.; Kedad, Z.; Bouzeghoub, M.: “A Framework for Data Quality Evaluation in Data Integration Systems”. Aceptado para el 19º Simposio Brasilero de Bases de Datos (SBBD’2004), Brasil, 2004.

[6] Peralta, V.: “Evaluating and Enforcing Data Freshness in Data Information Systems”. Technical Report. InCo. Universidad de la República, Uruguay.

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