Enfoque Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones

Mario José Diván

Facultad de Ciencias Económicas y Jurídicas – Facultad de Ingeniería
Universidad Nacional de La Pampa

15/07/2011

Tesis presentada para obtener el grado de Doctor en Ciencias Informáticas

Facultad de Informática - Universidad Nacional de La Plata

Director	**Dr. Luis Olsina** Grupo de Investigación en Ingeniería del Software y Web (GIDIS_WEB) Facultad de Ingeniería Universidad Nacional de La Pampa
Co - Director	**Dra. Silvia Gordillo** Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA) Facultad de Informática Universidad Nacional de La Plata

Resumen

Cuando se trata de tomar decisiones a un nivel ingenieril, medir no es una posibilidad sino una necesidad; representa una práctica sistemática y disciplinada por la cual se puede cuantificar el estado de un ente. Si hay un aspecto que se debe tener en claro en medición, es que para comparar mediciones diferentes las mismas deben ser consistentes entre sí, esto es, deben poseer la misma escala y tipo de escala además de obtenerse bajo métodos de medición y/o reglas de cálculos equivalentes. Los marcos de medición y evaluación representan un esfuerzo, desde la óptica de cada estrategia, por formalizar el modo de definir las métricas, sus objetivos, entre otros aspectos asociados, a los efectos de garantizar la repetitividad y consistencia en el proceso de medición que sustentan.

Existen aplicaciones capaces de procesar flujos de mediciones en línea, pero el inconveniente principal con el que se enfrentan, es que no contienen información con respecto al significado del dato que están procesando. Para este tipo de aplicaciones, la medición es un dato, es decir, una forma de representar un hecho captado, careciendo de información sobre el concepto al que se asocian o bien, el contexto en el cual es obtenida dicha medición.

Los dispositivos de medición, están en general desarrollados para captar una medida mediante un método dado, y en la mayoría de los casos, la forma de obtener dicha medida para su posterior procesamiento en otros entornos (ordenadores de escritorio, móviles, etc.), está en función de servicios o accesorios provistos por el fabricante.

Suponiendo que la totalidad de las mediciones, provenientes de diferentes dispositivos, pudieran ser incorporadas en un mismo canal de transmisión, pocos son los entornos de procesamiento de flujos de datos que incorporan comportamiento predictivo. En aquellos que se incorpora comportamiento predictivo, ninguno de los analizados se sustenta en una base conceptual, que permita contrastar una medida contra la definición formal de su métrica. Esto último, incorpora un serio riesgo de inconsistencia, que afecta directamente al proceso de medición y en consecuencia, a los posteriores análisis que en base a estos datos se realicen.

Nuestra Estrategia de Procesamiento de Flujos de Datos centrado en Metadatos de Mediciones (EIPFDcMM), se focaliza en permitir la incorporación de fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados C-INCAMI, permitan realizar análisis estadísticos de un modo consistente a los efectos de implementar un comportamiento detectivo y a su vez, permitan incorporar información contextual a las mediciones, para enriquecer la función de clasificación con el objeto de implementar el comportamiento predictivo. Tanto la implementación del comportamiento detectivo como del predictivo tendrán asociados mecanismos de alarma, que permitirán proceder a la notificación ante la eventual identificación de una zona de riesgo. De este modo, se pretende garantizar la repetitividad y consistencia en el proceso de medición que sustentan

Agradecimientos

Esta tesis hoy culmina una etapa. Etapa, en la que han participado numerosas personas, que con su granito de arena, bendijeron su conclusión. Mi más profundo agradecimiento:

- A mi esposa Laura, mis hijos Ignacio y Santiago, por el apoyo y amor incondicional que día a día me brindan. Ellos son mis luceros cuando todo parece perdido y/o de difícil realización.
- A mi madre, Anita Koller, La Anita o simplemente Chuschitas (como le solía decir), quien nos ha dejado un ejemplo de amor, seriedad, constancia y honestidad que rigen hoy nuestra conducta y por la cual, le estaré eternamente agradecido.
- A mi viejo, por su ejemplo de perseverancia.
- Al Dr. Luis Olsina, por su minuciosa y detallada dirección, por el empeño aplicado, su cordialidad, disposición y amabilidad manifiesta, no solo en el plano académico, sino también en el plano humano. Sus enseñanzas son invalorables.
- A la Dra. Silvia Gordillo por su disposición y soporte en la co-dirección de esta tesis.
- Al Cr. Roberto Vassia, por creer en este proyecto, por brindar su apoyo incondicional, por compartir su visión de investigación-sociedad, y no claudicar en ningún momento sobre la importancia de la formación de investigadores en el ámbito de nuestra Facultad de Ciencias Económicas y Jurídicas (Universidad Nacional de La Pampa -UNLPAM).
- Al Ing. Carlos D'amico, por brindar su colaboración incondicional al proyecto desde la Facultad de Ingeniería (UNLPAM)
- A mis compañeros del grupo de investigación GIDIS_WEB de la Facultad de Ingeniería, por su permanente colaboración y contención
- A todo el personal no docente de la Facultad de Ciencias Económicas y Jurídicas como así también al de la Facultad de Ingeniería. Gracias a su predisposición, cordialidad, esfuerzo y su apoyo diario, es que podemos desarrollar nuestra labor.
- Al Cr. Rinaldo Caumo, quien no dudo en respaldar el emprendimiento original de este proyecto de investigación a través de las becas cofinanciadas del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).
- A CONICET, por darme la posibilidad de hacer uso de la beca cofinanciada tipo I.
- Al Dr. Guillermo Galcerán, Cr. Hugo Abalo, Cr. Pablo Echeveste, Cr. Jorge Andrade, Cra. Marta Andreani y al Dr. Marcelo Piazza por la cordialidad, colaboración y disposición permanente brindada desde sus funciones, durante todo el proceso de formación.

A todos ellos, quiero decirles nuevamente ***¡¡¡muchas gracias!!!*** Porque lo hoy obtenido, es el fruto del esfuerzo de una comunidad que cree fervientemente en la universidad pública y el desarrollo social e inclusivo de la ciencia y la tecnología.

Mario Diván

Santa Rosa (La Pampa), septiembre de 2011

Índice

Lista de Figuras

Lista de Tablas

1 Introducción

En la actualidad existen aplicaciones de software que procesan un conjunto de datos a medida, que se generan en forma continua, con el fin de responder a consultas y/o adecuar su comportamiento en función del propio arribo de datos (Namit, Gehrke, et al., 2008). Este tipo de aplicaciones tienen por finalidad actuar de un modo pro activo y en donde la persistencia no es una prioridad, sino que el aspecto primordial radica en agilizar el procesamiento de los datos ante su arribo, evitando en lo posible el descarte de los mismos por cuestiones de desborde, minimizando el uso de recursos a los efectos de brindar una respuesta aproximada. En este último punto radica el sacrificio fundamental efectuado por este tipo de aplicaciones, es decir que se prefiere disminuir precisión descartando datos ante potenciales desbordes a los efectos de ganar agilidad en la respuesta en línea.

El presente capítulo expondrá en primer lugar las motivaciones y antecedentes que permitieron arribar al dominio del tipo de aplicaciones indicadas en el párrafo anterior. Presentados los antecedentes y motivaciones, se procede a efectuar el planteamiento del problema en concreto, para continuar con los requerimientos y supuestos que rigen al planteo del mismo. A seguir, se plantean las principales contribuciones y finalmente se describe la estructura de la presente tesis.

1.1 Motivación y Antecedentes

Como se mencionó anteriormente, existen aplicaciones de software que procesan los datos en el momento de su arribo primando el principio de agilidad en la respuesta, aunque ello implique un sacrificio en la precisión dado que arrojan resultados aproximados. Este tipo de aplicaciones de procesamiento de flujos de datos continuos, rigen su comportamiento principalmente a partir de los valores actuales de los datos y en donde, la historia presenta una incidencia relativa que si bien aporta, no es esencial para brindar una respuesta. Por ejemplo, las aplicaciones para el monitoreo de signos vitales de pacientes emiten mediciones continuamente, el histórico es importante por cuanto se podría estudiar la evolución de un paciente a partir del mismo, pero ante una situación de urgencia lo importante a los efectos de disparar las alarmas correspondientes, es fundamentalmente la interpretación de los valores actuales arrojados por las mediciones y no su histórico. En forma análoga a las aplicaciones de monitoreo de signos vitales, pueden encontrarse otras aplicaciones tales como las asociadas al comportamiento de los mercados financieros; de centrales nucleares; de tráfico aéreo; entre otras.

Durante el transcurso del 2003 y vinculado a la Maestría en Administración de Negocios que en aquel momento me encontraba desarrollando, se planteó la idea de construir un modelo de centro de distribución celular para acopio de miel en la cooperativa de Doblas (Departamento Atreucó – La Pampa) (Diván M. , 2006). En dicho modelo, uno de los aspectos fundamentales radicaba en el monitoreo continuo de las condiciones ambientales del centro de distribución y de los tambores de miel que contenía, a los efectos de evitar situaciones de deterioro del alimento que pudiesen producir riesgos a la salud derivados de su consumo (Congreso de la República Argentina, 1969). De este modo, el requerimiento de medición fue fundamental como así también la formalización de las métricas. Dado que se trataba de una Maestría en Negocios, en aquella tesis se empleó el enfoque de Kaplan y Norton (Kaplan & Norton, 2000) para la definición e implementación de las métricas que nutrirían los tableros de comandos involucrados en el monitoreo. Una de las debilidades que se pudo apreciar en el cuadro de mando integral de Kaplan y Norton radicaba en la ausencia de ontologías que sustenten formalmente los conceptos intervinientes en el proceso de medición, como así también la no consideración del contexto de medición.

En 2004 se toma contacto con el marco formal de medición y evaluación INCAMI (Information Need, Concept, Attribute, Metric and Indicator) (Olsina, Molina, et al., 2005), a través de la serie de cursos CuPIWeb organizados por la Facultad de Ingeniería de la Universidad Nacional de La Pampa (UNLPAM), el cual permitía formalizar las métricas e indicadores, como sus conceptos asociados careciendo en ese entonces de información contextual relacionada a las mismas. La incorporación de la ontología proveniente de INCAMI, suplía muchas de las dificultades del cuadro de mando integral de Kaplan y Norton que habían surgido durante el desarrollo del modelo de centro de distribución celular para acopio de miel, permitiendo la formalización del proceso de medición y evaluación y por ende su automatización. En 2007, el marco INCAMI incorpora la capacidad de modelar información contextual dentro del proceso de medición y evaluación pasándose a llamar C-INCAMI (Context - Information Need, Concept Model, Attribute,

Metric and Indicador) (Molina & Olsina, 2007), lo que permitió formalizar la situación en la cual el proceso de medición era llevado adelante, facilitando notablemente la posibilidad de automatización del proceso.

La idea de monitorización continua iniciada en el modelo de centro de distribución celular, incorporando la posibilidad de formalizar con información contextual el proceso de medición y evaluación con vistas a su automatización, abrió en su momento otro interrogante surgido del tipo de aplicaciones de procesamiento continuo mencionadas inicialmente ¿Se Necesita detectar o prevenir? La idea era prevenir para minimizar riesgos, motivo por el cual se incorpora un bagaje de herramientas asociadas a la minería de datos (Aleman-Beza, Halaschek, et al., 2003; Singh, Vajirkar, et al., 2003a) para actuar pro activamente, de modo predictivo e incorporar automatización en el proceso de toma de decisión, pero fundado ahora en información que proviene en base a un marco de medición y evaluación sustentado en una base ontológica (Olsina & Martín, 2004), a la cual se le incorporaba información relativa a su contexto.

1.2 Planteamiento del Problema

A diferencia de las aplicaciones de aprovisionamiento continuo de datos introducidas anteriormente, las bases de datos tradicionales han sido empleadas en situaciones donde se requería principalmente persistencia y respuesta ante consultas complejas. Estas se componen de un conjunto de tuplas relativamente estáticas, sobre las que podrían efectuarse operaciones de inserción, borrado, actualización y reorden con menor frecuencia que lo que se consultan (Golab & Özsu, 2003).

En aplicaciones tales como el monitoreo de signos vitales de pacientes, de centrales nucleares, de mercados financieros y en general, de cualquier aplicación que requiera procesamiento y verificación continua de los datos generados en tiempo real, es menester llevar adelante en forma ininterrumpida la consulta de los atributos críticos que parametrizan la naturaleza del problema u objeto (ente) bajo control, por lo que el enfoque tradicional de bases de datos basado en persistencia no sería el adecuado. De este modo, la caracterización de este nuevo tipo de aplicaciones surge bajo la denominación de *data stream, el cual se define como una secuencia en expansión continua e ilimitada de ítems de datos homogéneos que fluyen en tiempo real* (Getta & Vossough, 2003). La definición de data stream supone la idea de homogeneidad en los datos, tiempo y asume por ende una estructura (metadatos) en los mismos.

A los efectos de posibilitar la repetitividad, consistencia y automatización del proceso de medición y evaluación a partir de data streams, es necesario brindar una estructura a los datos del flujo, que permita incorporar significado del dato a procesar, como por ejemplo: tipo de dato, definición de métrica, indicadores asociados, precisión requerida, entre otros. Para ello, es necesario basarse en un marco formal de medición y evaluación con sustento ontológico que posibilite no solo la estructuración, sino también la determinación del contenido transmitido a los efectos de su procesamiento automático (Olsina & Martín, 2004). En este sentido, se parte del marco formal de medición y evaluación C-INCAMI, como referencia para la definición de metadatos asociados a las métricas, estructuración de las mediciones respectivas y base

ontológica para la incorporación de información contextual referida a la medición (Molina & Olsina, 2007; Olsina, Papa, et al., 2007).

Código del Indicador	**ETEMP**
Nombre del Indicador	Nivel de la temperatura corporal
Dominio	
(Hiperpirexia, Fiebre muy alta, fiebre, normal, riesgo de hipotermia, hipotermia)	
Escala	
Tipo de Escala	**Categórica**
Valores del Dominio	**Ordinal**
Unidad	-

Método de Cálculo (Modelo Elemental)

Nombre	**Análisis de la Temperatura**

Especificación

Temperatura Axilar (Métrica Directa)	Valor del Indicador
>= 41.50	Hiperpirexia
[38.30 , 41.50)	Fiebre muy alta
[37.50 , 38.30)	Fiebre
[36.50 , 37.50)	Normal
[35.00 , 36.50)	Riesgo de hipotermia
< 35.00	Hipotermia (A medida que la termperatura desciende, existe riesgo de encefalograma plano).

Referencias

1. Loscalzo, J., Fauci, A., Braunwald, E., Dennis L., Hauser, S., Longo, D. (2008). "Harrison's principles of internal medicine". McGraw-Hill Medical. pp. Chapter 17, Fever versus hyperthermia. ISBN 0-07-146633-9.
2. Marx, John (2006). "Rosen's emergency medicine: concepts and clinical practice". Mosby/Elsevier. p. 2239. ISBN 9780323028455.

Criterios de Decisión

Valor del Indicador	Nivel de Aceptabilidad
Hiperpirexia	**No aceptable**
Fiebre muy alta	**No Aceptable**
Fiebre	**Pobremente aceptable**
Normal	**Aceptable**
Riesgo de Hipotermia	**Pobremente aceptable**
Hipotermia	No aceptable

Tabla 1. Algunos detalles de la definición del indicador elemental "Nivel de la Temperatura Corporal" según C-INCAMI

Para aprovechar la gestión de metadatos asociados a las métricas y el resultado provisto a través de las mediciones, es necesario desarrollar un modelo de minería de datos activa que aborde el procesamiento de data stream basado en C-INCAMI a los efectos de poder detectar y/o prevenir situaciones de anomalías, tendencias o desvíos de los parámetros definidos dentro de los correspondientes metadatos del marco conceptual (Diván & Olsina, 2008). El objeto de incorporar al procesamiento un comportamiento predictivo, radica en abordar la prevención como primera alternativa y la detección como segunda alternativa. Es cierto que la predicción no representa una tautología, sino que es factible que existan errores de ajuste en la misma y que ocurran falsos positivos o negativos, pero ello no condiciona la etapa detectiva, donde ciertamente podrá indicarse si un indicador se encuentra o no en una zona de riesgo, debido a que la definición se sustenta sobre un marco formal de medición y evaluación tal y como expone la Tabla 1.

En la definición del indicador elemental “Nivel de la Temperatura corporal”, se plantean diferentes niveles de aceptabilidad con su correspondiente interpretación en base a la métrica “Valor de la Temperatura Axilar”, cuya escala es intervalo, su dominio de valores numérico, continuo y dentro de los reales positivos. Si se toma en consideración que la interpretación de los rangos expuestos en la Tabla 1 son categóricos, se tiene que la variable objetivo de la función de minería también será categórica, dado que se buscará predecir una situación futura del indicador en base a las mediciones actuales.

Se dispone entonces de un conjunto de mediciones numéricas como entrada, las cuales alimentan una serie de indicadores cuya interpretación se asocia a una variable categórica ordinal y todo ello basado en C-INCAMI, lo que posibilita su consistencia en el procesamiento automatizado. De este modo, configura adecuadamente el empleo de una función de clasificación en términos de minería de datos (Han & Lamber, 2001) a los efectos de poder “predecir” [1] *el nivel de aceptabilidad* del indicador en base a las mediciones entrantes en tiempo real.

De este modo, el núcleo de la tesis se focalizará en permitir la incorporación de fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados C-INCAMI, permitan realizar análisis estadísticos de un modo consistente a los efectos de implementar un comportamiento detectivo y a su vez, permitan incorporar información contextual a las mediciones, para enriquecer la función de clasificación con el objeto de implementar el comportamiento predictivo. Tanto la implementación del comportamiento detectivo como del predictivo tendrán asociados mecanismos de alarma, que permitirán proceder a la notificación ante la eventual identificación de una zona de riesgo.

1.3 Consideraciones y Requerimientos

Como se expuso anteriormente, la tesis se sustenta en obtener un conjunto de mediciones enriquecidas con metadatos basados en C-INCAMI desde fuentes de datos heterogéneas, para aprovisionar dos instancias: una detectiva y otra predictiva. La etapa detectiva tiene que ver con efectuar una serie de análisis estadísticos al conjunto de mediciones, para determinar si existen situaciones de riesgo en un momento dado; mientras que la etapa predictiva, tiene que ver con inferir el nivel de aceptabilidad de un indicador determinado a partir de las mediciones, a los efectos de predecir si se estaría ante una posible situación de riesgo.

La clasificación consiste básicamente en un proceso de dos etapas. La primer etapa, consiste en generar un modelo a partir de un conjunto de entrenamiento con la forma (X_1, X_2,..., X_n, Y), donde las variables ‘X_i’ se denominan variables independientes y se les asocia una etiqueta de clase o variable dependiente ‘Y’. En la segunda etapa del proceso, dicho modelo servirá para inferir la etiqueta de clase en aquel grupo de variables independientes sin etiquetar o sin valor conocido para la variable dependiente. En términos de clasificación y desde una perspectiva del

[1]. Cabe acotar que en esta tesis se entiende que clasificación y predicción son conceptos diferentes, la predicción supone que la variable objetivo es numérica mientras que la clasificación supone que la misma es categórica. Hecha la aclaración del caso, en adelante y los efectos de mejorar la comprensión del texto, se referirá en forma indistinta tanto a clasificación como a predicción

modelo relacional, es común referirse al conjunto de variables independientes y su relación con la variable dependiente como una tupla con ‘n’ atributos a la que le corresponde una etiqueta de clase dada (Han & Lamber, 2001).

La función de clasificación puede implementarse de varios modos, sea mediante redes neuronales, árboles de decisión, entre otras. En esta tesis en particular y a los efectos de acotar la misma, se ha optado por los árboles de decisión, dejando abierta la posibilidad a trabajos futuros que empleen otras alternativas de clasificación. El árbol de decisión posee una estructura de árbol (Langsam, Augenstein, et al., 1997) en donde sus nodos internos almacenan pruebas sobre un atributo, cada una de sus ramas implica una decisión efectivamente escogida sobre una prueba de atributo y cada nodo hoja representa una clase o distribución de la misma. El nodo superior, sin ‘padre’, es denominado raíz y al igual que los nodos internos plantea una prueba sobre un atributo, solo que en este caso representa la primera decisión a abordar con el objeto de recorrer el árbol.

El árbol de decisión a emplear en la etapa predictiva, será entrenado y aplicado dentro del contexto de data streams, entendiéndose por tal a un conjunto de flujos por el cual se abastecen tuplas posiblemente de forma no limitada, que arriban de un modo continuo sin mediar la necesidad de un orden predefinido, con la posibilidad de ausencia de valor o ruido en sus datos (Chaudhry, Shaw, et al., 2005).

Se asume que los datos provenientes a través de los flujos de datos representan mediciones y se estructuran de algún modo bajo el marco formal C-INCAMI. Dichas mediciones se suponen son numéricas y continuas, pudiéndose receptar una cantidad ilimitada de las mismas a partir de las fuentes y con la evolución del tiempo.

La etiqueta de clase se asociará con un criterio de decisión propio de un indicador global de C-INCAMI para un proyecto de medición y evaluación dado, lo que determinará que la cantidad de valores categóricos que podrá asumir es finita. La cantidad de métricas de las cuales depende el indicador global para obtener su valor también será finita.

La cantidad de memoria de utilizar para el procesamiento de los flujos, como así también para la generación y aplicación del árbol de decisión será finita, dependerá del proyecto de medición y evaluación y podrá escalar, siempre y cuando no comprometa la estabilidad de la unidad de procesamiento donde se esté ejecutando.

El procesamiento y aplicación de árboles de decisión sobre data streams implica una serie de consideraciones que deben tenerse en cuenta (Kirkby, 2007):

- **Procesar un caso a la vez e inspeccionarlo solo una vez en lo posible**: Las mediciones deberán ser analizada en términos de outliers (Johnson, 1998) y/o presencia de ruido sobre el mismo arribo, aplicando instantáneamente el árbol de decisión, permitiendo la obtención de la decisión como así también la actualización incremental del árbol para situaciones futuras.

- **Utilizar una cantidad limitada de memoria**: Las mediciones pueden arribar en forma ilimitada en términos temporales y requerirán ser procesadas como así también clasificadas a los efectos de valorar la decisión. Esto implica un riesgo en términos de memoria y capacidad de procesamiento que debe ser delimitado claramente a los efectos de garantizar la estabilidad del conjunto como sistema.

- **Trabajar en una cantidad limitada de tiempo**: Dado el arribo ilimitado de las mediciones, la cantidad de unidades de tiempo que podrán asignársele a cada grupo de medición con el objeto de procesarla y clasificarla es finita, dado que un exceso o sobrestimación repercutirían en generación de colas de servicio o recursos ociosos respectivamente.

- **Poder clasificar en cualquier momento**: El modelo de decisión debe poder clasificar en cualquier instante de tiempo, incluso cuando se encuentre procesando nuevas mediciones o bien, cuando no tenga ninguna medición aún con la cual actualizarse.

1.4 Principales Contribuciones

Como se ha mencionado anteriormente, la tesis se centra en generar un enfoque (centrado en modelos) que permita integrar fuentes de datos heterogéneas, que aprovisionen mediciones enriquecidas en base a C-INCAMI a dos instancias claramente diferenciadas: una detectiva y otra predictiva. La instancia predictiva buscará hacer uso de la información contextual y el conocimiento de la historia reciente del data stream a los efectos de predecir situaciones de riesgo; mientras que la instancia detectiva aplicará una serie de análisis estadísticos para intentar determinar si existen situaciones de riesgo en base a la definición formal del proyecto de medición y evaluación según C-INCAMI según las mediciones informadas. Con ello se pretende contribuir en los siguientes aspectos:

- **Desde el punto de vista de las fuentes de datos y su procesamiento**:
    - ***Definir una Interfase abierta que permita el empleo de C-INCAMI en cualquier dispositivo de medición***: Definir una interfase abierta que permita traducir las mediciones desde los dispositivos a un formato de intercambio basado en C-INCAMI, posibilitando la incorporación de información contextual sobre el arribo del dato y gestionando la comunicación de las mediciones. Se pretende que la interface sea abierta, entendiéndose por tal a no propietaria, de modo que cualquier dispositivo pueda emplearla si desease utilizar el marco de medición y evaluación.

    - **Permitir un borrado selectivo ante potenciales desbordes en colas de servicio basado en metadatos C-INCAMI**: Ante la situación en que la generación de mediciones provenientes desde diferentes dispositivos de medición, supere la tasa de procesamiento de datos y desborde la cola de servicio de cada procesador, los metadatos embebidos en las mediciones permiten diferenciar claramente la

semántica de cada una de ellas y el procesador podrá efectuar un descarte selectivo de la cola de modo que no perjudique los modelos de decisión a aplicar a posteriori.

- **Desde el punto de vista detectivo**:
  - **Permitir un análisis de datos más consistentes**: La incorporación de metadatos enriquecidos con factores contextuales, permite diferenciar el contexto de medición lo que aporta mayor información, disminuyendo por ende la incertidumbre y de este modo posibilitando un tratamiento más adecuado del ruido, valores faltantes y/o valores extremos.

- **Desde el punto de vista predictivo**:
  - **Generar y actualizar incrementalmente el modelo de clasificación adecuado al contexto**: Ante el arribo y actualización continua de las mediciones, se busca generar modelos de clasificación incrementales que incorporen la información contextual, con el objeto de ajustar la precisión de clasificación basada en el contexto. El hecho de que la actualización sea incremental es una restricción de contexto propia de los data streams (ver sub-sección 1.3), donde no es posible un proceso por lote de generación de modelos en tiempo de ejecución, dado que el arribo de datos es continuo y éstos empleados en un lote revestirían el carácter de históricos, no siendo los mismos adecuados para una clasificación con vistas a disparar alarmas proactivas.

  - **Promover acciones proactivas basadas en el contexto de medición:** La idea de generar incrementalmente un modelo de clasificación enriquecido con factores contextuales con vistas a incrementar su precisión, tiene por objeto servir de entrada a los criterios de decisión definidos mediante el marco de medición y evaluación, para habilitar la ejecución, si correspondiere, de una serie de alarmas preventivas con vistas a evitar daños mayores en la entidad bajo medición.

- **Desde el punto de vista de implementación y simulación del enfoque integrado**:
  - **Desarrollar una herramienta prototípica que implemente el procesamiento de mediciones y el modelo de decisión incremental:** Se desarrolló un prototipo que permite gestionar y aplicar la minería de datos al flujo de datos en el marco de la administración de metadatos y mediciones, para luego ser visualizada de un modo adecuado al proceso de toma de decisión. El prototipo se construyó partiendo de tecnología de desarrollo JAVA y haciendo su código libre bajo licencia Open Source a los efectos de que se pueda contribuir libremente a los avances del prototipo y del conocimiento en la temática

  - **Incorporar la funcionalidad de Análisis Preventivo en base a retroalimentación**: A partir de los datos y metadatos del proyecto de medición y evaluación definido en

base a C-INCAMI, el comportamiento detectivo (Análisis Estadístico) y el modelo de clasificación incremental implementados dentro del prototipo, permiten el descubrimiento de asociaciones entre la variación de indicadores o grupos de indicadores, con el objeto de contribuir a mejorar el proceso de toma de decisión.

- **La simulación resultante de aplicar el enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, permite validar el mismo sobre un caso de aplicación y definir los umbrales de aplicación.** La simulación efectuada sobre la definición de un proyecto de medición y evaluación de pacientes trasplantados ambulatorios, ha permitido validar inicialmente el comportamiento detectivo como así también el predictivo, facilitando la recolección de tiempos involucrados en cada etapa de procesamiento, lo cual es fundamental para delinear áreas potenciales de aplicación del enfoque.

1.5 Estructura de la Tesis

La presente tesis se estructura partiendo del estado del arte, en la que se analiza la situación actual de los modelos, conceptos y herramientas involucradas, a los efectos de discutir al enfoque integrado de procesamiento de flujos de datos basado en metadatos de mediciones, como una contribución válida que cubre un nicho específico. Posteriormente, a lo largo de este trabajo, se desarrolla el enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, analizando y elaborando cada una de sus componentes. En síntesis, el resto de los capítulos son los siguientes:

- **Capítulo 2**: Se aborda el Estado del Arte donde inicialmente se discute una serie de conceptos previos, se analizan los marcos de medición y evaluación, se analiza el procesamiento de flujos de datos y se hace hincapié en la diferencia paradigmática con respecto al contexto de bases de datos tradicionales para luego, avanzar con el estudio de los principales algoritmos de clasificación incremental basado en árboles de decisión sobre data streams y concluir, con la discusión sobre la necesidad del enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones.

- **Capítulo 3**: Se plantea un modelo integrado para el procesamiento de flujos de datos (data streams) y para el soporte al proceso de toma decisión, basado en datos con semántica incorporada en base a C-INCAMI. Aborda cada una de sus componentes y explica el rol de cada uno de ellos, como así también el efecto de su interacción con los restantes elementos que integran el modelo.

- **Capítulo 4**: Se aborda C-INCAMI y sus elementos subyacentes a nivel conceptual. En este sentido se revisará la propuesta del marco y se explicitará cada uno de los conceptos vertidos en el mismo, a los efectos de consensuar los aspectos ontológicos básicos que, a posteriori, sustentarán el modelo integrado para el procesamiento de flujos de datos.

- **Capítulo 5**: Se inicia el detalle del modelo integrado para el procesamiento de flujos de datos, a través de la problemática de configuración de las fuentes de datos heterogéneas. Se plantean los mecanismos y las secuencias de actividades necesarias para lograr la integración de dichas fuentes al modelo de procesamiento.

- **Capítulo 6**: Se aborda la problemática de la captura de los datos desde distintas fuentes como así también su transmisión, lo que representa la definición formal de los data stream que pueden intervenir en el modelo. Adicionalmente, se expone el mecanismo de incorporación de metadatos en forma conjunta con los datos y las estrategias de abastecimiento/recolección de los mismos.

- **Capítulo 7**: Se exponen las diferentes técnicas estadísticas y la estrategia de su aplicación a los ítems de datos provenientes desde los data streams. Se fundamentará de un modo sintético cada técnica, la necesidad de emplear la misma en el presente contexto y el modo en que se configurarán las mismas para depurar los ítems de datos.

- **Capítulo 8**: Se evalúan diferentes métodos de clasificación incrementales en base a árboles de decisión sobre data stream, los cuales serán comparados en función de datos y metadatos provenientes desde flujos basados en C-INCAMI según el modelo integrado de procesamiento abordado en el capítulo 3. El método que mayores méritos en la clasificación incremental posea, será incorporado como pieza fundamental del núcleo de toma de decisión del modelo.

- **Capítulo 9**: El capítulo plantea un caso de aplicación del enfoque integrado de procesamiento de flujos de datos presentado en el capítulo 3, como así también su implementación prototípica y simulación asociada.

- **Capítulo 10**: Sintetiza las principales contribuciones realizadas en la presente tesis y finaliza con los lineamientos fundamentales de los potenciales trabajos a futuro.

2 Estado del Arte

El capítulo 1 ha presentado la motivación y antecedentes de la presente tesis, seguido del planteamiento del problema como así también de las consideraciones y requerimientos asociados al mismo. Ha esbozado las principales contribuciones para culminar con la estructuración de la presente tesis.

En este capítulo se introduce una serie de conceptos previos, necesarios para avanzar en el desarrollo del problema, entre los que se encuentran conceptos asociados a minería de datos (data mining), flujos de datos (data streams), minería sobre flujos de datos (mining data streams), contexto (context) y orientación al contexto (context-aware).

Una vez introducidos estos conceptos, se analizan los marcos de medición y evaluación, a los efectos de seleccionar aquél que pueda garantizar repetitividad y consistencia en los procesos de medición y evaluación, que a su vez dan soporte al análisis y en definitiva a la toma de decisión. En este sentido, se analizan los marcos denominados Balanced-Scorecard, Goal Question Metric y C-INCAMI dado que son ya representativos en ámbitos industriales y/o académicos.

A seguir, se discute la minería de datos orientada al contexto (Context-Aware Mining), a los efectos de determinar el grado de formalización de los datos para con el proceso de minería basada en información contextual, y qué nicho de éste área permanece con aspectos pendientes o con oportunidad de un mejor abordaje conforme el objetivo de nuestro trabajo. Se exponen algunos casos destacados de aplicación, en donde se demuestra el accionar del contexto y su incidencia en determinados dominios.

A continuación, se analizan los sistemas de gestión de flujos de datos (Data Stream Management Systems -DSMS-) para determinar qué alternativas existen, como así también qué cuestiones permanecen sin resolver, a los efectos del procesamiento de flujos de datos cuyo comportamiento es instruido a partir de metadatos incorporados en el mismo.

Luego, se discuten las distintas familias de algoritmos que permiten la aplicación de árboles de clasificación sobre data

streams, a los efectos de seleccionar aquella familia de algoritmos que menor orden de procesamiento y consumo de memoria posean. En este sentido, se analizan aquellos que ya son representativos en ámbitos industriales y/o académicos tales como Very Fast Decision Tree Learner (VFDT), Streaming Ensemble Algorithm (SEA), entre otros.

Finalmente, el capítulo plantea la necesidad del enfoque integrado de procesamiento de flujos de datos, discutiendo qué aspectos de los conceptos asociados y planteados en el capítulo, permanecen al momento de la escritura de esta tesis sin una respuesta concreta y cuáles podrían ser mejorados a través de dicho enfoque.

2.1 Conceptos Previos

Como se ha expuesto en la sub-sección 1.2, el núcleo de la tesis se focalizará en incorporar fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados en C-INCAMI, permitan realizar análisis estadísticos de un modo consistente a los efectos de implementar un comportamiento detectivo y a su vez, permitan incorporar información contextual a las mediciones, para enriquecer la función de clasificación con el objeto de implementar el comportamiento predictivo.

De este modo y para una mejor comprensión, es necesario introducir conceptos fundamentales relacionados al núcleo del trabajo tal como *data mining,* vinculado a la etapa predictiva y consiguiente toma de decisión; *data streams,* concepto en el que se basa el aprovisionamiento de mediciones para su posterior procesamiento; *mining data streams,* necesario para aspectos del comportamiento predictivo aplicado a los flujos de datos; *context* a los efectos de entender qué se entiende por el mismo y cómo sirve el concepto para delimitar la ocurrencia de un suceso o situación, con respecto al ente de medición; y finalmente, *context-aware* que permite comprender qué implica la orientación al contexto, para luego poder utilizar dicho concepto dentro del flujo de mediciones, ya que la idea del flujo de mediciones como se ha expuesto, es que permita la incorporación de información contextual a los efectos de posibilitar análisis más consistentes y comparables.

2.1.1 Data Mining

"*Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories*" (Han & Lamber, 2001). Sobre la definición de Han & Lamber debe hacerse una aclaración importante a los efectos de diferenciar el concepto de dato con respecto al de información. Los autores implícitamente emplean ambos términos en forma indistinta, ya que la definición indica que el descubrimiento de conocimiento surge a partir de grandes almacenes de datos y al final de la definición se señala: "u otros repositorios de información". Por ello, es necesario hacer algunas acotaciones:

- Por dato se entiende a la representación de un antecedente (hecho, evento, etc.) captado y registrado en algún medio de almacenamiento necesario para llegar a un conocimiento (Diván, 2007)

- Por información se entiende al dato que satisface las características de oportunidad, interés, veracidad y consistencia en forma simultánea (Diván, 2007)

- Por conocimiento se entiende la información no trivial y factible de ser aplicada en un contexto determinado a los efectos de lograr la consecución de un objetivo concreto (Han & Lamber, 2001)

En este sentido, se puede reescribir la definición de Han & Lamber como sigue: "*La minería de datos es la aplicación de técnicas en grandes volúmenes de datos para descubrir información interesante, útil, aplicable y no trivial*"

2.1.2 Data Streams

"*A data stream is a possibly unbounded sequence of tuples*" (Koudas & Srivastava, 2005). La definición indica desde un principio, la diferenciación con el esquema de bases de datos tradicional desde el punto de vista que el 'data stream' no tiene limitación y se puede abordar como secuencia de tupla. Esta secuencia dará lugar a la noción de tiempo y por consiguiente, determinará una posibilidad de abordaje mediante esquemas de series temporales (Pérez López, 2005) desde la perspectiva de minería de datos.

"*For unbounded data streams many queries are interested in a subset or part of the complete stream"* (Chaudhry, Shaw, et al., 2005). Con vistas a efectuar el procesamiento del data stream, es posible interpretar una ventana como subconjunto del flujo, lo que permitirá disponer de un valor finito de tuplas a los efectos del cómputo. Existen diversas técnicas para obtener este subconjunto de tuplas (Chaudhry, Shaw, et al., 2005):

- **Time-Based Window (o físicas)**: El tamaño de la ventana estará expresado por el número de tuplas que arriben en un lapso de tiempo fijo. Por ejemplo: todas las tuplas que han arribado en los últimos 120 segundos.

- **Tuple-Based Window (o Lógicas)**: El tamaño de la ventana estará expresado por un número fijo de tuplas. Por ejemplo: 200 tuplas por ventana, independientemente del tiempo que insuma el arribo de las mismas.

- **Sliding Window**: El ancho de la ventana es fijo y los nuevos valores que arriban van sustituyendo a las viejas tuplas mediante un esquema de colas (FIFO).

- **Landmark Window**: Uno de los extremos de la ventana permanece fijo mientras el segundo extremo de la ventana se mueve incorporando nuevas tuplas. Por ejemplo: transferencias en dólares de los clientes de un banco desde la última vez que el dólar superó los 3 pesos.

2.1.3 Mining Data Streams

"*Mining Data Streams is concerned with extracting knowledge structures represented in models and patterns in non stopping streams of information*" (Medhat Gaber, Zaslavsky, et al., 2005b). Es menester aclarar que el flujo es de datos no de información. Lo que ocurre es que muchos autores hacen la aclaración al inicio del texto sobre la diferencia de los conceptos y dependiendo del objetivo del documento, pueden luego emplearlos como sinónimos a lo largo del mismo.

"*Mining data streams has raised a number of research challenges for the data mining community. These challenges include the limitations of computational resources, especially because mining data streams of data most likely be done on a mobile device with limited resources. Also due to the continuality of data streams, the algorithm should have only one pass or less over the incoming data records*" (Medhat Gaber, Zaslavsky, et al., 2005a). Los desafíos señalados por los autores con respecto a la aplicación de métodos y técnicas de minería de datos sobre data streams, reafirman las diferencias paradigmáticas con el contexto de bases de datos tradicional. La minería de datos sobre data streams, por definición, se encuentra limitada temporalmente y computacionalmente con respecto a la lectura del dato como así también por ende a la generación y/o actualización del modelo. Del mismo modo el recolector de datos, el cual también podría emplear mecanismos inferenciales, en general es representado por dispositivos móviles cuyo poder computacional es limitado y su principal rol viene de la mano con el tratamiento previo de los datos en post de garantizar su consistencia.

2.1.4 Context y Context-Aware

"*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*" (Dey, 2001).

Desde el punto de vista de la ejecución de una medición, el concepto reviste una particular importancia, ya que dado una entidad bajo medición, cualquier persona, objeto y/o características del sitio donde reside la misma, que sean plausibles de medición, contribuirán a disminuir la incertidumbre sobre ésta y permitirán analizar las correlaciones entre sus variables representativas.

"*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on user's task*" (Dey, 2001). A partir de lo señalado por Dey, un sistema orientado a la medición es context-aware, en la medida que incorpore algún mecanismo que permita extraer datos útiles referenciales al contexto de la entidad bajo análisis. En este último sentido, las métricas orientadas a parametrizar el contexto que circunda a la entidad bajo medición, representan un mecanismo válido de abstracción que permite incorporar en el modelo información complementaria a la medición efectuada sobre la propia entidad. Por ejemplo, si la temperatura corporal de una persona hipertensa es 36,5°C, en forma aislada nos diría que la temperatura de la misma estaría dentro de los parámetros normales, pero si adicionalmente se tomase la temperatura ambiente inmediata a la posición de la misma y nos

arrojase 40°C, este complemento nos estaría indicando un potencial riesgo para la persona (Petersa, Becketta, et al., 2009; Diván & Olsina, 2009a).

2.2 Marcos de Medición y Evaluación

Si se desea garantizar la repetitividad y consistencia de un proceso de medición y evaluación, se entiende que el proceso debe estar bien definido, como así también sustentarse en un marco con base conceptual robusta, como por ejemplo, en una ontología. De lo contrario, se estaría llevando adelante procesos de medición no repetibles y peor aún, inconsistentes, al no tener en claro lo que se entiende por métrica, indicador, entidad y otros conceptos asociados, como unidad, escala, tipo de escala, método de medición y/o cálculo, etc. Por lo que la ausencia de un marco de medición y evaluación que permita la formalización de los conceptos involucrados (que implique la clara especificación de datos y metadatos), puede dificultar los procesos de análisis en cuanto a la comparabilidad y consistencia en el procesamiento, y comprometer la calidad en la toma de decisión.

Como se manifestó en la sub-sección 1.2, el núcleo de la tesis se focalizará en permitir la incorporación de fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados en C-INCAMI, permitirán realizar análisis estadísticos de un modo consistente, con vistas a implementar un comportamiento detectivo y a su vez, admitan incorporar información contextual a las mediciones. Así y dado que se persigue integrar fuentes de datos heterogéneas, debe existir un modo común de informar los datos conjuntamente con sus metadatos, motivo por el cual la base conceptual debe estar claramente definida a los efectos de homogenizar los conceptos y sus datos a informar. De este modo, la tesis hace principal hincapié en el empleo de marcos con bases conceptuales robustas, para que mediante el reuso de los mismos, se permita estructurar los datos y sus metadatos (cualquiera sea su origen) en forma transparente y consistente, para fortalecer los aspectos asociados a la comparabilidad y consistencia en el procesamiento de los mismos.

A continuación, se analiza un conjunto de marcos de medición ya referenciados en ámbitos industriales y/o académicos, que podrían utilizarse en caso de que se requiera llevar adelante un proyecto de medición. Los mismos, si bien enfocados originalmente desde puntos de vistas diferentes, tanto en lo conceptual como en lo operativo, buscan establecer métricas que permitan determinar un valor numérico, con el objeto de saber empíricamente si un objetivo es o no satisfecho sobre una entidad bajo análisis. Así, los mismos buscan a su modo, formalizar de alguna manera el proceso de medición y evaluación, a los efectos de garantizar su repetitividad y facilitar la comparación entre mediciones realizadas en diferentes instantes de tiempo (y aún en diferentes proyectos a nivel organizacional o inter-organizacional).

2.2.1 Balanced-Scorecard

El Cuadro de Mando Integral o Balanced Scoredcard (Kaplan & Norton, 1992; Kaplan & Norton, 1993), fue desarrollado por Robert Kaplan y David Norton en 1996, los autores plantearon un sistema de gestión y planificación estratégica con el objeto de medir la performance asociada a la implementación de la planificación estratégica en una organización.

Balanced Scorecard traduce los planes estratégicos de una organización en definición concreta de objetivos, umbral o valor a lograr en un período de tiempo dado y la métrica propuesta para su medición. La traducción de los planes estratégicos es en forma coordinada y se sustenta sobre las perspectivas financieras, del cliente, los procesos de negocios interno y aprendizaje, retroalimentación y crecimiento como puede apreciarse en la Figura 1.

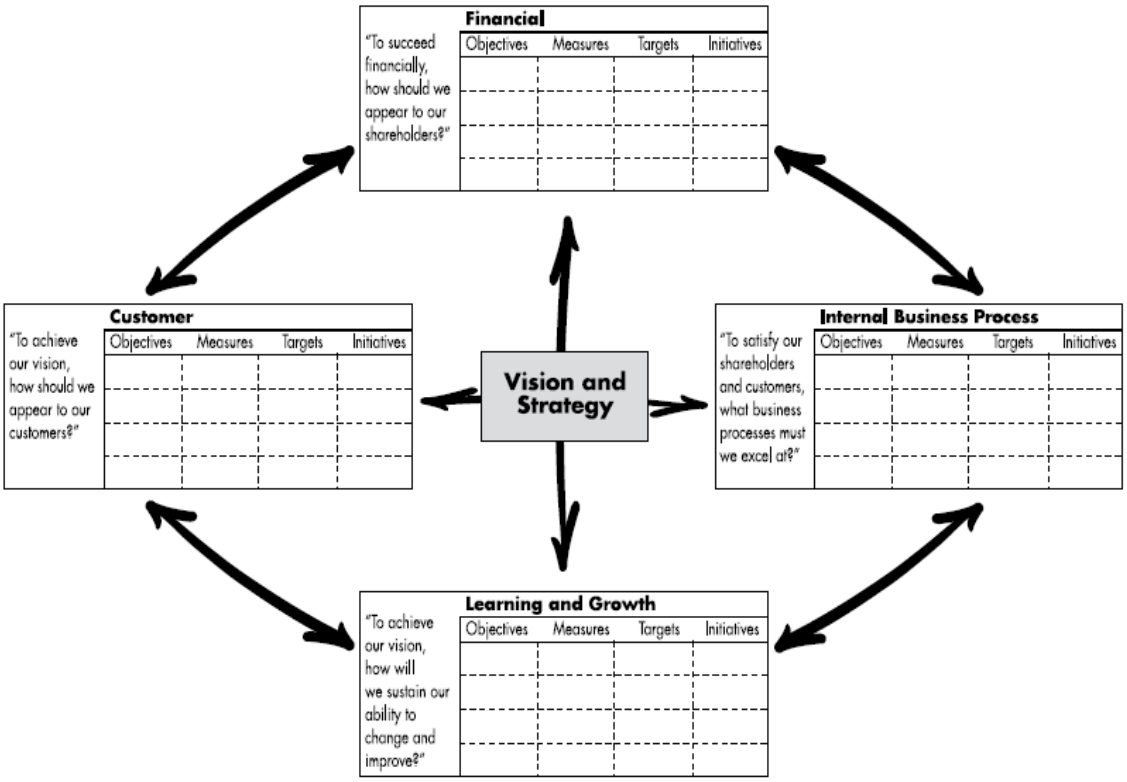


Figura 1. Traduciendo Visión Estratégica: Las Cuatro Perspectivas según (Kaplan & Norton, 1996)

- **Financiera**: Define una serie de objetivos financieros alineados con el plan estratégico, junto con el objetivo a lograr, su umbral, la métrica para determinar el estado actual y una serie de iniciativas que se abordarán para cada objetivo.

- **Cliente**: Esta visión plantea el cómo debiera aparecer la empresa ante sus clientes de cara a satisfacer el plan estratégico. De este modo, definen una serie de objetivos con respecto al modo de abordaje de los clientes con respecto al plan estratégico, fijan el umbral a lograr, la métrica a emplear para determinar el estado actual y la serie de iniciativa que acompañan a cada objetivo.

- **Procesos de Negocios Internos**: Se focaliza sobre los procesos de negocios que se deben potenciar dado el plan estratégico organizacional e involucrando a empleados y accionistas. Así, definen los objetivos a lograr, umbral, las métricas e iniciativas asociadas.

- **Aprendizaje y Crecimiento**: Se centra en los mecanismos de cambio y mejora a sostener de modo de promocionar la implementación del plan estratégico. Define una serie de objetivos consistentes con la promoción del plan, el umbral, métrica e iniciativas a desarrollar.

En general, todo objetivo tiene asociado una métrica y un umbral, el umbral representa la cuantificación del objetivo y es el valor que se desea lograr. La métrica, representa la regla de cálculo mediante la cual se cuantificará la situación actual del objetivo planteado, de este modo, mediante la comparación del valor resultante de la instanciación de la métrica (medición) con el umbral, se sabe si se ha satisfecho el objetivo y se lo puede monitorear de este modo a lo largo del tiempo.

Como puede apreciarse en la Figura 2, la construcción del cuadro de mando integral es un proceso iterativo, que parte de la clarificación de la estrategia y la búsqueda de consenso (Proceso de Traducción de la Visión); en segundo lugar define los objetivos, premios sobre logros, comunicación y educación institucional (Proceso de Comunicación e Integración); en tercer lugar establece las metas, alinea la planificación con el plan estratégico, asigna los recursos y define puntos de control (Proceso de Planificación de Negocio) y finalmente, articula la visión compartida organizacionalmente, obtiene la retroalimentación de implementar los planes hasta un punto de control dado, aprende y revisa la estrategia para volver a comenzar desde el proceso de traducción de la visión.

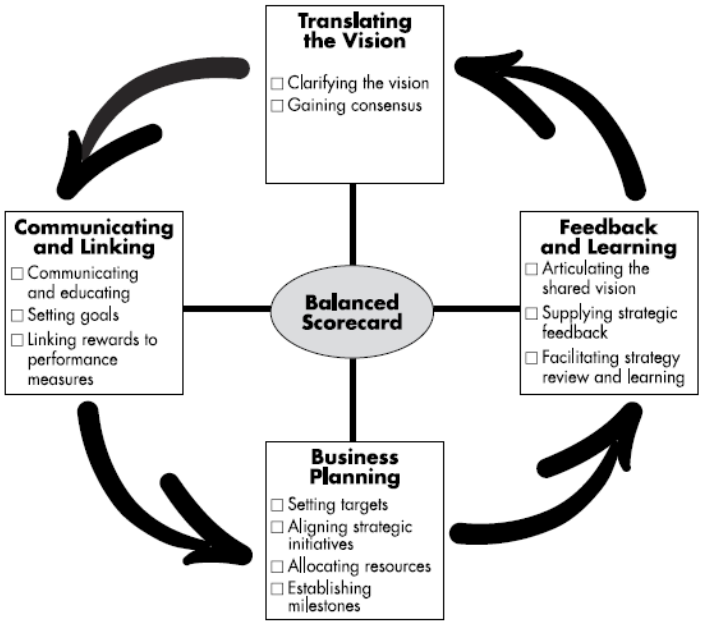


Figura 2. Gestionando la Estrategia: Los Cuatro Procesos

El cuadro de mando integral tiene como virtud su pensamiento pragmático y la fiel convicción, desde el punto de vista de la administración, de alinear el proceso de medición al plan estratégico de la empresa basado en las cuatro perspectivas indicadas. Se encuentra orientado a

una forma iterativa de definir el proceso de medición pero carece de sustento conceptual explícito. Ello indica que no cuenta ni siquiera con un glosario que permita diferenciar el concepto de escala, unidad, métrica, indicador, entre otros conceptos fundamentales. Esto último, plantea inconvenientes importantes, por cuanto confunde la idea de métrica con indicador, es decir, confunde la idea de la definición de la regla de cálculo que obtendrá una medida dada, con la interpretación y los criterios asociados a la misma. En este sentido, debe recordarse que el principal capital del cual pretende hacer uso esta tesis, es la base conceptual sobre la que se sustenta un proceso de medición y evaluación que debe ser consistente y repetible, por cuanto necesitamos homogenizar y estructurar datos y conceptos provenientes desde fuentes heterogéneas, para fortalecer los aspectos asociados a la comparabilidad y consistencia en el procesamiento de los mismos.

Adicionalmente, el cuadro de mando integral no plantea nada con respecto al modelado del contexto en el cual la medición es obtenida, lo que incluso desde el punto de vista de la administración tendría grandes ventajas a los efectos interpretativos.

Otra debilidad crítica del cuadro de mando integral, es que no modeliza el objeto o ente bajo medición, y por ende no está claramente especificado qué se está midiendo, pudiéndose tornar ambigua cualquier tipo de comparación por cuanto no sabemos si las mediciones a comparar corresponden al mismo objeto.

Una fortaleza del cuadro de mando integral radica en el análisis basado en dimensiones, lo cual permite agrupar lógicamente las métricas por perspectiva. Sin embargo, como se indicó anteriormente, al carecer de una base conceptual explícita y robusta, dificulta enormemente llegar a acuerdos en la práctica sobre cuestiones fundamentales de un proceso de medición, tal cómo el significado de escala, unidad, dominio, indicador elemental y global, métricas directas e indirectas, entre otros conceptos.

2.2.2 Goal Question Metric

Este enfoque está basado en el supuesto de que para que una organización mida en un determinado modo, ésta debe primero definir para sí misma su objetivo y sus proyectos. A partir de ello, se deberán establecer la vinculación de los objetivos a los datos con los que se relaciona, a los efectos de fijar objetivos operacionales (Basili, Caldiera, et al., 1994).

El resultado de la aplicación de Goal Question Metric (GQM), como puede observarse en la Figura 3, es la especificación de un modelo de medición con tres niveles: conceptual, operacional y cuantitativo.

El nivel conceptual está definido por la determinación de los objetivos sobre un objeto, entendiéndose por un objeto o ente de medición a un producto, proceso o recurso. El objetivo (o meta) se define para un objeto en base a varios aspectos de calidad, desde varios puntos de vista de usuario y relativo a un entorno en particular.

El nivel operacional, es representado a través de un conjunto de preguntas que caracterizan el modo en que el aseguramiento de un objetivo específico será ejecutado, basado en algún modelo característico.

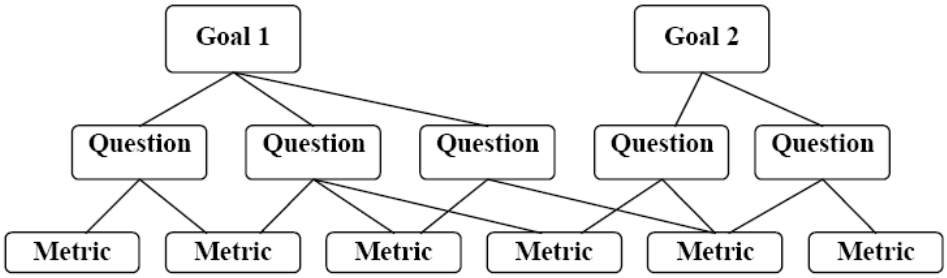


Figura 3. Goal Question Metric. Niveles del Modelo de Medición (Basili, Caldiera, et al., 1994)

El nivel cuantitativo es representado por un conjunto de métricas que están en función de cada una de las preguntas del nivel operacional, con el objeto de responder desde un punto de vista cuantificado a la misma. En este último sentido, debe tenerse en cuenta que los datos asociados a la métrica pueden ser objetivos, si se asocian a un mecanismo de medición aplicado directamente al objeto bajo análisis sin intervención de quien toma la medición, o bien subjetivos, si en el proceso de toma de medición interviene el punto de vista de quien mide además del dato tomado desde el objeto (es decir, la medida depende del juicio humano).

Si bien es un modelo pragmático, difundido en la academia y en la industria, y de relativamente fácil aplicación, se torna subjetivo por cuanto las métricas dependen de preguntas, que en realidad buscan delinear un objetivo de medición que no queda claro y no se vinculan de modo alguno, a atributos asociados al objeto bajo medición. El objeto de medición o entidad bajo análisis, no está formalizado en el modelo, al igual que los términos relacionados a métricas, con lo que la comparación de resultados podría tornarse inconsistente. Esto último, plantea la principal limitante para su reuso en esta tesis como marco de medición y evaluación, dado que al no tener una base conceptual explícita y robusta, se torna complejo la estructuración y homogenización de los datos y sus conceptos asociados para su posterior procesamiento, máxime ante un entorno en el que las fuentes de datos son heterogéneas por definición.

Adicionalmente, GQM no modeliza cuestiones contextuales que permitan asociar al objeto bajo medición en su entorno o situación y peor aún, confunde conceptos de métrica (para medición) con los de indicador (de utilidad para evaluación e interpretación). De hecho, no plantea criterios de interpretación bien establecidos (indicadores) para los resultados de las métricas, entre otros aspectos.

2.2.3 C-INCAMI

C-INCAMI (Olsina, Papa, et al., 2007; Molina & Olsina, 2007) es un marco conceptual que define los módulos y conceptos que intervienen en el área de medición y evaluación, para organizaciones de software. Se basa en un enfoque en el cual la especificación de requerimientos, la medición y evaluación de entidades y la posterior interpretación de los resultados están

orientadas a satisfacer una necesidad de información particular. Está integrado por los siguientes componentes principales:

- Definición del Proyecto de Medición y Evaluación
- Definición y Especificación de Requerimientos no Funcionales
- Especificación del Contexto del Proyecto
- Diseño y Ejecución de la Medición
- Diseño y Ejecución de la Evaluación
- Especificación del Análisis y Recomendación

La mayoría de los componentes están soportados por muchos de los términos ontológicos definidos en (Olsina & Martín, 2004; Olsina, Papa, et al., 2007).

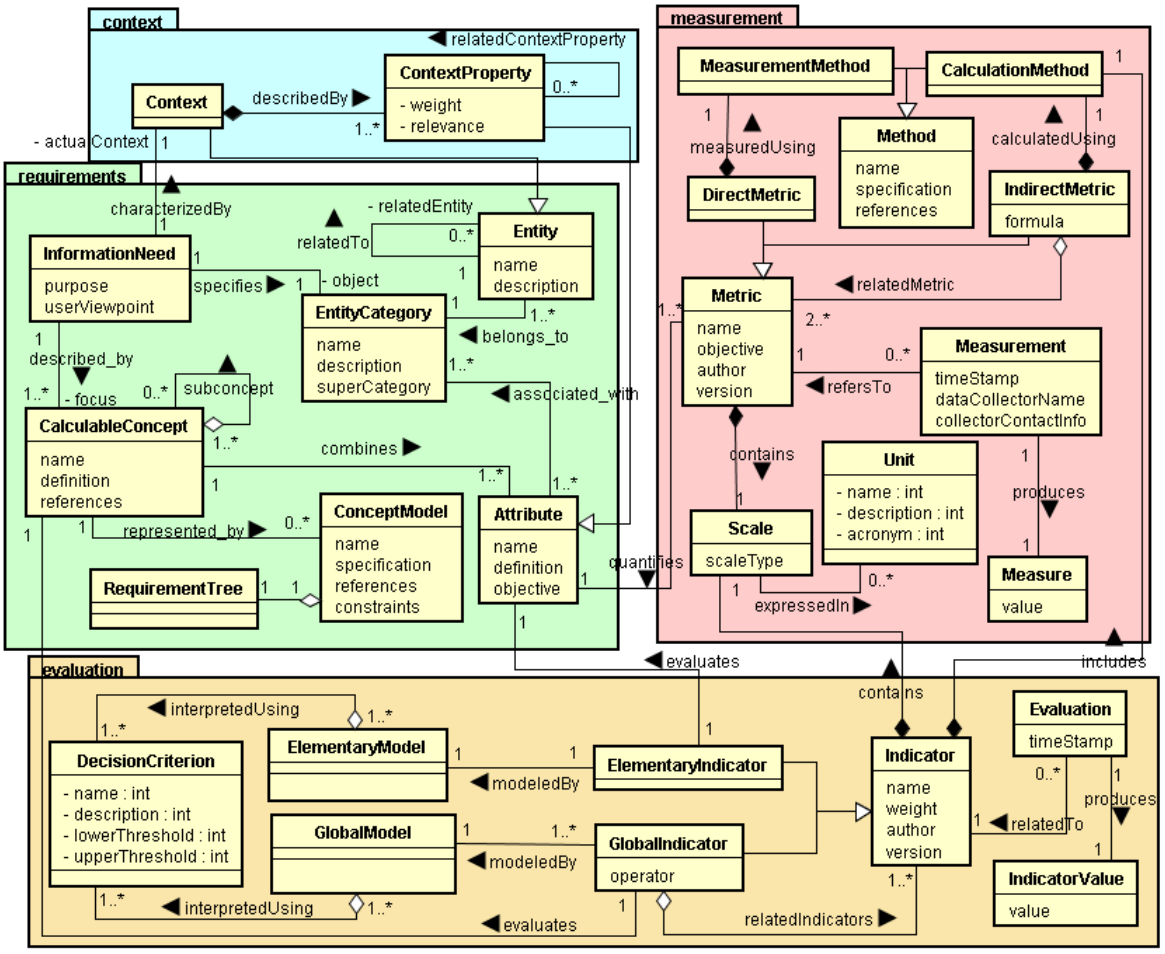


Figura 4. Principales Conceptos y Relaciones para la Especificación de Requerimientos No Funcionales, Especificación de Contexto y de los Componentes Diseño y Ejecución de la Medición y Evaluación

La Figura 4 muestra un diagrama en donde se especifican los componentes de contexto, requerimientos, medición y evaluación, como así también los principales términos y sus

relaciones. En lo que sigue se resaltará el texto con itálica cuando se mencionen términos y atributos que forman parte de estos componentes.

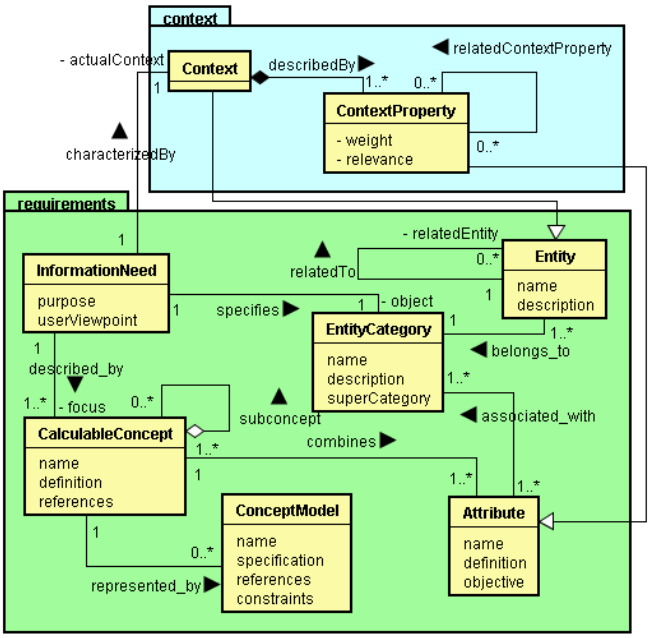


Figura 5. Principales Conceptos y Relaciones de los paquetes *requirements* y *context*

El componente *Definición del Proyecto de Medición y Evaluación*, permite definir el proyecto sobre el que se especificarán luego, los requerimientos no funcionales para las actividades de medición y evaluación. Dentro de los datos asociados a la definición del proyecto, pueden encontrarse su nombre o denominación, período de vigencia, director, entre otros datos asociados.

El componente *Definición y Especificación de Requerimientos no Funcionales*, permite especificar de forma concisa la necesidad de información (*Information Need*) de un proyecto de medición y evaluación. Generalmente surge de un objetivo específico a nivel de proyecto u organización. La necesidad de información describe el objetivo o propósito (*purpose*) y el punto de vista de usuario (*userViewpoint*). A su vez, identifica un concepto calculable (*CalculableConcept*) y especifica la categoría de las entidades a evaluar (*EntityCategory*). Un concepto calculable (característica o dimensión) puede ser definido como una relación abstracta entre atributos de un ente y una necesidad de información, el cual puede ser representado por un modelo (*ConceptModel*). En C-INCAMI, la calidad es medida y evaluada mediante la cuantificación de conceptos de menor nivel de abstracción como son los atributos (*Attribute*) de un ente. En la Figura 5, se pueden visualizar los términos, atributos y relaciones que forman parte del componente citado (paquete *requirements*).

El componente *Especificación del Contexto del Proyecto*, permite describir de forma estructurada el contexto relevante (*Context*) para una necesidad de información por medio de propiedades (*ContextProperty*). Observar en la Figura 5 (paquete *context*) que Context es un tipo

de Entity y una ContextProperty es un tipo de Attribute. Para mayor información sobre modelado contextual en C-INCAMI remitirse a (Molina & Olsina, 2007).

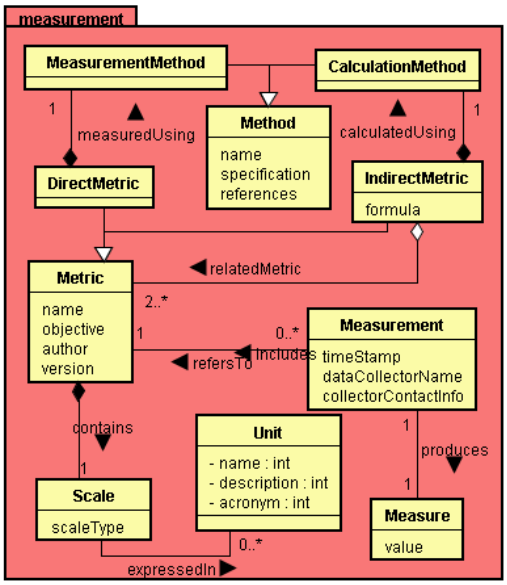


Figura 6. Principales Conceptos y Relaciones del Paquete *measurement*

El componente *Diseño y Ejecución de la Medición* (paquete *measurement* en Figura 4 y Figura 6), permite especificar las métricas utilizadas en la medición y registrar los valores medidos de los atributos de cada ente. Un atributo puede ser cuantificado por muchas métricas, pero en un proyecto de medición específico sólo se utiliza una métrica para su cuantificación. La métrica (*Metric*) define el método de medición o cálculo (*Measurement Method* o *Calculation Method*) para obtener el valor del atributo y la escala (*Scale*) de los valores. Un método de medición se aplica a una métrica directa (*Direct Metric*), mientras que un método de cálculo (en el cual interviene una fórmula) se aplica a una métrica indirecta (*Indirect Metric*). Una vez que las métricas fueron seleccionadas, se utiliza su definición para efectuar la medición (*Measurement*) y así producir una medida (*Measure*) para cada atributo. No obstante, el valor obtenido por una métrica en particular no representa el nivel de satisfacción de un requerimiento elemental (atributo) sino que es necesario realizar una nueva correspondencia utilizando indicadores.

El componente *Diseño y Ejecución de la Evaluación* permite especificar indicadores que son la base para la interpretación de atributos y conceptos calculables. Hay dos tipos de indicadores: elementales y globales. Se define un indicador elemental (*Elementary Indicator*) como aquel que no depende de otros indicadores para evaluar o estimar un concepto de más bajo nivel de abstracción como son los atributos. Por otro lado, un indicador parcial o global (*Global Indicator*) es derivado de otros indicadores para evaluar o estimar un concepto de alto nivel de abstracción (como son los subconceptos y conceptos calculables, por ej. de un modelo de calidad). El valor del indicador global finalmente representa el grado de satisfacción global de los requerimientos para dicha necesidad de información. En la Figura 4 y en la Figura 7, se pueden visualizar los términos, atributos y relaciones que forman parte del componente citado (paquete

evaluation). Para mayor información sobre este componente remitirse a (Olsina, Papa, et al., 2007).

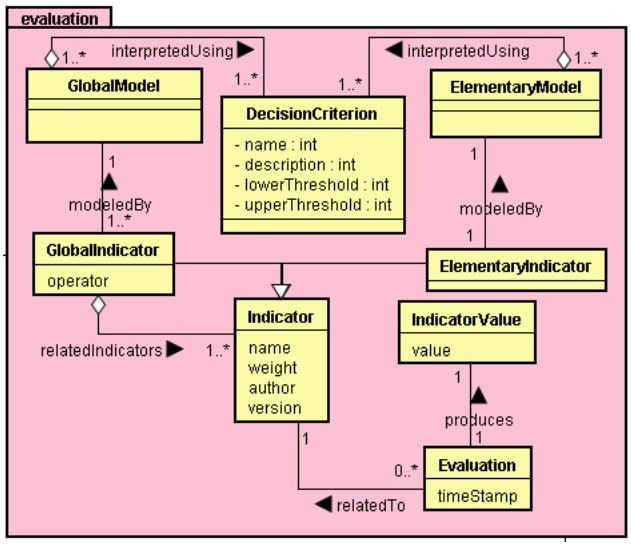


Figura 7. Principales Conceptos y Relaciones del Paquete *evaluation*

El componente *Especificación del Análisis y Recomendación* debe soportar, desde el punto de vista conceptual, un proceso en el que se diseña el análisis, se implementa el mismo y en base a sus resultados, se elaboren reportes que posibiliten luego la elaboración de las recomendaciones (Becker, Molina, et al., 2010). A la fecha, se encuentra definida la *Especificación del Análisis y Recomendación* desde el punto de vista del proceso, tal como puede observarse en la Figura 8, permaneciendo pendiente la definición de la base conceptual específica y asociado al mismo.

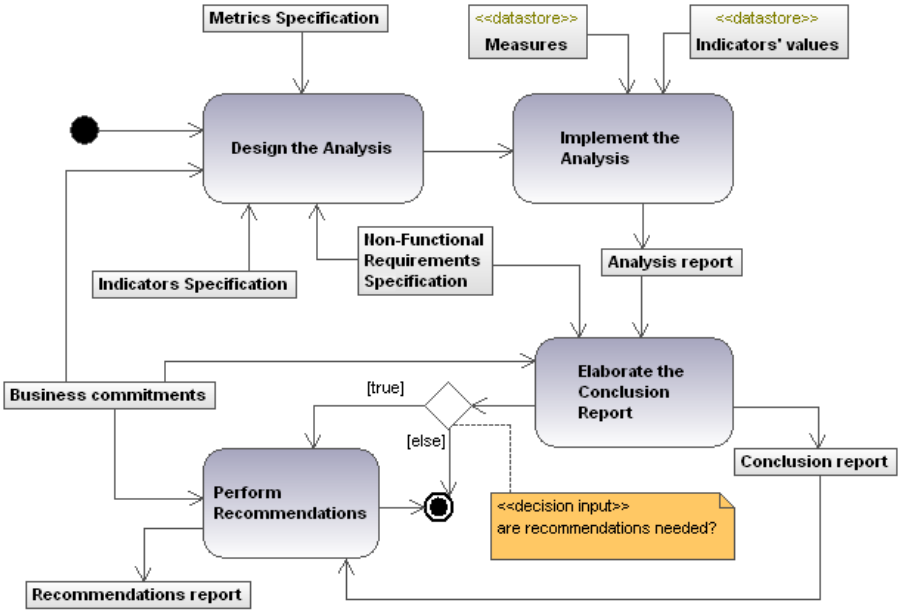


Figura 8. Sub Actividades Involucradas en la Actividad de Análisis y Recomendación (Becker, Molina, et al., 2010)

C-INCAMI tiene como fortaleza el hecho de sustentar su definición sobre una ontología diferenciando claramente los conceptos involucrados en la medición. En este último aspecto, radica el principal aporte de dicho marco para con ésta tesis, dado que debe recordarse que el principal capital del cual se pretende hacer uso, es la base conceptual sobre la que se sustenta un proceso de medición que debe garantizar consistencia y repetitividad. Es decir, nuestro requerimiento (recordar sub-sección 1.3) es que necesitamos homogenizar y estructurar los datos y conceptos provenientes desde fuentes heterogéneas, para fortalecer los aspectos asociados a la comparabilidad y consistencia en el procesamiento de los mismos.

C-INCAMI es un marco específico que permite sustentar conceptualmente, en forma explícita y consistente, a un proceso de medición. Incorpora manejo del contexto y al formalizar una base conceptual común en la definición, permite la repetitividad y comparación de los resultados que se obtuviesen a través del proceso de medición.

No obstante y tal como se discutirá en la sub-sección 4.2, la base conceptual no plantea la idea de grupo de seguimiento, es decir que una serie de métricas aplicadas a una misma entidad en un mismo contexto puedan ser agrupadas a los efectos de su análisis. Por ejemplo, en un paciente trasplantado al cual se le mide la presión arterial, temperatura axilar y frecuencia cardíaca, en tanto que en su contexto inmediato se mide a la humedad ambiente, temperatura y presión ambiental, sería altamente deseable que las mediciones sean agrupadas y analizadas conjuntamente a los efectos de verificar si existen efectos de arrastres entre las mediciones. Este último aspecto, si bien necesario para el procesamiento de flujos de mediciones, es una cuestión que se puede salvar fácilmente al proponer una extensión al componente respectivo del marco C-INCAMI.

Finalmente, de los marcos de medición analizados, el que presenta una base conceptual más robusta y consistente es C-INCAMI, por lo cual es el que hemos seleccionado. Esta elección fue fundamental para poder fortalecer los aspectos asociados a la comparabilidad y consistencia en el procesamiento de los datos y metadatos, provenientes de los flujos de mediciones desde fuentes de datos heterogéneas.

2.3 Minería de Datos Orientada al Contexto (Context-Aware Mining)

Uno de los lineamientos de la tesis, parte de que la información contextual incorporada en forma conjunta con las mediciones del objeto bajo análisis, permitiría lograr un análisis más consistente del flujo de mediciones. Esta sección, discute conceptos asociados al comportamiento predictivo aplicado en base a información contextual, como así también analiza trabajos relacionados donde se aplicó de algún modo minería de datos basada en información contextual.

"*Contexts consist of circumstantial aspects of the user and domain that may affect the data mining process. The underlying motivation is mining datasets in the presence of context factors may improve performance and efficacy of data mining as identifying the factors, which are not easily detectable with typical data mining techniques*" (Singh, Vajirkar, et al., 2003a). De la definición se extraen algunos aspectos fundamentales a considerar:

- **El contexto influye sobre los métodos de minería de datos**. La información proveniente del contexto permite brindar una semántica adicional que afectará al pre procesamiento y aplicación de las diferentes técnicas de minería de datos. Por ejemplo, no sería lo mismo intentar generar modelos de clasificación para analizar el crecimiento de larvas de polillas dentro de un laboratorio que hacerlo en su hábitat natural, inclusive aunque se intenten recrear exactamente las mismas condiciones ambientales dentro del laboratorio. Esto último, se debe a que la interacción natural del ambiente con el hábitat de las larvas, posee una dinámica compleja de reproducir en entornos controlados, como laboratorios, provocando que se termine por acotar las variables bajo estudio a los efectos de simplificar las pruebas.

- **Minar el conjunto de datos en presencia de factores contextuales puede mejorar el rendimiento y eficacia de los métodos de minería de datos**. La incorporación de factores contextuales permite incorporar nueva información al proceso de minería de datos que permite circunscribir los datos bajo estudio, a un nuevo enfoque dado por el contexto definido mediante sus factores.

- **Context-aware mining puede aplicarse a data stream pero su utilización no se encuentra limitado por éste**. Es importante comprender que la minería de datos basada en contexto busca incorporar los factores contextuales para mejorar la exactitud y agilizar el proceso de descubrimiento de información. De este modo, Context-Aware mining podría emplearse sobre data streams pero no por ello se limita a éste. Así, un repositorio de bases de datos tradicional podría ser enriquecido con datos contextuales para mejorar el proceso de Knowledge Discovery in Databases (KDD) clásico (Han & Lamber, 2001).

2.3.1 Minería de datos basada en contexto empleando ontologías

En (Singh, Vajirkar, et al., 2003a) se plantea un marco formal para context-aware data mining donde el contexto 1) será representado en una ontología, 2) será automáticamente capturado durante el proceso de minería de datos y 3) permitirá un comportamiento adaptativo a los efectos de incrementar la potencia de los métodos de minería de datos. Los factores contextuales implícitos podrían ser utilizados para interpretar y mejorar la entrada explícita del usuario y además afectar los resultados del proceso de minería de datos para entregar resultados de la predicción más relevantes y precisos. Las ontologías en este marco, proveen un medio para representar información o conocimiento que será procesable por una máquina, comunicado entre diferentes agentes y cuyos términos estarán en función del dominio bajo consideración. De este modo, se diferencia Context-aware Data Mining en dos partes:

- La representación actual de los factores del contexto para un dominio en la ontología correspondiente.

- Un marco formal genérico en el que puede consultarse esa ontología, invocar el proceso de minería de datos y coordinarlos de acuerdo al diseño de la ontología.

Se proponen diferentes tipos de contextos que incidirán en el proceso de minería:

- **Contexto del dominio**: Describe la información pertinente al contexto inmediato de la entidad bajo estudio.

- **Contexto de ubicación**: Es el conjunto de datos primarios formado a partir de la población que está situada cerca de la entidad bajo estudio.

- **Contexto de datos**: Es el conjunto de datos combinados que serán empleados para el proceso de minería. En este contexto la combinación de datos no se da desde el punto de vista estructural sino semántico.

- **Contexto de usuario**: Describe la información del usuario responsable de la consulta de datos y la historia de las consultas efectuadas por cada usuario.

El marco formal propuesto por (Singh, Vajirkar, et al., 2003a) está orientado a modelos de clasificación dentro del campo de la medicina, pudiendo generalizarse a otras áreas. El objetivo del marco es utilizar la información contextual para construir modelos de minería efectivos pero no indica cómo modelar dichos factores contextuales sino que se orienta a cómo usarlos para alcanzar el objetivo. Sin embargo, si bien identifica la entidad bajo estudio no menciona cómo las métricas se asocian a los atributos de la misma, no diferencia claramente el concepto de indicador y criterio de decisión del de métrica, sino que basa todo comportamiento en la métrica. No diferencia el concepto de métrica directa e indirecta, lo cual podría aportar detalles interesantes en la clasificación. La ausencia de los formalismos indicados, plantea cierta interrogante sobre cómo desarrolla el proceso de clasificación mismo, si es que la propia entrada de datos se ve afectada por la subjetividad.

2.3.2 Casos de Aplicación

Como se ha indicado anteriormente que con la incorporación de información contextual dentro del flujos de mediciones, bajo la hipótesis que permitiría efectuar un análisis de un modo más consistente, se discuten aquí algunos casos de aplicación a los efectos de que sirvan como antecedente del empleo de factores contextuales en procesos de minería de datos, como así también se analizan las fortalezas y debilidades de los mismos en cada una de sus áreas de aplicación.

2.3.2.1 Marco de Minería Orientado al Contexto para Aplicaciones Médicas Inalámbricas

En esta sub-sección se considera el trabajo de (Singh, Vajirkar, et al., 2003b), en donde se plantea una arquitectura de sistema para el marco formal de minería de datos orientado al contexto planeado en (Singh, Vajirkar, et al., 2003a). Este sistema es empleado en aplicaciones médicas wireless y expone cómo la arquitectura puede aplicarse a diferentes áreas como el tratamiento de la diabetes y el monitoreo de riesgos de ataques cardíacos.

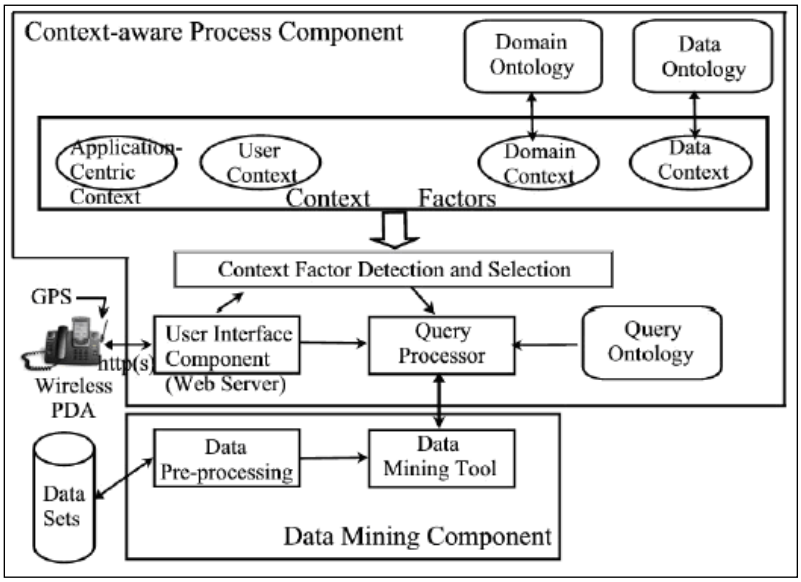


Figura 9. Arquitectura del Sistema para el marco formal de minería de datos orientado al contexto

Los componentes mostrados en la Figura 9 tienen asociados las siguientes responsabilidades:

- **Context-aware Process Component**: Administra los factores de contexto para los componentes del procesador de consulta y minería de datos. En este componente, los factores de contexto tales como el contexto de la aplicación, el contexto del usuario, el contexto del dominio y el contexto de datos son definidos tal como se indicó en 2.3.1. Con el objeto de identificar los factores del contexto a partir de la información implícita de las consultas de usuario, los autores proponen ejecutar los procesos incluyendo la verificación de los atributos de la consulta del usuario, aprendiendo de este modo sobre las entidades presentes del contexto de la consulta. Para comprender el funcionamiento del componente, el artículo plantea el siguiente ejemplo: "Heather es una paciente que sufre de un síntoma alérgico y reside en Hunstville, AL. El modelo podría analizar el grano de polen de Hunstville para el período del año y estación, e intentar detectar qué variaciones ambientales estarían afectando a la paciente. Para ello, requiere invocar procesos subsecuentes como la selección del conjunto de datos (data set) con el objeto de descubrir entidades ausentes mediante el contraste con el modelo mismo. La posterior clasificación de las entidades faltantes basada en factores contextuales, permite resolver aspectos de información contradictoria, ausente o ambigua con mínima intervención del paciente".

- **Ontology Component (Domain Ontology, Data Ontology & Query Ontology)**: Define información de metadatos de alto nivel sobre los conjuntos de datos existentes, usuarios y herramientas de minería. Por ejemplo, suponga al conocimiento almacenado en el dominio de la ontología sobre el conjunto de datos existentes, como relaciones entre conjuntos de datos a nivel estructural y también a nivel semántico. Si se emitiese una

consulta en un conjunto de datos sobre diabéticos con un valor ausente cuya variable asociada es la presión sanguínea, el valor ausente puede ser inferido desde el conjunto de datos históricos asociados a la variable.

- **Data Mining Component**: Consiste de dos subcomponentes. El primero consiste en un pre-procesamiento el cual convierte el formato del conjunto de datos existente dentro de un formato procesable por la herramienta de minería de datos. El segundo componente es la herramienta propiamente dicha, que en este caso los autores emplean el software denominado WEKA (Frank & Witten, 2005) el cual requiere un formato de archivo tipo *Attribute-Relation File Format* (ARFF) como entrada. Este subcomponente acepta el conjunto de datos pre-procesados y retorna el resultado de la consulta.

- **User Interface Component**: Tiene la totalidad de la información explícitamente dada por el usuario en el formulario de la consulta y la totalidad de la información implícita desde los componentes del contexto. El procesador de consulta puede utilizar la salida de una consulta como entrada a otra consulta y continuar de dicho modo hasta obtener la totalidad de las entradas requeridas para la consulta original que el usuario ha ingresado. En caso de que algún atributo continúe sin valor, se le solicitará al usuario que especifique su valor. De este modo los autores consideran al procesador de consultas como un gestor de meta tareas, el cual gestiona múltiples tareas atómicas que son parte de otras tareas de nivel superior. Luego de que las tareas han sido ejecutadas, el resultado final es retornado al usuario sobre http(s) a un dispositivo PDA como la predicción o resultados de la consulta.

Es un caso interesante desde el punto de vista de la aplicación de minería de datos con información del contexto, aunque presenta limitantes en tres aspectos fundamentales de lo planteado en la tesis: (a) *Se basa en un entorno de almacenamiento finito*. Note que los componentes de minería de datos requieren un archivo con formato *ARFF* de WEKA el cual se asocia a una estructura de archivo particular, de tamaño limitado, requerido por el software para su procesamiento. (b) *No contempla el procesamiento en tiempo real de los flujos*. Relacionado a lo arriba mencionado, requiere un almacenamiento y estructuración previa para proceder a su análisis, no contempla un comportamiento pro activo ante el arribo de los datos (c) Las limitaciones algorítmicas de los modelos de minería de datos sobre flujos de datos versus aquellos basados en almacenamiento finito son absolutamente diferentes, partiendo desde los requisitos de memoria hasta el tiempo de procesamiento involucrado.

2.3.2.2 Hacia una Minería de Datos Orientada al Contexto sobre el Interés del usuario en el Consumo de Documentos Multimedia

En esta sub-sección se considera el trabajo de (Stamou & Wallace, 2002) en donde se analiza la problemática de los sistema de recuperación de información (SRI), aplicados a documentos multimedia a partir de las necesidades de los usuarios. Con el objeto de obtener dicha información, los SRI estudian las acciones de los usuarios sobre un período de tiempo y

minan su contenido ad-hoc, por ejemplo a los efectos de determinar o aproximar el perfil de los mismos.

El problema de la creación automática de perfiles de usuario, según los autores, permanece bajo estudio y éste involucra la extracción de información conceptual sobre humanos de un modo completamente automatizado. El primer paso en este sentido es identificar un tema simple o grupo de temas, que haga al documento multimedia de interés para el usuario. Ejemplos de este punto si se toma como modelo las películas podrían ser datos sobre el director, el tipo de película, etc. En un segundo paso se necesita definir las medidas semánticas de similitud (o disimilitud) del documento, basado en los aspectos indicados en el primer paso. En tercer lugar se busca un camino semánticamente significativo para particionar documentos que son de interés para el usuario, dentro de grupos que corresponden a sus grupos/intereses. Finalmente, el último paso buscará encontrar los parámetros mínimos necesarios para completar la descripción del interés de un usuario. De este modo, mediante la secuencia indicada se extraen los parámetros de cada uno de los grupos de documentos formados en el tercer paso.

En resumen, los autores proponen un nuevo algoritmo de agrupamiento jerárquico orientado al contexto. Específicamente, dado un conjunto de datos (tales como documentos), entre los cuales una variedad de medidas de disimilitud se encuentran definidas, se produce un conjunto de grupos de elementos que se parecen entre ellos. La forma en la cual los elementos en cada uno de los grupos son similares entre sí, es también provista como parte de la salida del algoritmo. Es un caso interesante a los efectos de analizar cómo la información contextual contribuye en la determinación de un perfil de usuario pero adolece de las mismas limitantes indicadas para el caso en 2.3.2.1, es decir, supone almacenamiento finito, posee un comportamiento no proactivo y no contempla el aprovisionamiento continuo de datos.

2.3.2.3 Hacia un Cómputo Orientado al Contexto en Cuidados Clínicos

En esta sub-sección se considera el trabajo de (Jahnke, 2005), en donde se expone la síntesis de la arquitectura conceptual con el objeto de implementar una aplicación para cuidados médicos sensible al contexto.

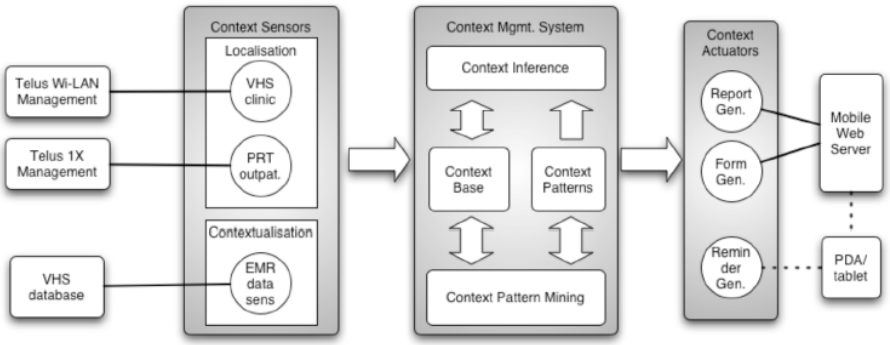


Figura 10. Arquitectura Conceptual para implementar una aplicación de cuidados médicos sensible al contexto

Las componentes de la arquitectura (ver Figura 10), poseen las siguientes responsabilidades:

- **Context Sensors**: El componente está constituido de tres elementos de software subordinados, con la tarea de importar información contextual desde la base de datos de Victoria Hospice Society (VHS) y la información sobre la localización de los usuarios móviles. Dos localizaciones diferentes de sensores son previstas para soportar el cuidado de los pacientes internados, así como el cuidado de pacientes ambulatorios. VHS tiene un equipo de respuesta paliativo y enfermeras que visitan a los pacientes ambulatorios regularmente como así también en casos de emergencia. Los dispositivos acceden dentro del hospital mediante un esquema de red Wireless 802.11g y extienden la señal mediante puntos de acceso distribuidos en el área de cobertura, que permiten la localización de los pacientes fuera de VHS. Los sensores de localización para pacientes ambulatorios simplifican el acceso a la información geográfica de la ubicación del paciente a través de los puntos de acceso y los sensores de contextualización, permiten acceder a su historia clínica, medicación y demás información relevante al contexto.

- **Context Management System**: El sistema se conforma de cuatro sub-componentes:

    - Un contexto base como un repositorio de la información contextual
    - Un conjunto de patrones contextuales representando reglas que gobiernan la entrega de información en contextos particulares
    - Un motor de inferencia contextual que empareja los patrones contextuales contra los datos contenidos en el contexto base
    - Una herramienta de minería de patrones contextuales que analiza los datos del contexto recolectados durante la utilización del dispositivo con el objeto de sacar patrones de uso.

- **Context Actuators**: El propósito de los actuadores del contexto en la infraestructura es presentar a los usuarios de dispositivos móviles con los diálogos de información correctos en el contexto correcto.

Es un caso de aplicación atractivo por cuanto expone un monitoreo de la situación contextual del paciente intentando determinar su localización por triangulación wireless, se basa en una ontología reconocida (Dolin, Alschuler, et al., 2001) para modelar la información específica del dominio del negocio y presenta un comportamiento de análisis posterior para determinar el patrón de uso de la información. No obstante, no es una arquitectura orientada en absoluto al procesamiento de flujos de datos, no presenta un comportamiento proactivo y/o detectivo y la aplicación de algoritmos asociados a las funciones de minería de datos, se efectúa sobre repositorios finitos y en forma posterior a que el hecho ha sido captado y almacenado.

2.4 Sistemas de Gestión de Flujos de Datos (Data Stream Management Systems)

Dado que uno de los puntos importantes de la tesis es procesar los flujos ante su propio arribo, la gestión involucrada en el procesamiento de los flujos de datos, es un aspecto a considerar. En este sentido, debe considerarse desde la gestión de memoria, a los efectos de garantizar el funcionamiento continuo y por ende el comportamiento detectivo y/o predictivo mencionado, hasta aspectos asociados a cómo debiera reaccionar un procesador de flujos, si la tasa de arribo de los datos supera a la tasa de procesamiento a los efectos de garantizar la estabilidad.

Desde el punto de vista clásico del procesamiento de data stream, existen trabajos relacionados tales como los llevados adelante en el proyecto STREAM de la Universidad de Stanford (Babcock, Datar, et al., 2004; Babcock, Babu, et al., 2004; Srivastava & Widom, 2004) y el proyecto Aurora, desarrollado en colaboración entre el MIT y las Universidades de Brown y Brandeis (Hwang, Balazinska, et al., 2005; Ryvkina, Maskey, et al., 2006; Tatbul & Zdonik, 2006). Básicamente, estos proyectos han abordado desde diferentes puntos de vista los tópicos y problemáticas de las operaciones tradicionales de bases de datos dentro del contexto de procesamiento de data streams.

Desde un punto de vista aplicado de los DSMS y analizando una línea de trabajo más afín a la presente tesis, como la gestión de datos en redes de sensores, se presentan distintos enfoques relacionados con sus respectivas particularidades:

- **Presto** (Li, Ganesan, et al., 2006): Presenta una arquitectura en dos capas que comprende un proxy y sensores que cooperan unos con otros para adquirir datos y procesar consultas. PRESTO construye un modelo de serie temporal basado en SARIMA (Box & Jenkins, 1991) de acuerdo a las tendencias observadas y transmite los parámetros del modelo a los sensores. Los sensores contrastan la medición con su valor predicho, de acuerdo a los parámetros informados por el proxy. En caso de que el valor sensado u obtenido se corresponda con el predicho (más o menos un umbral definido como parámetro en el sensor), el dato no es informado al proxy, mientras que en caso de desvío éste si es informado. De este modo la arquitectura pretende disminuir la sobrecarga de sincronización (overhead) en la red, por la sola transmisión de los desvíos a los valores predichos por el modelo de serie temporal y adicionalmente, ajustar el modelo en función de los nuevos desvíos. Un aspecto adicional e interesante en la resolución de consultas en PRESTO, es que el proxy en caso de que la precisión exigida en la consulta sea inferior o igual a la administrada por el mismo, puede responderla en base a sus datos locales e interpolar los faltantes mediante el modelo de series temporales, garantizando siempre la posibilidad de respuesta y evitando la consulta a los sensores. En caso de que la consulta que se le exige al proxy sea de una precisión superior a la administrada, entonces deberá solicitar los datos a los sensores mediante un mecanismo “pull” para resolver la misma.

- **Procesamiento de datos sobre data streams basados en RFID** (Diao, Liu, et al., 2009; Tranh, Sutton, et al., 2009)**:** Se plantea un sistema capaz de recibir lecturas de datos crudas o sin tratamiento previo, provenientes de diferentes lectores Radio Frecuency Identification (RFID), los cuales son objeto de inconvenientes tales como ruido, incompletitud y/o imprecisión. Estos son abordados mediante modelos probabilísticos e inferencia a los efectos de generar una descripción de dicha incertidumbre que envuelven los datos y se emplean diferentes técnicas estadísticas para capturar los cambios que va sufriendo dicha 'incertidumbre'. El sistema modela las variables como aleatorias y continuas, diferenciando entre dos tipos: observadas y ocultas. Las variables observadas son los datos de entrada que provienen desde la realidad captada mediante los lectores RFID. Como las variables observadas posiblemente incorporen imprecisiones, incompletitud y/o ruido, se modela la incertidumbre mediante funciones de probabilidad conjuntas entre las variables observadas y ocultas. Las variables ocultas representan dentro del modelo la verdadera ubicación de un objeto, con lo que en definitiva dicha función de probabilidad tratará de modelar la incertidumbre que proviene desde las variables observadas, estableciendo intervalos de confianza a los efectos de satisfacer requerimientos de precisión en la ubicación de un objeto.

- **Storage Manager of Streams (SMS)** En este artículo (Botan, Alonso, et al., 2009) se expone la necesidad de separar el procesamiento de consultas de la gestión de almacenamiento en términos de data streams a los efectos de desacoplar las componentes del DSMS. Cabe destacar que no pretende almacenar el data stream como tal, sino que por el contrario pretende brindar una alternativa independiente para lectura, actualización y control de accesos sobre cada data stream centralizando la gestión de dichas operaciones en el componente SMS y facilitando al motor de consultas, mediante una API predefinida, el acceso a cada ítem de dato en el data stream. El SMS plantea distintos parámetros arquitecturales, funcionales y relacionados con el rendimiento que le permiten adicionalmente al motor de consultas particularizarlos de acuerdo a la necesidad de lectura del data stream que desee definir.

- **DejaVu** (Lindar, Güc, et al., 2009)**:** Se trata de un sistema abocado al procesamiento de eventos, el cual integra emparejamiento de patrones sobre flujos históricos y 'vivos'. Se entiende por flujos históricos a aquellos eventos que han sido almacenados de un modo total o selectivo en un archivo físico; mientras que los flujos vivos, representan el arribo continuo de los datos mediante un mecanismo de empuje, los cuales son almacenados en memoria, actuando esencialmente como una cola. Lo interesante de este sistema en particular, es que emplea SMS (Botan, Alonso, et al., 2009) para la gestión de almacenamiento en memoria de los flujos vivos y adicionalmente, ha desarrollado una extensión al procesador de consultas MySQL agregando la implementación de una máquina de estados finitos para guiar las consultas de emparejamiento de patrones, ejecutándose ésta última como parte integral del plan de consultas MySQL.

El sistema ha sido utilizado a nivel de prototipo en seguimientos de eventos RFID en una librería. En la misma, tanto libros como personas disponían de etiquetas RFID las cuales se leían continuamente a los efectos de detectar eventos de extracción y/o depósito de libros como así también hurtos.

- **MaxStream** (Botan, Cho, et al., 2010; Tatbul, 2010)**:** Es un sistema de procesamiento de flujos federados que transparentemente integra motores de procesamiento de flujos (Stream Processing Engines - SPE) y bases de datos tradicionales. La idea central de su arquitectura es actuar como una capa intermedia de acceso homogéneo entre las aplicaciones clientes para con los SPEs y los Sistemas de Gestión de Bases de Datos tradicionales (SGBD) mediante una interface única, permitiendo la reutilización del soporte preexistente al Structured Query Language (SQL) y extendiéndolo para facilitar el empleo de consultas continuas y tratamiento de ventanas.

Las alternativas de sistemas de gestión de flujos de datos arriba descriptos presentan enfoques interesantes desde el punto de vista del procesamiento, sin embargo no plantean la incorporación de metadatos que guíen el procesamiento de los datos en sí, sino que interpretan al dato de un modo sintáctico sin considerar el significado que pudiese tener asociado. Adicionalmente, no modelan el contexto en que los datos son obtenidos, lo cual por ejemplo en RFID podría posiblemente ayudar a disminuir la incertidumbre con respecto a posibles interferencias del ambiente ante la transmisión de datos. En general, los mecanismos de integración de fuentes de datos son “propietarias”, es decir que no permitirían la integración de otra fuente de datos de modo dinámico sin una etapa de adaptación previa. El análisis detectivo no está presente debido a que en principio no se trata de sistemas cuya finalidad sea el monitoreo o control de algún ente.

Sin perjuicio de lo dicho, debe recordarse que el objetivo de este tipo de aplicaciones es la gestión del procesamiento de flujos de datos específicamente y si bien, no es exactamente el planteo fundamental de esta tesis, deben puntualizarse algunas cuestiones destacadas de los modelos mencionados, que deben tenerse en cuenta en el abordaje de esta problemática.

PRESTO incorpora al procesamiento de flujos de datos un mecanismo predictivo basado en series temporales para disminuir la sobrecarga de sincronización (overhead) de la red, mientras que la predicción es realizada por el sensor, quien decide en última instancia si transmite o no el dato. Este punto es de suma utilidad por lo que (a) delega responsabilidad funcional en el sensor y adicionalmente, (b) el sensor puede, dentro de sus restricciones computacionales, predecir efectivamente.

El procesamiento basado en RFID plantea un abordaje muy interesante del tratamiento del ruido en la transmisión de los datos, máxime desde el punto de vista estadístico, diferenciando claramente entre variables observadas (las cuales se asocian a la realidad medida y por consiguiente, adolece de ruido, incompletitud e imprecisiones) y ocultas (las que representan la verdadera ubicación del objeto y que se desean determinar).

La iniciativa en SMS de abstraer el procesamiento de consultas de la gestión de almacenamiento ha sido relevante, por cuanto permite evitar situaciones en donde la tasa de arribo supera a la de procesamiento (cuello de botella) e independizar de hecho, la operatoria de administración de recursos asociadas al flujo con respecto a la resolución de consultas que de por sí son disímiles.

El enfoque de DejaVu se destaca en cuanto a la utilización que efectúa de SMS y en lo referido a la aptitud de combinar flujos de datos en forma conjunta con datos persistentes, incorporando dicha situación en un motor de bases de datos tradicional.

Por último, MaxStream se plantea como una capa intermedia la que permite la intercomunicación entre entornos legados y flujos de datos, a la vez que permite el empleo de una extensión del bien conocido SQL, para realizar las consultas.

2.5 Árboles de Clasificación sobre Data Streams

Un aspecto a destacar de esta tesis (indicado en la sub-sección 1.2), radica en que a partir de la incorporación de fuentes de datos heterogéneas con mediciones estructuradas y enriquecidas mediante metadatos embebidos basados en C-INCAMI, se permitiera, luego de una etapa detectiva asociada con análisis estadísticos, llevar adelante la *función de clasificación* con el objeto de implementar el comportamiento predictivo. El análisis y selección de la familia de algoritmos que permiten llevar adelante la clasificación sobre flujos de datos se discute en esta sub-sección.

Desde un punto de vista de la algorítmica de clasificación sobre data stream basada en árboles, existen diferentes alternativas ya reconocidas en ámbitos industriales y/o académicos, a saber:

- **ADWIN Bagging and Adaptative-Size Hoeffding Tree (ASHT)** (Bifet, Holmes, et al., 2009): El trabajo presenta a los algoritmos ASHT y ADWIN Bagging. ASHT es un algoritmo derivado de Hoeffding Tree el cual limita su tamaño máximo e incluso durante su proceso de crecimiento, si se excediese el umbral establecido, borra selectivamente alguno de sus nodos para reajustar su tamaño. Esta última propiedad garantiza que un conjunto de data stream con altas tasas de arribo no complicarán el empleo de memoria y estabilidad del algoritmo.

  ADWIN es un estimador y detector de cambios que resuelve el problema de seguimiento de la media en un flujo de números reales. El mismo, mantiene una ventana de longitud variable de los ítems leídos recientemente en donde su longitud máxima es consistente estadísticamente con la hipótesis “No han existido cambios en la media dentro de la ventana”. ADWIN es libre de parámetros o supuestos previos en el sentido que automáticamente detecta y se adapta a la tasa actual de cambio. Su único parámetro es un límite de confianza δ, el cual indica cuan confiable deseamos que sea la salida del algoritmo, inherente a todos los algoritmos que tratan con procesos aleatorios.

Finalmente, ADWIN Bagging es el método de bagging online de (Oza & Rusell, 2001) con el agregado del algoritmo ADWIN como detector de cambios y como un estimador para los pesos o ponderaciones del método de boosting.

- **Very Fast Decision Tree Learner (VFDT)** (Domingos & Hulten, 2000): Se presenta una implementación basada en Hoeffding Tree que permite el empleo de ganancia de información o el índice de Gini como medida de evaluación de atributos. Adicionalmente, incorpora posibilidad de desempate a partir de un umbral establecido por el usuario, cuando dos atributos poseen valores muy cercanos en el criterio de división. Así, permite al usuario establecer la cantidad mínima de ítems de datos por hoja, para poder efectuar el recálculo del criterio de división. VFDT emplea una cantidad finita y estable de memoria, la cual está en función de la contabilización del crecimiento total de hojas, con posibilidad de 'desactivar' selectivamente las mismas en caso de riesgos de exceso. Complementariamente, VFDT descarta selectivamente los atributos que no aportan a la clasificación, no es necesario un gran volumen de datos para su inicialización y permite se active (o no) la relectura de datos previamente leídos, en caso de tasas de arribo lentos.

- **Very Fast Decision Tree – C Learner (VFDTc)** (Gama, Rocha, et al., 2003)**:** Expone un algoritmo basado en VFDT el cual incorpora la capacidad de tratar con datos numéricos y la posibilidad de aplicar clasificadores Naive Bayes en las hojas del árbol. Inicialmente un ítem se clasifica recorriendo el árbol desde la raíz hasta la hoja correspondiente, una vez allí el ítem se basa en las probabilidades previas de los datos posicionados en dicha hoja. Lo que ocurre, es que en situaciones de bajo volumen de datos donde se estaría inicializando el árbol, el volumen es relativamente pequeño y un clasificador bayesiano, abordaría el cálculo de la probabilidad basándose no solo en las probabilidades previas sino también en las probabilidades condicionales valor-atributo dado en la clase. Una virtud de Naive Bayes particularmente beneficiosa en esta situación, es la capacidad de desempeñarse adecuadamente en forma incremental, con datos heterogéneos y ante la presencia de ruido.

- **Very Fast Decision Tree Learner (VFDT) y Numerical Interval Pruning (NIP)** (Jin & Agrawal, 2003)**:** Se realiza una revisión del dominio del problema planteado por (Domingos & Hulten, 2000) con respecto al algoritmo VFDT en entornos de data streams y se efectúan dos aportes: el primero plantea NIP, incorporando de este modo en VFDT, la capacidad de manejar valores numéricos sin pérdida de precisión, mientras que el segundo, plantea una estrategia de muestreo sobre valores numéricos y aprovechando la propiedad de distribución normal de los valores estimados de la función de ganancia, obtiene que el tamaño de muestra requerido utilizando una prueba de normalidad siempre será menor o igual al tamaño de muestra requerido por un Hoeffding Tree.

- **Streaming Ensemble Algorithm (SEA)** (Street & Kim, 2001)**:** Este estudio aborda la función de clasificación sobre data stream a partir de un conjunto de clasificadores. La idea

subyacente es tomar subconjuntos del data stream a medida que arriban y crear un clasificador sobre cada subconjunto, de este modo los clasificadores que sobrevivirán dentro del conjunto selecto serán los de mayor precisión. Así, el algoritmo evita releer los datos arribados, mantiene un conjunto finito de clasificadores, trabaja en forma iterativa depurando los mismos, hace un uso finito de memoria y puede clasificar un nuevo valor en cualquier momento.

- **Instante-Based Ensemble Pruning** (Wang, Fan, et al., 2003)**:** Plantea la estrategia de clasificación sobre data stream basado en un conjunto de clasificadores. Al igual que SEA (Street & Kim, 2001), el hecho de disponer de un conjunto de clasificadores y que cada uno de ellos se entrene sobre un subconjunto de datos disjuntos del data stream, abre el camino para la optimización mediante procesamiento paralelo de un modo natural. Este método en particular poda su conjunto de clasificadores garantizando que el conjunto podado tendrá la misma precisión que el conjunto original, soportando fluctuación de conceptos (Concept-Drifting). Originalmente el método ordena su conjunto de clasificadores en forma descendente de acuerdo a su peso, los cuales se calculan según (Domingos, 2000; Fan, Chu, et al., 2002). Los clasificadores son consultados partiendo del que dispone mayor peso hacia el de menor peso, la consulta culmina en dos situaciones: cuando no queda clasificador por consultar o bien, cuando el subconjunto de clasificadores consultados cumple con las restricciones de confiabilidad impuesta por la consulta.

- **Concept-Adapting Very Fast Decision Tree (CVFDT)** (Hulten, Spencer, et al., 2001): El algoritmo se basa en VFDT de (Domingos & Hulten, 2000), aborda la problemática de data stream con cambios continuos, obteniendo de un modo incremental un árbol de decisión con similar exactitud a la aplicación de VFDT sobre una ventana corrediza o deslizante con la gran diferencia, de que CVFDT siempre se encuentra actualizado. De este modo, el algoritmo implementa la situación en que los datos pueden ser generados en base a más de una clase objetivo y en donde los parámetros asociados a la misma fluctúan con el tiempo.

Existen estudios de algoritmos de clasificación efectuados sobre data streams, basados en conjuntos de datos reales, entre los que se destaca (Bifet, Holmes, et al., 2009) por su vinculación al sub grupo específico SIGKDD de ACM, donde se verifica el rendimiento del procesamiento, consumo de memoria y estabilidad en el abordaje de los flujos de datos en cada una de las familias de algoritmos aquí abordadas. En base al citado estudio, puede observarse que ASHT con Bagging logró globalmente el mejor rendimiento, dado que obtuvo en general una exactitud superior a 80%, dependiendo del data set y la variación de conceptos, con un consumo de memoria que no alcanzó los 4 mega bytes (mb) en ningún momento y con tiempos de procesamiento que no alcanzaron los 1200 segundos para procesar 10 millones de instancias o mediciones.

Si bien desde el punto de vista algorítmico y sintáctico, ASHT con Bagging resulta la alternativa más ventajosa, debe indicarse que ninguno de los algoritmos planteados, sean clasificadores individuales o conjunto de clasificadores, han expuesto resultados de rendimiento y/o adaptabilidad en presencia de semántica asociada a los datos y/o en presencia de marcos formales de medición y evaluación. Esto último, requirió una simulación del enfoque integrado de procesamiento de flujos de datos basado en metadatos de mediciones, a los efectos de poder analizar estadísticamente su comportamiento global, no solo en términos de integración y procesamiento, sino también en lo referido al proceso de toma de decisión que constituye el núcleo de aplicación de ASHT con Bagging, la cual es abordada en detalle en el capítulo 9.

2.6 Necesidad de un Enfoque Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones

Cuando se trata de tomar decisiones a un nivel ingenieril, medir no es una posibilidad sino una necesidad; representa una práctica sistemática y disciplinada por la cual se puede cuantificar el estado de un ente. Si hay un aspecto que se debe tener en claro en medición, es que para comparar mediciones diferentes las mismas deben ser consistentes entre sí, esto es, deben poseer la misma escala y tipo de escala además de obtenerse bajo métodos de medición y/o reglas de cálculos equivalentes. Los marcos de medición y evaluación (discutidos en la sub-sección 2.2) representan un esfuerzo, desde la óptica de cada estrategia, por formalizar el modo de definir las métricas, sus objetivos, entre otros aspectos asociados, a los efectos de garantizar la repetitividad y consistencia en el proceso de medición que sustentan.

Balanced-Scorecard es principalmente empleado en la planificación estratégica en el área de administración, e intenta conceptualizar aspectos contextuales del plan estratégico, como la visión del cliente, pero adolece de sustento conceptual explícito a los efectos de definir el proyecto de medición y su contexto. Establece un objetivo, una métrica y un umbral. Este último, comparado contra la medición desempeñaría el rol de un indicador elemental pero ese es su límite, no permite sustentar explícitamente aspectos conceptuales del contexto de aplicación de las métricas, a los efectos de dar la posibilidad de generar una base de conocimiento en este punto, las métricas son representadas como reglas de cálculo pero no permite definir explícitamente conceptos tales como escala, tipo de escala, unidades, métodos asociados y/o herramientas para la medición. Las métricas están en función de un objetivo y no de una entidad bajo análisis, el inconveniente es que un objetivo puede tener asociado ‘n’ entidades a monitorear, con lo que no está claramente especificado qué se está midiendo, pudiéndose tornar ambigua cualquier tipo de comparación por cuanto no sabemos si las mediciones a comparar corresponden al mismo objeto. Un aspecto destacado del enfoque de Kaplan y Norton, es que reconocen un proceso iterativo en la definición de las métricas y adicionalmente, la estrategia del proceso de medición se basa en múltiples dimensiones.

GQM implica también un enfoque iterativo, el cual considera una estrategia de abordaje del proceso de medición basada en múltiples dimensiones, pero al no tener una base conceptual explícita y robusta, se torna complejo la estructuración y homogenización de los datos y sus conceptos asociados (metadatos) para su posterior procesamiento, máxime ante un entorno en el

que las fuentes de datos son heterogéneas por definición. En GQM la métrica es definida como una regla de cálculo, pero no se sabe más nada de ella, es decir, no es posible definir explícitamente conceptos tales como escala, unidad, contexto, entre otros conceptos fundamentales necesarios para sustentar la repetitividad y consistencia del proceso de medición.

C-INCAMI es un marco de medición y evaluación sustentado en una ontología, que permite de modo robusto y explícitamente definir conceptos tales como el de métricas, indicadores, contexto de medición, entidades bajo análisis, atributos asociados a las entidades sujetos de medición, entre otros conceptos que permiten sustentar la repetitividad y consistencia del proceso de medición. La base conceptual de C-INCAMI no plantea la idea de grupo de seguimiento, es decir que una serie de métricas aplicadas a una misma entidad en un mismo contexto puedan ser agrupadas a los efectos de su análisis. Este último aspecto, si bien necesario para el procesamiento de flujos de mediciones, es una cuestión que se puede salvar fácilmente al proponer una extensión al componente respectivo del marco C-INCAMI.

De igual modo, la situación en que una medición es no determinista y su resultado es en realidad una distribución de probabilidad (Abajo Martínez, 2004) no está contemplada en la base conceptual de C-INCAMI y afecta tanto a métricas como a elementos contextuales. Por lo que propondremos una extensión de la clase *dataset* en la base conceptual, para que administre valores deterministas y no deterministas (o probabilistas).

De los marcos de medición analizados, el que presenta una base conceptual más robusta y consistente es C-INCAMI, por lo cual es el que hemos seleccionado. Esta elección fue fundamental para poder fortalecer los aspectos asociados a la comparabilidad y consistencia en el procesamiento de los datos y metadatos, provenientes de los flujos de mediciones desde fuentes de datos heterogéneas.

Existen trabajos que han planteado la necesidad de definir ontologías en el marco de la minería de datos orientada al contexto, tales como el expuesto en 2.3.1. Estos enfoques intentan previamente acordar sobre los conceptos fundamentales asociados al contexto, para avanzar lógicamente sobre su aplicación a través de diferentes métodos y técnicas de minería de datos. Esto último representa un paso necesario cuando se desea incorporar semántica en los datos que se están procesando, a los efectos de enriquecer el proceso de análisis de datos. El punto inconcluso tanto en los casos de aplicación como en la ontología mencionada en 2.3.1, radica en cómo formalizar las métricas, indicadores, escalas, métodos, etc. a los efectos de tomar una medición y garantizar su repetitividad. Esto representa el talón de Aquiles de este enfoque, dado que si no existe garantía de repetitividad en el proceso de medición, de nada sirve la semántica sobre el contexto, ya que el análisis posterior sobre los datos se podrá encontrar sesgado.

En cuanto al procesamiento de flujos de datos, existen tópicos que a la fecha permanecen bajo estudio o débilmente tratados. Algunos trabajos tratan sobre esquemas de JOINING entre flujos de datos (Arasu, Ganti, et al., 2006; Arasu, Chaudhuri, et al., 2008), optimización de consultas sobre flujos de datos (Babcock & Chaudhuri, 2005; Babu, Bond, et al., 2007; Babu & Duan, 2007), lenguaje de consulta sobre flujos de datos (Arasu, Babu, et al., 2006; Toman, 2009),

entre otros. Dentro de los sistemas de gestión de flujos de datos, analizados en 2.4, ninguno se basa en marcos formales a los efectos de brindar sustento a los metadatos basados en variables numéricas, sino que cada uno plantea de un modo ad-hoc su propia semántica, lo cual afectará inevitablemente la interoperabilidad de los datos y la aplicación de los prototipos en diferentes industrias. No obstante, existen planteos muy interesantes como el de independizar el componente de gestión de almacenamiento (SMS) del motor de consultas, brindando una interface de acceso uniforme, actualmente empleada por DejaVu y Maxstream; o bien, como en el caso de PRESTO, incorporar la posibilidad de que el sensor haga empleo de un modelo predictivo SARIMA mediante una tolerancia de error predefinida, el cual es empleado para analizar en el mismo si el dato captado difiere del pronosticado más o menos la tolerancia y solo es transmitido, en caso de que el error de pronóstico supere la misma.

Los algoritmos incrementales sobre data stream basados en árbol de decisión que se han analizado en 2.5, son básicamente variantes de Hoeffding Tree. ADWIN Bagging and Adaptative-Size Hoeffding Tree (ASHT) permiten administrar la fluctuación de conceptos, variables numéricas, garantizar la estabilidad del árbol, el uso de memoria, procesamiento como así también actuar en combinación con métodos de Bagging. Very Fast Decision Tree Learner (VFDT) incorpora el índice Gini e implementa un mecanismo de desempate en el criterio de split dentro de las hojas pero no aborda el tratamiento de variables numéricas. Concept-Adapting Very Fast Decision Tree (CVFDT) se basa en VFDT, con la diferencia que aborda la problemática de data stream con cambios continuos, obteniendo el árbol de decisión con similar exactitud a la aplicación de VFDT sobre una ventana corrediza y encontrándose siempre actualizado. Very Fast Decision Tree – C Learner (VFDTc) incorpora la capacidad de tratar con datos numéricos y la posibilidad de aplicar clasificadores Naive Bayes en las hojas del árbol. Very Fast Decision Tree Learner (VFDT) y Numerical Interval Pruning (NIP) proponen la incorporación de variables numéricas junto con una estrategia de muestreo sobre las mismas, que con tamaños inferiores a los planeados en Hoeffding (o a lo sumo igual) garantizan igual precisión. Hasta aquí se han analizado algoritmos que generan un clasificador, pero existen otros enfoques que se han expuesto que plantean la posibilidad de generar un conjunto de clasificadores en post de incrementar la precisión y disminuir el error. Streaming Ensemble Algorithm (SEA) es uno de los algoritmos que caen en la última situación indicada, aborda la idea de tomar subconjuntos del data stream a medida que arriban y crear un clasificador sobre cada subconjunto, de este modo los clasificadores que sobrevivirán dentro del conjunto selecto serán los de mayor precisión. Instante-Based Ensemble Pruning se basa en un conjunto de clasificadores y dispone de un mecanismo de poda para con su conjunto de clasificadores que garantiza que el subconjunto resultante tendrá la misma precisión que el conjunto original, soportando fluctuación de conceptos. Según estudios comparativos realizados por (Bifet, Holmes, et al., 2009), el mejor equilibrio global entre capacidad de procesamiento, consumo de memoria y estabilidad es obtenido por ASHT con Bagging, por lo que éste ha sido el algoritmo seleccionado en esta tesis para implementar la función de clasificación mediante árboles incrementales. No obstante, debe indicarse que ninguno de los algoritmos planteados, sean clasificadores individuales o conjunto de clasificadores, han expuesto resultados de rendimiento y/o adaptabilidad en presencia de metadatos asociados a los datos y/o en

presencia de marcos formales de medición y evaluación. Esto último, requirió una simulación del enfoque integrado de procesamiento de flujos de datos basado en metadatos de mediciones, la cual es abordada en detalle en el capítulo 9

Concluyendo el capítulo, si bien existen intentos específicos en cada área en particular, *no existe un planteo integrado y global que permita simultáneamente*:

1. *Integrar flujos de mediciones a través de fuentes de datos heterogéneas*
2. *Permitir que el proyecto de medición se sustente completamente en un marco de medición y evaluación*
3. *Que el flujo de mediciones informado desde las fuentes de datos, incorpore metadatos basado en un marco formal de medición y evaluación, que permita dilucidar el significado de la medición*
4. *Que incorpore una fase detectiva, basada en análisis estadístico ante el propio arribo de los datos, que permita procesar los datos e incorporar un comportamiento pro activo en base a un marco formal de medición y evaluación*
5. *Que incorpore una fase predictiva ante el propio arribo de los datos, que permita anticipar situaciones de riesgo basado en información contextual, modelada formalmente a través de un marco formal de medición y evaluación*

El *Enfoque Integrado de Procesamiento de Flujos de Datos* centrado en Metadatos de Mediciones (EIPFDcMM) (Diván & Olsina, 2009a; Diván, Olsina, et al., 2011a), a discutir en el siguiente capítulo, tiene por finalidad satisfacer las necesidades indicadas anteriormente, a los efectos de poder sustentar la repetitividad y consistencia de los procesos de medición automatizados, actuar detectivamente en base a la formalización de un proyecto de medición y evaluación, como así también anticipar riesgos ante el propio arribo de las mediciones enriquecido con información contextual.

3 Panorama del Enfoque Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones

En el capítulo 2 se ha introducido conceptos claves para el procesamiento de flujos de datos centrado en metadatos de mediciones tales como data mining, data streams, mining data streams, context y context-aware. Luego, se ha analizado los marcos de medición y evaluación, fundamentales para garantizar la consistencia, repetitividad y automatización de los procesos de medición, y además, se ha discutido desde el punto de vista conceptual y de aplicación, los conceptos de minería de datos con información contextual. Por último, se analizó el estado del arte de los sistemas de gestión de flujos de datos y el asociado a árboles de clasificación sobre data streams, para concluir finalmente con la necesidad de un enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones (EIPFDcMM).

El presente capítulo presenta, en primer lugar, una visión global del EIPFDcMM, para luego avanzar sobre sus principales componentes y procesos. El orden de abordaje de éstos, es determinado por la lógica del proceso de medición, análisis y toma de decisión, por lo que a seguir se describirán los procesos de recolección y adaptación, donde podrá observarse de qué modo se puede integrar diferentes fuentes heterogéneas a los efectos de transmitir mediciones.

Luego, se continúa con el análisis de los procesos de corrección y análisis, los cuales implementan la fase detectiva del EIPFDcMM. Finalmente, se examinan los procesos de toma de decisión, los cuales implementan la etapa predictiva del modelo, concluyendo el capítulo con la presentación de un caso de aplicación (que se empleará a lo largo de la tesis) para el enfoque integrado propuesto.

3.1 Modelo Conceptual

La meta de la estrategia integrada (Diván & Olsina, 2009a) en términos de procesamiento de flujos (ver Figura 11), es la siguiente: Los datos son captados por uno o más dispositivos de medición, éstos remiten los datos sin procesar a una interface capaz de comprender el lenguaje del dispositivo e incorporar metadatos en las mediciones. Dicha interface, denominada

adaptador de mediciones (Measurement Adapter -MA), comunicará el ingreso de los datos y metadatos a una función $F^{rt}(d_{si})$ (Gathering Function -GF-), que tendrá por objeto la reunión de los flujos en función de las métricas asociadas. Los flujos reunidos, se transmiten a una función $F^{ut}(d_{si})$ (Analysis & Smoothing Function -ASF-) cuyo objetivo será suavizar las mediciones vinculadas a cada métrica. Una vez que las mediciones han sido suavizadas, se aplicará el modelo actual de clasificación $G(d^t_{si})$ (Current Classifier) en base al marco de métricas e indicadores definidos junto con el conocimiento previo almacenado, produciendo una decisión en tiempo t, a la cual se denomina D^t. Luego, el modelo actual de clasificación (Updated Classifier) se ajusta y/o sustituye incrementalmente en base a los nuevos datos y situación contextual para producir la decisión en tiempo t+1 con el nuevo modelo, permitiendo la comparación de decisiones entre D^t y D^{t+1} la cual estará a cargo del Tomador de Decisiones (Decision Maker –DM-), como así también la interpretación de los indicadores elementales definidos en función de los datos arribados y el análisis del conocimiento pre existente en función del contexto actual de la entidad bajo estudio.

Ambas decisiones permitirán proactivamente al tomador de decisiones, disparar alarmas, notificaciones, ajustes y/o lo que el usuario de C-INCAMI haya definido en términos de indicadores, a los efectos de detectar las desviaciones on-line.

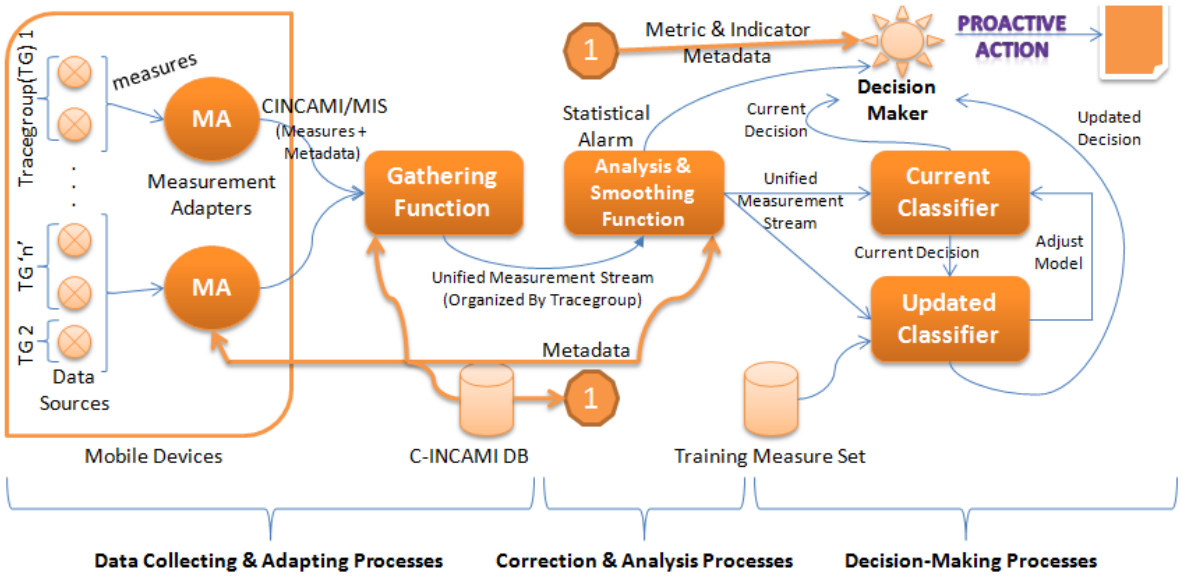


Figura 11. Esquema Conceptual del Modelo Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones

El modelo se agrupa lógicamente en procesos, los cuales a seguir se describen y se detallan en las siguientes sub-secciones del capítulo:

- **Procesos de Recolección y Adaptación**: Tiene por principal responsabilidad la recolección de mediciones. Sus funciones inician a partir de la intercomunicación con los dispositivos de medición (sensores, etc.) hasta la comunicación a los procesos de corrección y análisis de un flujo de mediciones unificado.

- **Procesos de Corrección y Análisis**: Su responsabilidad central radica en dar coherencia al orden de procesamiento de las diferentes mediciones en función de sus metadatos, como así también identificar y resolver automáticamente problemas típicos de datos tales como ruido, outliers y ausencia de valor. Sus funciones comienzan con la recepción del flujo unificado desde los procesos de recolección y adaptación, culminando con la transferencia del flujo de datos con un orden de procesamiento dado y consistente numéricamente a los procesos de toma de decisión.

- **Procesos de Toma de Decisión**: Su objetivo principal es tomar una decisión en función de los datos provistos y disparar el evento correspondiente. Sus funciones comienzan con la recepción de un flujo de datos bajo los supuestos de que el orden de procesamiento seriado en el que arriban es el adecuado y que los mismos presentan la menor cantidad posible de problemas típicos de datos, como los citados. Sus funciones finalizan al momento en que el componente que toma la decisión dispara el evento que correspondiese en caso de ser necesario.

Como puede apreciarse en el modelo conceptual del enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones (ver Figura 11), todos los procesos que abordan el procesamiento de datos tienen acceso a lo que se denomina la base de datos C-INCAMI. Esto es así, debido a que el flujo de mediciones proviene con el dato de la medición más una serie de metadatos que permiten identificar el significado del mismo como así también a la instancia de medición. El subconjunto de metadatos transmitidos con el flujo de mediciones es mínimo, por cuestiones de rendimiento (performance) de la comunicación y requieren ser extendidos con la información complementaria en la BD C-INCAMI ante su arribo, para lograr un procesamiento consistente.

3.2 Procesos de Recolección y Adaptación

Los procesos de recolección y adaptación se centran en como adecuarse a los diferentes dispositivos de medición, a los efectos de tomar las mediciones y comunicarlas luego a los procesos de corrección y análisis. Los principales componentes de la recolección y adaptación son los flujos de datos propiamente dichos, el adaptador de mediciones y la denominada función de reunión.

Sintéticamente, las mediciones se generan en las fuentes de datos heterogéneas, las cuales abastecen a un módulo denominado *adaptador de mediciones* (*MA en* Figura 12) generalmente embebido en dispositivos móviles por una cuestión de portabilidad y practicidad, aunque podría embeberse en cualquier dispositivo de cómputo con asociación a fuentes de datos. MA incorpora junto a los valores medidos, los metadatos del proyecto de medición y los informa a una *función de reunión central* (*Gathering Function –GF*). GF incorpora los flujos de mediciones en un buffer organizado por grupos de seguimiento –modo dinámico de agrupar a las fuentes de datos definido por el director del proyecto de M&E- con el objeto de permitir análisis estadísticos consistentes a nivel de grupo de seguimiento o bien por región geográfica donde se localicen las fuentes de datos, sin que ello implique una carga adicional de procesamiento. Adicionalmente, GF

incorpora técnicas de *load shedding* (Rundensteiner, Mani, et al., 2008) que permiten gestionar la cola de servicios asociada a las mediciones, mitigando los riesgos de desborde independientemente el modo en que se agrupen.

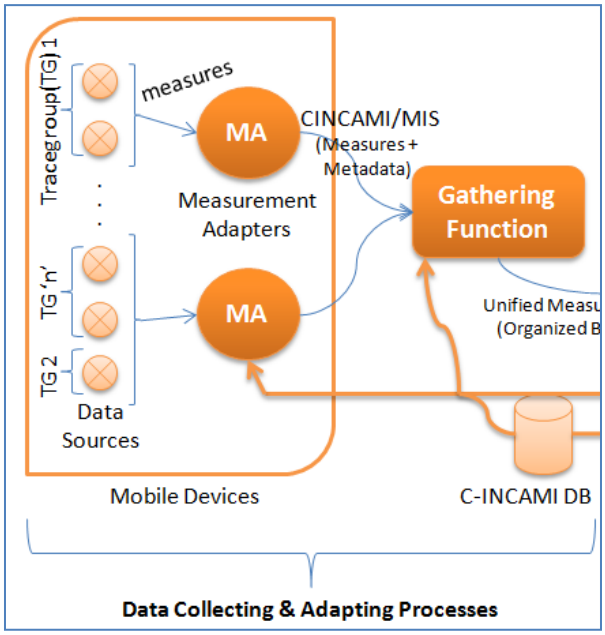


Figura 12. Procesos de Recolección y Adaptación

Las fuentes de datos solo transmiten datos, es decir, no tienen ningún tipo de gestión de metadatos (ver Figura 12), de modo que ésta representa a cualquier dispositivo capaz de exportar su medición e informarla al componente denominado Adaptador de Mediciones (Measurement Adapter -MA). Dicha iniciativa tiene como principal objetivo, independizarse de la tecnología de implementación de los dispositivos de medición, permitiendo a cualquier dispositivo transmitir sus mediciones a través de una interface común definida en MA. De este modo, la fuente de datos más allá de cómo llevará adelante el proceso de medición para el cual fue diseñado, no tiene más que implementar la interface común de comunicación para con el MA si necesitara formar parte del enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones.

El adaptador de mediciones tiene además una responsabilidad más compleja en términos funcionales con respecto a la fuente de datos. Inicialmente debe mantener registro de qué fuentes de datos han solicitado transmitir mediciones a través de éste y a qué entidad se asocian (bajo algún proyecto de medición dado). Esto último impone una etapa de configuración del adaptador de mediciones para con la fuente de datos, en donde el MA buscará asociar la fuente de datos con la métrica o métricas asociadas a algún atributo en particular de una entidad dentro de un proyecto de medición. Tal configuración, supone una consulta a un repositorio donde esté claramente definido los diferentes componentes del proyecto de medición y es allí donde, desde el MA, se consulta la base de datos C-INCAMI. Adicionalmente, el MA puede actualizar el repositorio de metadatos C-INCAMI con información tal como los tipo de dispositivos que implementan la métrica, último dispositivo asociado, si la métrica está siendo efectivamente

implementada en un momento de tiempo dado, entre otros parámetros que permiten enriquecer el monitoreo del proyecto de medición.

A partir de la asociación entre flujos de datos y métricas que se implementa, el MA *conoce* cuáles son sus metadatos a los efectos de incorporarlos en el flujo de mediciones. El MA debiera ser lo más ágil posible en términos de procesamiento y empleo de memoria, ya que debe residir en las proximidades de la localización de los dispositivos de medición y esto en general, impone límites de recursos propio de los dispositivos móviles. Es por ello, que el MA debe incorporar solo los metadatos básicos y necesarios en los datos a comunicar y transmitir finalmente cuando sea necesario. Así, toma particular importancia la idea planteada en los sensores de PRESTO (Li, Ganesan, et al., 2006), donde un modelo de serie temporal basado en SARIMA (Box & Jenkins, 1991) es construido de acuerdo a las tendencias observadas, transmitiéndose a los sensores los parámetros del modelo a los efectos de que determinen si la medición es acorde a la predicción con el fin de informar solo las desviaciones. Siguiendo con esta última línea de razonamiento, se sabe que tiene un costo computacional estimar los parámetros de un modelo SARIMA, pero dada la información observada para una serie temporal, limitada y de historia reciente de datos, almacenada dentro de la base de datos C-INCAMI, es posible disponer de parámetros iniciales e informarlos al MA para que efectúe la predicción. De este modo y al igual que PRESTO, el MA informará desviaciones a la predicción ante la presencia de parámetros del modelo SARIMA, lo que permitirá optimizar el empleo de recursos como anchos de banda, disminuir el overhead de la comunicación, adecuar tiempos de procesamiento, entre otros. En caso de no disponer el MA de los parámetros del modelo SARIMA, transmitiría la totalidad de las mediciones bajo la modalidad de ráfagas, en función de la configuración y limitación de la memoria local del dispositivo que ejecuta el MA.

De lo mencionado, se deduce por un lado que la estimación de los parámetros SARIMA no es responsabilidad del MA, sino que en realidad caen bajo las obligaciones de los procesos de toma de decisión, los cuales deberán informar los mismos mediante el proceso de inicialización de un MA. Por otro lado, una fuente de datos solo puede asociarse con un MA; esto es así para evitar la duplicación de datos y una doble predicción sobre las mediciones en MA diferentes. Finalmente, un MA puede recepcionar tantos parámetros SARIMA como métricas diferentes sean implementadas por las diferentes fuentes de datos, ya que cada serie de mediciones asociada a una métrica puede entenderse como una serie temporal en sí misma.

Una vez que el MA ha calculado sus predicciones y determina qué información deberá transmitir a la función de reunión (Gathering Function -GF-), viene el aspecto de con qué estrategia transmitir. Esto último implica decidir sobre si gestionará un buffer local o bien, transmitirá sobre cada desviación detectada. Ambas posibilidades pueden ser implementadas por el componente, bajo la premisa de mantener estabilidad del sistema, uso limitado de memoria y capacidad de procesamiento, teniendo en cuenta que posiblemente se encontrará sobre una plataforma de dispositivos móviles aunque no está restringido a ellos.

La función de reunión recepcionará los datos y metadatos desde el adaptador de mediciones bajo un esquema CINCAMI/MIS (Diván, Molina, et al., 2008) y podrá agrupar las mismas si el objeto de medición fuese equivalente. Esto último implica que si una métrica aplicada al mismo atributo de entidad bajo un proyecto dado, es implementada por dos fuentes de datos diferentes, al arribar los datos, la función de reunión deberá agruparlas bajo un mismo flujo a los efectos del análisis estadístico posterior, dado que se trata en definitiva de dos dispositivos físicos que implementan igual objetivo de medición. Es importante mencionar que la función de reunión puede efectuar tal agrupamiento de flujos, gracias a que es capaz de consultar la base de datos C-INCAMI donde residen los metadatos del proyecto de medición. Esta situación particular de la función de reunión, abre la posibilidad a que se aborde un entorno de procesamiento paralelo, dado la independencia de los flujos de datos y la agilidad de fusionamiento que se dispondría en caso de flujos vinculados, como la situación mencionada en que se implementen igual métrica en diferentes dispositivos de medición.

3.3 Procesos de Corrección y Análisis

Sintéticamente, y recordando a la Figura 11, una vez que las mediciones se encuentran organizadas en el buffer de la función de reunión (Gathering Function), se aplica mediante arrastre de los datos desde el buffer *análisis descriptivo, de correlación y componentes principales* (*Analysis & Smoothing Function –ASF-*) guiados por sus propios metadatos, a los efectos de detectar situaciones inconsistentes con respecto a su definición formal, tendencias, correlaciones y/o identificar las componentes del sistema que más aportan en términos de variabilidad. De detectarse alguna situación en ASF, se dispara una alarma estadística al *tomador de decisiones* (*Decision Maker -DM*) para que evalúe si corresponde o no disparar la alarma externa (vía, e-mail, SMS, etc) que informe al personal responsable de monitoreo sobre la situación.

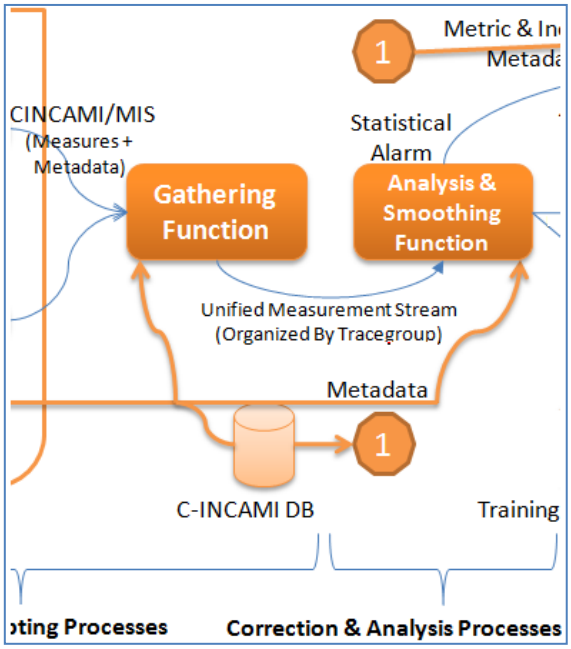


Figura 13. Procesos de Corrección y Análisis

El proceso de corrección se basa en un abordaje estadístico de los datos que se sustenta en la presencia de metadatos, esto permite enriquecer el análisis mediante el conocimiento previo de la semántica del mismo.

Adicionalmente, el proceso abordará y determinará el orden coherente de procesamiento, el cual es consensuado mediante la prioridad que se haya fijado para cada una de las métricas dentro del proyecto de medición. Esto último incorpora la idea de que no todas las métricas dentro del proyecto poseen igual importancia, en efecto, existe en cualquier circunstancia de medición aspectos más relevantes y por ende más prioritarios que otros. De este modo, este proceso permite abordar selectivamente, previo a aplicar técnicas estadísticas, el tratamiento de los datos prioritarios.

Una vez que se ha producido el reordenamiento de los flujos de mediciones dado por la prioridad de las métricas, la cual es obtenida mediante consulta a la base de datos C-INCAMI (ver Figura 13), se aplican distintas técnicas estadísticas (Análisis Descriptivo, Análisis de Componentes Principales y Análisis de Correlación) para abordar problemáticas de ruidos, outliers y valores faltantes. Claro está, que dichas técnicas requerirán que se configuren las acciones a tomar o parámetros previos tales como confianza requerida dentro del proyecto de medición, para que el análisis sea en verdad automatizado y coherente.

Culminado el tratamiento de los datos de acuerdo a los parámetros del proyecto de medición, se informa el flujo al proceso de toma de decisión y en particular a dos componentes: Al modelo de clasificación y al tomador de decisión. El modelo de clasificación requerirá los datos para tomar una decisión (Current Classifier en Figura 11) y posteriormente actualizar su modelo, mientras que el tomador de decisión (Decision Maker -DM- en Figura 11) se abastecerá de los datos bases más las decisiones de los modelos de clasificación para sustentar su accionar.

3.4 Procesos de Toma de Decisión

Una vez que los nuevos flujos de mediciones son comunicados al *clasificador vigente* (*Current Classifier* –CC- en Figura 11) desde la función de reunión (Gathering Function -GF-), éste deberá clasificar las nuevas mediciones si corresponden o no a una situación de riesgo e informar dicha decisión al DM. Simultáneamente, se reconstruye el CC incorporando las nuevas mediciones al conjunto de entrenamiento y produciendo con ellas un *nuevo modelo* (*Updated Classifier –UC- en* Figura 11). El UC clasificará las nuevas mediciones y producirá una decisión actualizada que también será comunicada al DM. El DM determinará si las decisiones indicadas por los clasificadores (CC y UC) corresponden a una situación de riesgo y en cuyo caso con qué probabilidad de ocurrencia, actuando en consecuencia según lo definido en el umbral mínimo de probabilidad de ocurrencia definido en el indicador, por el director del proyecto. Finalmente, independientemente de las decisiones adoptadas, el UC se torna en CC sustituyendo al anterior, en la medida que exista una mejora en su capacidad de clasificación según el modelo de ajuste basado en curvas ROC (*Receiver Operating Characteristic*) (Duin, Tortorella, et al., 2008).

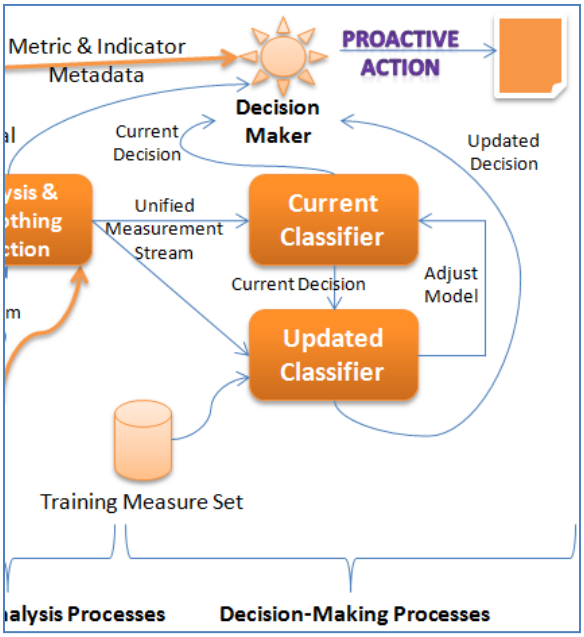


Figura 14. Procesos de Toma de Decisión

El proceso de toma de decisión deberá tomar los flujos de datos, bajo el supuesto de que los mismos han sido previamente tratados en términos estadísticos y presentan la menor cantidad posible de inconvenientes en términos de ruidos, outliers y valores faltantes. No obstante, aunque minimizada la problemática debe considerarse el margen de error propio de la parametría del proyecto de medición y por ende, la posibilidad de que persista rastro de incongruencias de datos no detectados y a los cuales deberán hacer frente los clasificadores al momento de procesar los datos.

El proceso de toma de decisión, como puede apreciarse en la Figura 14, se compone de una BD de entrenamiento, un clasificador actual, un clasificador actualizado y el tomador de decisión. La BD de entrenamiento almacena en forma estructurada los parametros y datos con la historia reciente, que son empleados para la estimación de los parámetros de SARIMA a los efectos de inicializar los MA, como así también el subconjunto de mediciones de entrenamiento basado en experto, de los casos de riesgo y normalidad para la entidad bajo medición, con los cuales se entrena a los clasificadores durante su inicialización.

Se dispone de dos clasificadores incrementales basados en árbol, uno clasifica sin actualizar el modelo conforme al dato recientemente arribado, denominado $G(d^t_{si})$ (Current Classifier en Figura 14), mientras que el segundo actualiza el modelo con el dato recientemente arribado y con el nuevo modelo, denominado $G'(d^t_{si})$(Updated Classifier en Figura 14), clasifica el dato recientemente arribado. Ambas clasificaciones, se informan en forma conjunta con el flujo de datos proveniente del proceso de corrección y análisis al tomador de decisiones quien, en base a la información de los indicadores definida en el proyecto de medición, tomará las acciones pertinentes. El clasificador $G(d^t_{si})$ es reemplazado por $G'(d^t_{si})$, de modo que ante un nuevo arribo

de datos, el clasificador que en un tiempo t+1 se consideró actualizado, en el tiempo t+2 será el actual, $G(d^t_{si})$.

3.5 Caso de Aplicación

El caso de aplicación que se presenta en esta sub-sección tiene por objetivo ilustrar el EIPFDcMM (Diván & Olsina, 2009a). La idea subyacente es que los médicos del centro de salud puedan evitar reacciones adversas y daños mayores en la salud del paciente (trasplantado ambulatorio) en la medida en que puedan disponer de un seguimiento continuo del mismo. Es decir, que puedan disponer de un mecanismo por el cual les informe ante variaciones no previstas y/o inconsistencias en indicadores de salud definidos por ellos (como expertos) para un tipo de trasplante realizado en particular. En definitiva, la idea central es que exista proactivamente algún mecanismo que, basado en las métricas e indicadores de salud definidas por los especialistas para un tipo de trasplante (y potencialmente para segmentos de edades), informe sobre situaciones que pudieran afectar la salud del paciente bajo monitoreo.

A los efectos de ejemplificar los factores contextuales a evaluar como así también los atributos de los cuales se desea mantener información permanente del paciente, los especialistas definen el siguiente proyecto de medición en base al marco C-INCAMI, presentado en la sub sección 2.2.3.

La necesidad de información es "*monitorear los principales signos vitales en un paciente trasplantado al momento en que se le da el alta desde el centro médico*". La entidad bajo análisis es el *paciente trasplantado ambulatorio* (notar que podría definirse entes más específicos como *paciente trasplantado anciano*, entre otros, con el fin de ser más precisos en la definición e interpretación de indicadores). Según los expertos, la *temperatura corporal*, la *presión arterial sistólica* (máxima), la *presión arterial diastólica* (mínima) y la *frecuencia cardiaca* representan los atributos de los signos vitales relevantes a monitorear en este tipo de paciente. Además, los expertos señalan que es necesario monitorear la *temperatura ambiental*, la *presión ambiental*, la *humedad* y la *posición del paciente* (latitud y longitud) como parte de las propiedades de contexto. La necesidad de información junto con la definición de la entidad, sus atributos y contexto, forman parte de la "*Definición y Especificación de Requerimientos no Funcionales*" y de la "*Definición del Contexto del Proyecto*" (ver sub sección 2.2.3).

La cuantificación de los atributos se realiza por medio de las métricas conforme al componente *Diseño y Ejecución de la Medición*. Para el monitoreo, se desea disponer de las métricas que cuantifiquen a los atributos citados, a saber: la presión arterial sistólica, presión arterial diastólica, temperatura corporal y frecuencia cardiaca.

Con el objeto de ejemplificar cómo se definen los metadatos vinculados a las métricas según C-INCAMI, se especifica en la Tabla 2, la definición de métrica asociada al atributo temperatura corporal.

Atributo	Temperatura Corporal
Métrica	Valor de la temperatura axilar
Código Id.	VTA
Tipo de Métrica	Directa
Escala	**Tipo de Escala** — **Intervalo** **Dominio de Valores** — **Numérica, Continua, Real^{+}** **Unidad** — **Grados Centígrados (^{o}C)**
Método	**Nombre** — **Axilar** **Método de Medición** — **Objetivo** **Especificación** — **Embebida en el instrumento** **Instrumento** Nombre: Omron ECO-TEMP Versión: ECO-TEMP Proveedor: Omron Descripción: Termómetro electrónico digital para uso oral, rectal o axilar

Tabla 2. Definición de la Métrica Valor de la Temperatura Axilar

En cuanto a las propiedades de contexto, se desea disponer de un monitoreo sobre la temperatura ambiental, la presión ambiental, la humedad y la posición del paciente (latitud y longitud). La Tabla 3 especifica la definición para una de ellas: La Temperatura Ambiente.

Context ID	CtxPacienteTrasplantadoAmbulatorio
Propiedad de Contexto	**Temperatura Ambiente**
Métrica	Valor de la temperatura ambiental del paciente trasplantado
Código Id.	VTAPT
Tipo Métrica	Directa
Escala	**Tipo de Escala** — **Intervalo** **Dominio de Valores** — **Numérica, Continua, Real^{+}** **Unidad** — **Grados Centígrados (^{o}C)**
Método	**Nombre** — **Sensado de temperatura** **Método de Medición** — **Objetivo** **Especificación** — **Embebida en el instrumento** Instrumento **Nombre** — **Termómetro Digital** **Versión** — **TCH305003** **Proveedor** — **Technidea S.A.** **Descripción** — **Dispositivo que permite medir la temperatura interna, externa y humedad relativa al ambiente.** **Rangos:** • **Temperatura 10^{o} a +60^{o}** • **Humedad 10% a 99%**

Tabla 3. Definición de la Propiedad de Contexto Temperatura Ambiente

Para el escenario actual, los expertos ya han consensuado el conjunto de métricas y propiedades contextuales a monitorear. Ahora, deben establecer los indicadores elementales, a los efectos de sentar la base para la interpretación de los atributos y conceptos calculables. De este modo, han definido los siguientes indicadores elementales: el nivel de temperatura corporal, el nivel de presión, el nivel de la frecuencia cardiaca y el nivel de diferencia de la temperatura

corporal versus la temperatura ambiental. Estos indicadores integran el componente *Diseño y Ejecución de la Evaluación* (ver sub-sección 2.2.3) y se ejemplificó en la Tabla 1 en la sub-sección 1.2, algunos de los metadatos asociados a la definición del indicador elemental "*Nivel de Temperatura Corporal*".

Una vez establecida la necesidad de información, el ente a monitorear, los atributos a medir, las propiedades contextuales, las métricas que los cuantifican y los indicadores elementales que los interpretan conforme a los criterios dado por los expertos, se estará en condiciones de instalar y configurar el MA en un dispositivo móvil –el del paciente-, el cual trabajará en forma conjunta con los sensores tal y como se expone en la Figura 15.

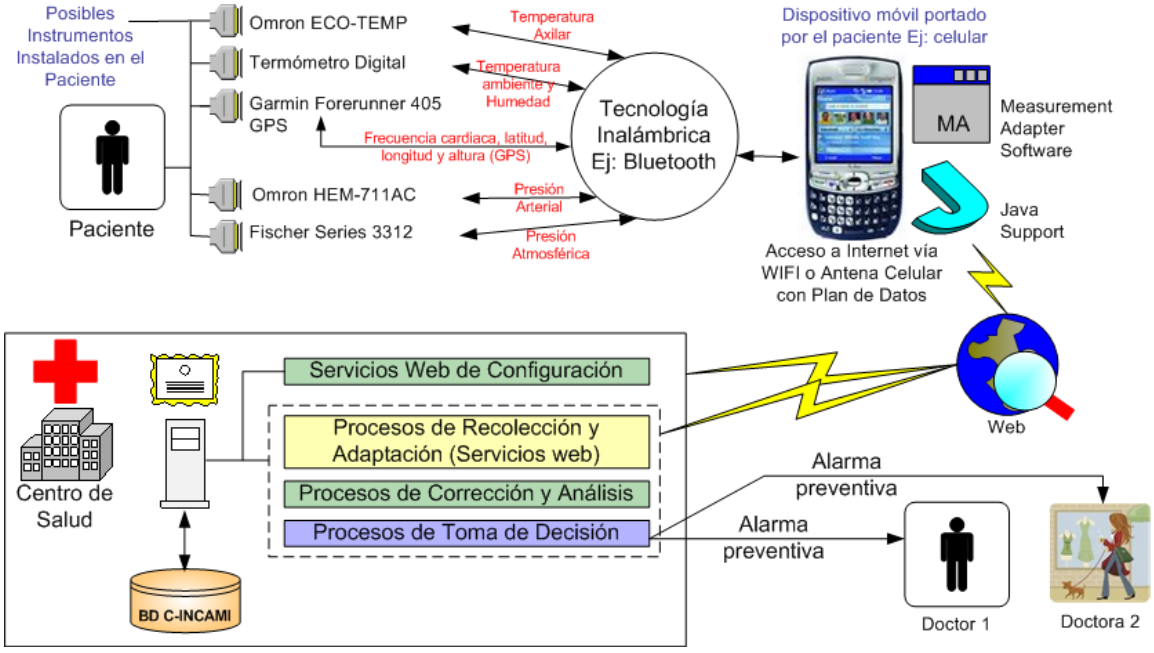


Figura 15. Esquema de Aplicación del Enfoque Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones a Pacientes Ambulatorios Trasplantados

El MA tomará las mediciones desde los sensores (fuentes de datos) e incorporará los metadatos vinculados a las métricas y propiedades de contexto que se relacionaron con éstos. Siguiendo con el ejemplo, incorpora el identificador de la propiedad de contexto temperatura ambiente (VTAPT, ver Tabla 3) en forma conjunta con el valor del dato a transmitir, supongamos 37^{o}C para la temperatura ambiente; y así para cada atributo y propiedad de contexto. Datos y metadatos se transmiten mediante el esquema C-INCAMI/MIS (Diván, Molina, et al., 2008), el cual será profundizado en el siguiente capítulo, a la función de reunión (Gathering Function -GF-, ver Figura 12). Como puede visualizarse en la Figura 15, La función de reunión es el único componente de los procesos de recolección y adaptación que se sitúa en el centro de salud en forma conjunta con los procesos de corrección y análisis y toma de decisión.

Cuando la función de reunión recibe las mediciones desde los diferentes pacientes bajo monitoreo, ordena a las mismas por paciente (Grupo de Seguimiento) y las transmite a los

procesos de corrección y análisis los que, principalmente, tratarán de resolver problemáticas propias de los datos tales como los valores faltantes, ruidos, etc. En este sentido y gracias a los metadatos, si se recepcionara por ejemplo para la métrica "Valor de la Temperatura Axilar" un valor 0, por la misma definición de la métrica, el modelo de procesamiento identifica automáticamente un error en el dato, dado que el tipo de escala es intervalo y su dominio de valores es Numérico, Continuo y Real+.

Si bien desde el paciente arriban y se analizan en simultáneo todos los valores de las métricas y propiedades de contexto configuradas por los expertos, por un momento considere que sólo se están recibiendo datos de la temperatura axilar y la temperatura ambiental desde el paciente y que es factible visualizarlo. De este modo, si observa la Figura 16, podrá apreciar el límite inferior y superior definido para el indicador "*Nivel de la Temperatura Corporal*" junto con la evolución de la *temperatura ambiental* (serie indicada con línea continua) y la *temperatura axilar* (serie indicada con línea punteada).

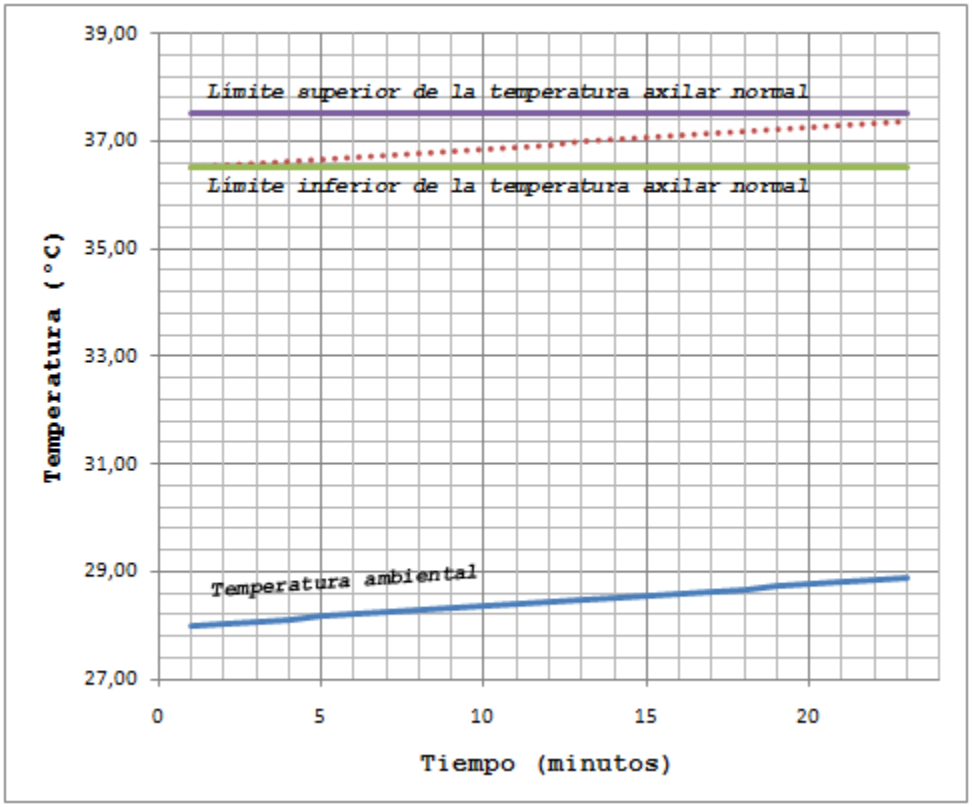
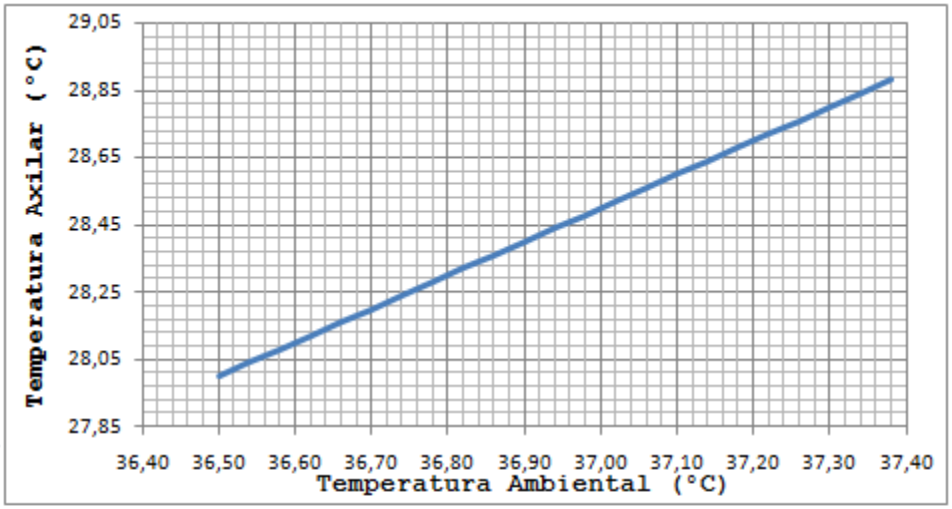


Figura 16. Visualización de las Mediciones para la Temperatura Axilar y la Temperatura Ambiental

Los datos de las medidas, e incluso el nivel de aceptabilidad del indicador elemental mostrados en la Figura 16, arrojarían que el paciente se encuentra en una situación normal. No obstante, los procesos de toma de decisión (ver Figura 14 en la sub-sección 3.4) además de analizar los indicadores en sí mismos, analizan en forma conjunta la interacción entre indicadores, métricas y factores contextuales lo que permite detectar una situación como la expuesta por la Figura 17 a).



(a)

Coeficientes de Correlación

Correlación de Pearson: coeficientes \ probabilidades

	AxillaryTemp	EnvironmentalTemp
AxillaryTemp	1,00	0,00
EnvironmentalTemp	1,00	1,00

(b)

Figura 17. (a) Análisis de Correlación de la Temperatura Axilar versus la Temperatura Ambiental (b) Matriz de Correlación

Lo que parecía normal y evidente, posiblemente no lo era tanto, dado que el modelo detectó una correlación, como puede apreciarse en la Figura 17 b), entre la temperatura axilar y la temperatura ambiental. Esto determinará posiblemente una emisión de alarma preventiva desde el centro de salud a los médicos, ya que el incremento de la temperatura ambiental arrastra a la corporal y esta situación puede implicar un incremento gradual del riesgo para el paciente.

4 Fundamentos de C-INCAMI y su Adaptación al Procesamiento de Flujos de Mediciones Centrado en Metadatos

El capítulo 3 ha permitido obtener una visión global del enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones. Ha expuesto claramente los objetivos de los procesos de recolección y adaptación, el rol de los procesos de suavización y análisis para con la fase detectiva, como así también la relación entre los procesos de toma de decisión y la fase predictiva.

En este capítulo, se retoma la discusión de C-INCAMI efectuada en la sub-sección 2.2.3, para profundizar en primer lugar, los conceptos fundamentales asociados a un proyecto de medición y evaluación. A seguir, se analizan los principales aspectos de C-INCAMI a considerar para el intercambio de datos conjuntamente con metadatos, a los efectos de proponer las modificaciones y extensiones conceptuales necesarias por el enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones (EIPFDcMM). Dentro de las modificaciones conceptuales, se analiza la necesidad de un grupo de seguimiento, aspectos asociados con la precisión y tolerancia de error, como así también lo referido al concepto de unidad lógica de medición y la base de entrenamiento requerida, para la implementación de los aspectos predictivos en los procesos de toma de decisiones.

4.1 Conceptos Fundamentales

Como se ha expuesto en la sub-sección 2.2.3, C-INCAMI es un marco conceptual que define los módulos y conceptos que intervienen en el área de medición y evaluación, para organizaciones de software. Se basa en un enfoque en el cual la especificación de requerimientos, la medición y evaluación de entidades y la posterior interpretación de los resultados están orientadas a satisfacer una necesidad de información particular. Está integrado por los siguientes componentes principales: *Definición y Especificación de Requerimientos no Funcionales*, *Especificación del Contexto del Proyecto*, *Diseño y Ejecución de la Medición* y *Diseño y Ejecución de la Evaluación.*

Esta sección definirá los principales conceptos, tomados de los trabajos de (Olsina, Papa, et al., 2007; Becker & Olsina, 2010).

Los mismos, representan el sustento ontológico para llevar adelante cualquier tipo de intercambio de datos/metadatos dentro del EIPFDcMM y se exponen, organizados del siguiente modo: *Términos asociados al proyecto, términos asociados a requerimientos no funcionales, términos asociados al contexto, términos asociados a la medición y términos asociados a la evaluación*.

4.1.1 Términos asociados al proyecto

Los conceptos asociados al proyecto, se relacionan principalmente con la definición del tipo de proyecto a llevar adelante. Dentro de los principales se encuentran:

- ***Project:*** Es un esfuerzo temporal planificado, el cual adopta la especificación de restricciones de recursos y actividades que se ejecutan para lograr un objetivo particular.

- ***Requirement Project:*** Es un proyecto que permite la especificación de requerimientos no funcionales para actividades de medición y evaluación.

- ***Measurement Project***: Es un proyecto que permite, a partir de un proyecto de requerimiento, asignar métricas a atributos y registrar los valores en un proceso de medición.

- ***Evaluation Project***: Es aquel proyecto que permite, a partir de un proyecto de medición y un modelo de conceptos de un proyecto de requerimiento, asignar indicadores y ejecutar los cálculos en un proceso de evaluación.

- ***MEProject***: Es un proyecto que integra en forma relacionada proyectos de requerimiento, medición y evaluación permitiendo mantener la trazabilidad o seguimiento de la totalidad de los datos y metadatos vinculados.

4.1.2 Términos asociados a requerimientos no funcionales

Se corresponde con los conceptos asociados a distintos elementos que permiten definir el alcance del proyecto en base a los requerimientos no funcionales. Dentro de los principales se encuentran (ver Figura 4):

- ***Information Need:*** Es la visión necesaria para gestionar objetivos, metas, riesgos y problemas.

- ***Entity Category:*** Es la categoría de objetos que será caracterizada mediante la medición de sus atributos.

- ***Entity (synonym: Object):*** Es un objeto concreto que pertenece a una categoría de entidad.

- ***Attribute (synonyms: Property, Feature):*** Es una propiedad abstracta o física mensurable de una categoría de entidad.

- ***Calculable Concept (synonym: Measurable Concept in (ISO 2002), also known as Characteristic and Sub-characteristic):*** Es una relación abstracta entre atributos de entidades y necesidades de información.

- ***Concept Model (synonyms: Factor , Feature Model):*** Es el conjunto de sub-conceptos y las relaciones entre ellos, los cuales proveen las bases para especificar el requerimiento conceptual y su estimación o evaluación adicional.

- ***Requirement Tree:*** Es una restricción al tipo de relaciones entre los elementos del modelo de conceptos, considerando la teoría de grafos.

4.1.3 Términos asociados al contexto

Son términos que se vinculan con la definición de conceptos asociados a aspectos que circundan el objeto bajo análisis. Dentro de los principales se encuentran (ver Figura 4):

- ***Context:*** Es un tipo especial de entidad que representa el estado de la situación de una entidad, el cual es relevante para una necesidad de información en particular. La situación de una entidad involucra las tareas, los propósitos de las tareas y la interacción de la entidad con otras entidades en cuanto a tareas y propósitos.

- ***Context Property (synonyms: Context Attribute, Feature):*** Es un atributo que describe una característica del contexto de una entidad dada y se asocia a una de las entidades que participa en el contexto descrito.

- ***Contextual Entity:*** Es una entidad cuyo uso y/o interpretación es sensible al contexto objetivo en el que es aplicado y/o considerado.

4.1.4 Términos asociados a la medición

Corresponden a términos que **s**e encuentran relacionados con aquellos conceptos que permiten formalizar el modo, la forma y los mecanismos de cuantificación de una medición. Dentro de los principales se encuentran (ver Figura 4):

- ***Unit:*** Es una cantidad particular definida y adoptada por convención, con el cual otras cantidades del mismo tipo son comparadas a fin de expresar su magnitud relativa a dicha cantidad.

- ***Scale:*** Es un conjunto de valores con propiedades definidas. El tipo de escala depende de la naturaleza de la relación entre los valores de la escala. Los tipos de escala más utilizados en Ingeniería del Software son clasificados en nominal, ordinal, intervalo, ratio y absoluta.

- ***Calculation Method:*** Una secuencia lógica particular de operaciones especificadas, con vistas a permitir la realización de la descripción de una fórmula o indicador por medio de un cómputo.

- ***Metric:*** Es el método de cálculo o medición definido junto con su escala de medición.

- ***Measure:*** Es el número o categoría asignada a un atributo de una entidad mediante la realización de una medición.

- ***Measurement:*** Es una actividad que utiliza la definición de la métrica con el objeto de producir un valor de la medida.

- ***Direct Metric (synonyms: Base, Single Metric):*** Es una métrica de un atributo que no depende de otra métrica de cualquier otro atributo.

- ***Indirect Metric (synonyms: Derived, Hybrid Metric):*** Es una métrica de un atributo que es derivada a partir de métricas de uno o más de los restantes atributos.

- ***Measurement Method (synonyms: Counting Rule, Protocol):*** Es la secuencia lógica particular de operaciones y las posibles heurísticas especificadas, que permiten la realización de una descripción de la métrica directa mediante la medición.

4.1.5 Términos asociados a la evaluación

Los términos asociados a la evaluación (ver Figura 4), tienen que ver con aquellos conceptos que permiten formalizar el modo, la forma y los mecanismos de cuantificación de la evaluación.

- ***Decision Criterion (synonym: Acceptability Level):*** Son umbrales, objetivos o patrones utilizados para determinar la necesidad de actuar, extender la investigación o describir el nivel de confianza en un resultado dado.

- ***Indicator (synonym: Criterion):*** Es el método de cálculo y escala definida adicionalmente al modelo y criterio de decisión, con el objeto de proveer una evaluación o estimación de un concepto calculable con respecto a las necesidades de información definidas.

- ***Indicator Value (synonym: Preference Value):*** Es el número o categoría asignada a un concepto calculable mediante la realización de una evaluación.

- ***Elementary Indicator (synonyms: Elementary Preference, Criterion):*** Es un Indicador que no depende de otros indicadores para evaluar o estimar su concepto calculable.

- ***Global Indicador (synonyms: Global Preference, Criterion):*** Es un indicador que es derivado a partir de otros indicadores para evaluar o estimar un concepto calculable.

- ***Elementary Model (synonym: Elementary Criterion Function):*** Es un algoritmo o función con criterio de decisión asociado que modelan un indicador elemental.

- ***Global Model (synonyms: Scoring, Aggregation Model or Function):*** Es un Algoritmo o función con criterio de decisión asociado que modelan un indicador global.

- ***Evaluation (synonym: Calculation):*** Es una actividad que utiliza la definición de un indicador para producir un valor del indicador.

4.2 Principales aspectos de C-INCAMI asociados al intercambio de datos y metadatos

Los conceptos antes expuestos, representan los términos de la base ontológica necesaria para poder definir formalmente un proyecto de medición y evaluación (para mayores detalles vea (Olsina & Martín, 2004; Olsina, Papa, et al., 2007)). La formalización del proyecto de medición y evaluación, representa un aspecto central para nuestro enfoque, si se considera que el objetivo del mismo se asocia con el procesamiento automático de mediciones y con la repetitividad del proceso de medición en sí mismo. La incorporación de etiquetas y su semántica asociada al dato de la medición, permitirían enriquecer adicionalmente, el tratamiento y análisis de datos en un modo consistente.

Si bien C-INCAMI fue diseñado originalmente para proyectos de M&E en organizaciones de software, los conceptos que integran su base conceptual ontológica son homogéneos a través de cualquier proyecto de M&E, sean estos asociados o no a software. Así, conceptos tales como el de métrica, escala, tipo de escala, indicadores, entre otros, son aplicables tanto a software como a cualquier entidad bajo análisis, como ser un paciente trasplantado ambulatorio. De este modo, es posible reutilizar la base conceptual de C-INCAMI para la formalización de cualquier proyecto de M&E, tal y como el asociado al monitoreo de pacientes trasplantados ambulatorios, introducido en la sub-sección 3.5. Este aspecto, es fundamental por cuanto, como se mencionó anteriormente, la formalización del proyecto de M&E es quien permite, en definitiva, el procesamiento automático de mediciones y la repetitividad del proceso de medición.

A partir de los principales conceptos y relaciones del marco C-INCAMI expuestos en la Figura 4, se deduce un orden natural de definición de los elementos del proyecto de medición, con vistas a su formalización. Dicho orden, desde el punto de vista del proceso, está bien documentado y puede ser abordado en detalle a través del trabajo de (Becker, Molina, et al., 2010).

Esta relación, asigna un orden determinado y lógico al momento de configurar una fuente de datos dentro del modelo de procesamiento, en el sentido que una fuente no puede asociarse con una métrica dada, si desconoce a qué atributo y transitivamente a qué entidad se vincula. Se

podría asociar una determinada métrica solo con una fuente de datos a los efectos de simplificar la definición del proyecto, pero ello no es conveniente, dado que al recibir las mediciones desde la fuente de datos, las mismas deben referir no solo a qué métrica corresponden, sino también a qué atributo cuantifican y de qué entidad en particular. De este modo, durante la configuración, la fuente si desease implementar una métrica *deberá* indicar conjuntamente a qué atributo se asocia, como así también la entidad que caracteriza.

A partir del orden lógico asignado por las relaciones de los distintos conceptos de C-INCAMI y su proceso asociado, el proceso de configuración de las fuentes de datos dentro del modelo integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, tendrá una secuencia bien definida y la cual es abordada en el capítulo posterior. No obstante, puede adelantarse a modo de ejemplo, una secuencia tentativa de cómo podría llevarse adelante la incorporación de una nueva fuente de datos a un proyecto de medición:

1. ¿Qué proyectos activos definidos formalmente hay?
2. De los proyectos activos informados en “1” ¿Qué entidades están definidas para el proyecto “P1”?
3. De las entidades definidas dentro del proyecto activo “P1” ¿Cuáles son los atributos disponibles para la entidad “E1”?
4. De los atributos disponibles para la entidad “E1” en el proyecto activo “P1” ¿Qué métricas asociadas se disponen para el atributo “A1”?

Resueltos estos interrogantes, se contará con la información suficiente para efectuar la vinculación entre la fuente de datos y demás conceptos, por lo que se estará entonces registrando una asociación de la fuente de datos actual con la métrica “M1” vinculada al atributo “A1” de la entidad “E1” dentro del proyecto activo “P1”.

4.3 Modificaciones y Extensiones Conceptuales a C-INCAMI

La presente sub-sección analiza la necesidad de extender algunos conceptos de C-INCAMI a los efectos de adaptarlo al procesamiento de flujos de datos centrado en metadatos de mediciones. De este modo, inicialmente se discute el concepto de grupo de seguimiento, el cual intenta agrupar lógicamente el origen de los flujos de medición indistintamente que dispongan de fuentes de datos o dispositivos diferentes. Seguido, se analizan aspectos asociados a precisión requerida y tolerancia a error para finalmente, concluir con la unidad lógica de medición y la base de entrenamiento para el comportamiento predictivo del modelo.

4.3.1 Grupos de Seguimiento (TraceGroup)

Como podrá apreciarse en el capítulo 5, cualquier fuente de datos puede incorporarse al modelo integrado de procesamiento de flujos de datos implementando una interface denominada DataSource y a partir de allí, transmitir mediciones identificando qué métrica, atributo y entidad tiene asociada.

De este modo, si bien las mediciones estarían vinculadas a una entidad, donde la entidad representaría una gran bolsa donde recaen las mediciones asociadas a sus atributos, no nos brinda

la posibilidad de generar dinámicamente sub-agrupaciones de las mismas que permitan efectuar un seguimiento. En este sentido y usando el caso de aplicación de la sub-sección 3.5, donde se dispone de una entidad "Paciente Trasplantado Ambulatorio", uno de los atributos indicados para la entidad era la Temperatura Corporal proponiéndose como una de las métricas asociadas el Valor de la Temperatura Axilar. Suponga que ha recibido un volumen muy grande de mediciones asociados a la métrica "Valor de la Temperatura Axilar", ¿Qué le dice ese volumen del Paciente Trasplantado Juan José Augusto? Ante tal situación contará con mucha información sobre la entidad en general pero nada sobre el objeto de medición en particular. Podría plantearse especializar la clase entidad pero ello obligaría a que el seguimiento sea solo asociado a una entidad dada "Juan Augusto", siendo incapaz de generar un grupo de seguimiento paralelo al objeto bajo medición.

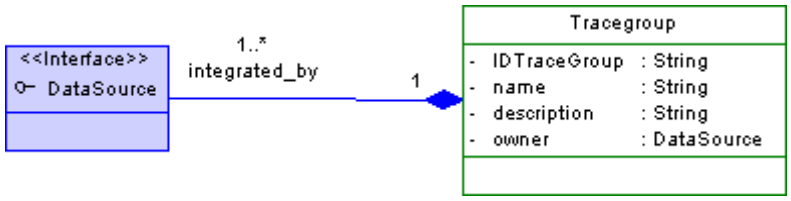


Figura 18. Conceptualización de la Clase TraceGroup

Para hacer frente a la situación planteada, se propone la clase TraceGroup, la cual es una composición de fuentes de datos tal y como se muestra en la Figura 18. La clase TraceGroup representa un modo de agrupar las fuentes de datos y en forma transitiva, toda medición proveniente de cada una de ellas.

Cada grupo de seguimiento dispondrá de un único propietario pero el mismo puede integrarse de varias fuentes de datos participantes. De este modo una fuente de datos puede integrar un grupo de seguimiento en carácter de propietario o participante, abriendo la posibilidad a que un grupo de mediciones dadas sean propagadas en cada uno de los grupos en los que participa una fuente. Así, lo que pareciera a prima facie redundancia, en realidad representa la consecución de diferentes formas de agrupar las mediciones para lograr un análisis en paralelo, aportando diferentes puntos de vistas y permitiendo que los modelos de decisión trabajen sobre los grupos de seguimiento.

De este modo, si los modelos de decisión pueden abordar su labor sobre grupos de seguimiento, las acciones preventivas pueden ser generadas desde diferentes puntos de vista y ser complementarias. Por ejemplo, el modelo de decisión para el paciente José Alonso puede estar arrojando resultados normales, pero resulta que del cruce de las propiedades contextuales de los pacientes trasplantados ambulatorios de General Pico a través del grupo de seguimiento por localidad (ver Figura 19), detecta un posible cambio brusco de temperatura que podría afectarlo, de este modo la alarma es complementariamente disparada desde el modelo de decisión asociado al grupo de seguimiento de General Pico.

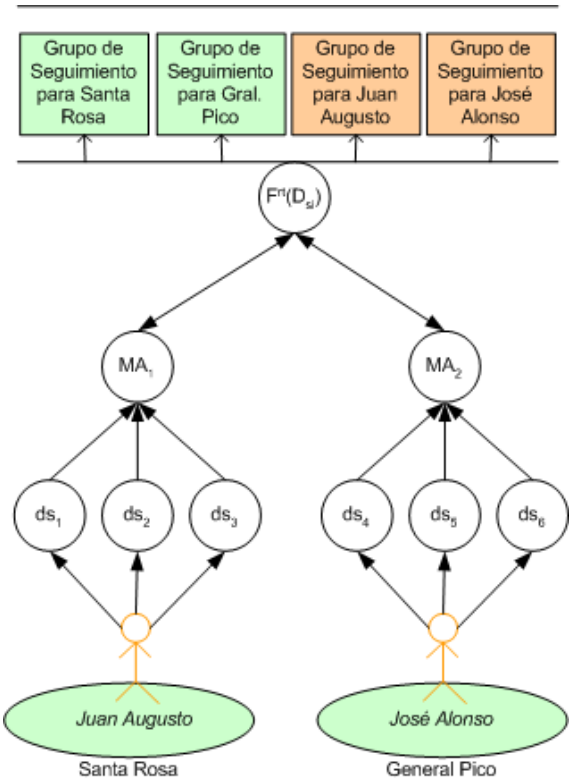


Figura 19. Esquema Conceptual de Grupos de Seguimientos

La incorporación de los grupos de seguimiento, abre la posibilidad a la generación de grupos de monitoreo complementarios y dinámicamente establecidos por los especialistas que llevan adelante los proyectos de medición. Es un modo de redundancia controlada de mediciones, que son distribuidas dentro de la propia función de reunión (Gathering Function -GF- en Figura 12 de la sub-sección 3.2) al momento de captar las mediciones desde las fuentes de datos.

4.3.2 Precisión Requerida y Tolerancia Máxima de Error

Uno de los aspectos centrales en un proceso de medición está asociado a la precisión requerida para una métrica dada como así también la ofrecida por una fuente de datos.

De este modo, lógicamente se deduce que *no toda fuente de datos puede implementar cualquier métrica*, solo lo podrá hacer en la medida que satisfaga los requerimientos de precisión y tolerancia de error. Originalmente C-INCAMI si bien consideraba el aspecto de precisión no explicitaba los requerimientos de precisión entera y decimal a nivel de escala ni tampoco la tolerancia de error en la métrica. Estos últimos aspectos han representado la propuesta de modificación, como puede apreciarse en la Figura 20, donde se incorpora la precisión entera y decimal a nivel de escala en forma conjunta con la tolerancia de error máximo permitido a nivel de métrica.

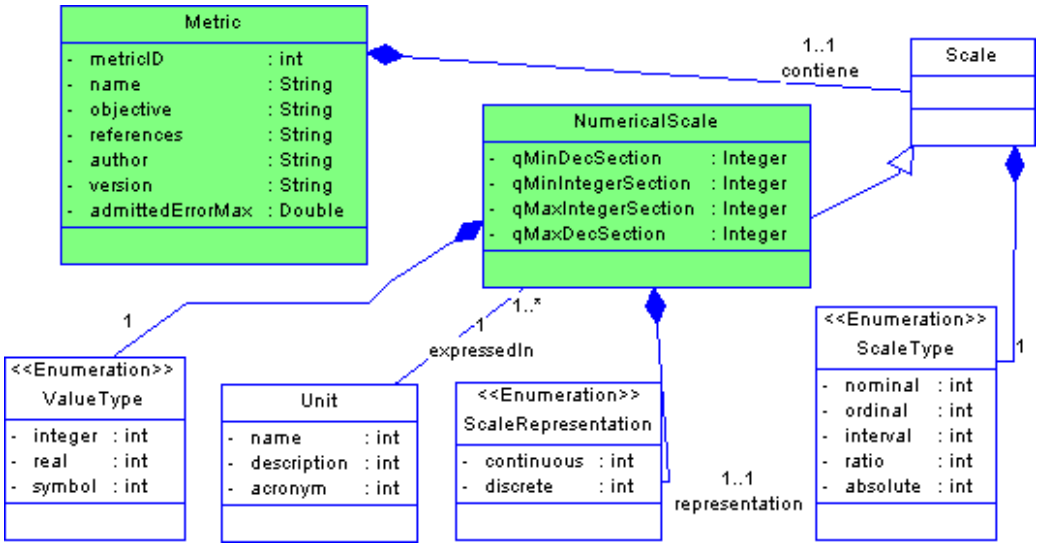


Figura 20. Incorporación de Precisión Requerida y Tolerancia de Error en C-INCAMI

Así, para que una fuente de datos pueda implementar una métrica se genera una condición necesaria la cual está asociada con que satisfaga los requerimientos de precisión y una condición suficiente que requiere que la fuente de datos satisfaga los requerimientos de tolerancia de error máxima permitida. Concluyendo en este sentido, una fuente de datos podrá implementar una métrica asociada a un atributo de una entidad dada, sí y solo sí es capaz de llevar adelante la medición bajo los requisitos de precisión impuestos por la escala numérica de la métrica y dentro de los límites de error impuestos por la tolerancia máxima de error asociado a la misma.

4.3.3 Base de Entrenamiento y Unidad Lógica de Medición

Los modelos de decisión empleados dentro del modelo integrado de procesamiento de flujos de datos son implementados a través de modelos de clasificación y en particular, árboles de decisión (ASHT con Bagging, ver sub-sección 2.5).

Los árboles de decisión, requieren un entrenamiento inicial para ser construidos y a posteriori, dentro de la misma aplicación, pueden ser actualizados incrementalmente. En esta última línea es que se planteó la necesidad de agrupar un conjunto de mediciones en forma persistente, bajo una agrupación que permita luego ser la base de entrenamiento previo para instruir al árbol de clasificación. Así es que surge la necesidad de generar la clase KnowledgeDataSet (ver Figura 21), la cual agrupa una o más mediciones persistentes, escogidas a partir de la recomendación de especialistas sobre los casos a monitorear en forma permanente a través de los flujos de datos

A través de la unidad lógica de medición expuesta en la Figura 21, las mediciones en el flujo viajarán asociadas con su distribución de probabilidad si correspondiese y conjuntamente con la situación de las propiedades contextuales, al momento en que la medición fue efectuada. Desde el punto de vista de procesamiento se estaría logrando una importante descongestión de la función de reunión sin incrementar la carga de trabajo del adaptador de mediciones y desde el punto de vista semántico, permite definir una unidad mínima de transmisión la cual en sí misma,

aporta información no solo de la medición sino también de la situación contextual dada en ese instante temporal.

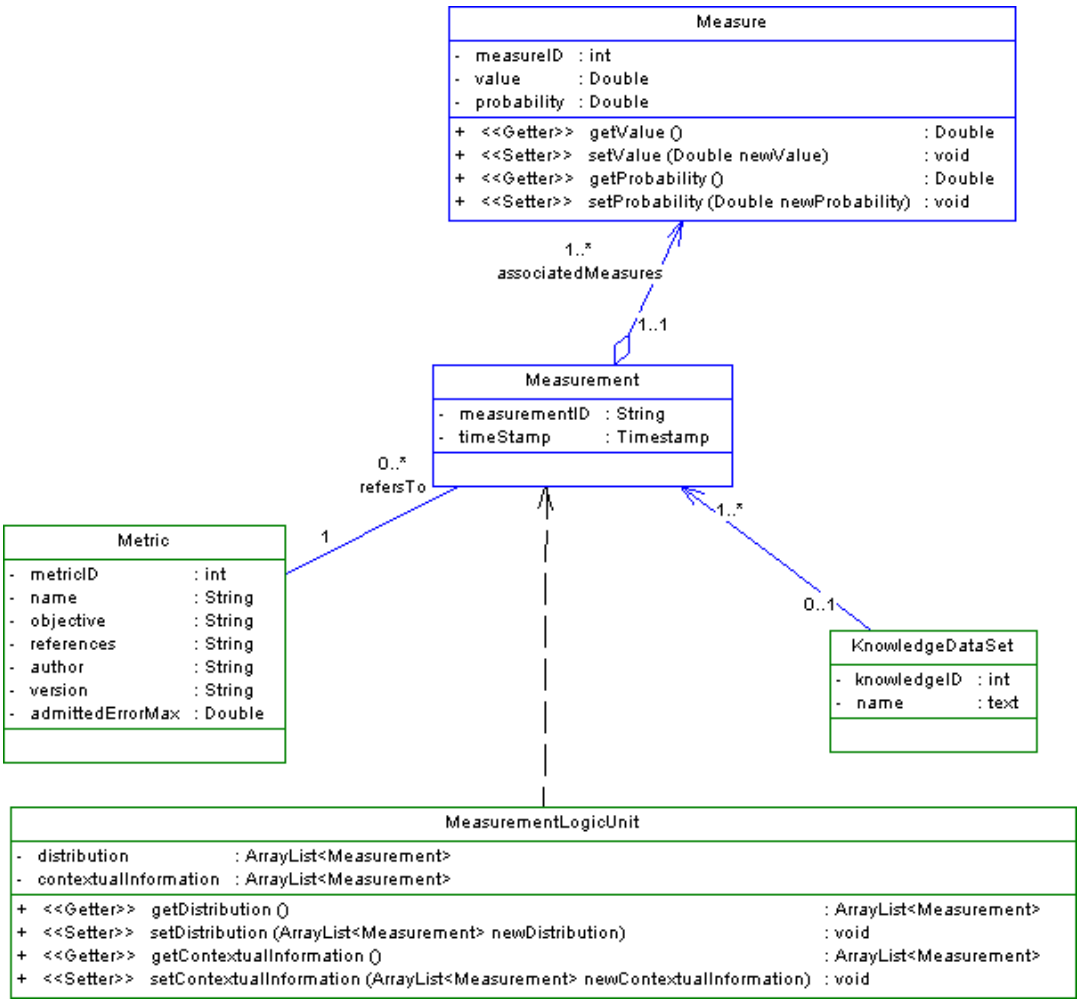


Figura 21. Clases KnowledgeDataSet y MeasurementLogicUnit

5 Configuración de Fuentes de Datos Heterogéneas

El capítulo 4, ha retomado la discusión de C-INCAMI efectuada en la sub-sección 2.2.3, para profundizar en primer lugar, los conceptos fundamentales asociados a un proyecto de medición y evaluación. Luego, se han analizado los principales aspectos de C-INCAMI a considerar para el intercambio de datos conjuntamente con metadatos, a los efectos de proponer las modificaciones y extensiones conceptuales necesarias por el enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones. Dentro de las modificaciones conceptuales, se analizó la necesidad de un grupo de seguimiento, aspectos asociados con la precisión y tolerancia de error, como así también lo referido al concepto de unidad lógica de medición y la base de entrenamiento requerida, para la implementación de los aspectos predictivos en los procesos de toma de decisiones.

Este capítulo discute el modo de integración de las fuentes de datos heterogéneas con respecto al EIPFDcMM. A seguir, se analizan los servicios de configuración intervinientes para enlazar la entidad bajo medición, sus atributos, las métricas definidas en un proyecto de medición y evaluación y los sensores/dispositivos utilizados para obtener las medidas.

A continuación, se analiza el gestor de fuentes de datos y su vínculo con la interface Datasource como con los sensores/dispositivos responsables de obtener las medidas. Se examina la necesidad del gestor de fuentes de datos como componente coordinador de la recolección de mediciones en mecanismos de arrastre (POP) de datos.

Luego, se discuten los modelos predictivos dentro de la etapa de recolección de mediciones, su vínculo con el gestor de fuentes de datos y la aplicación de los mismos para el filtrado consistente de los datos a transmitir a la función de reunión.

Finalmente, se analizan los procedimientos de configuración de los que deberá hacer empleo el adaptador de mediciones con respecto al EIPFDcMM, para poder asociar la entidad bajo análisis, los atributos de la entidad, el dispositivo/sensor que recolecta las medidas y la/s métricas del proyecto de medición y evaluación, con el objeto de transmitir

datos y metadatos estructurados en base a C-INCAMI a la función de reunión.

5.1 Interface Datasource

Como se ha dicho en la sub-sección 1.2, parte central del núcleo de la tesis se focaliza en permitir la incorporación de fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados en C-INCAMI, permitan realizar análisis estadísticos de un modo consistente a los efectos de implementar un comportamiento detectivo y a su vez, permitan incorporar información contextual a las mediciones, para enriquecer la función de clasificación con el objeto de implementar el comportamiento predictivo.

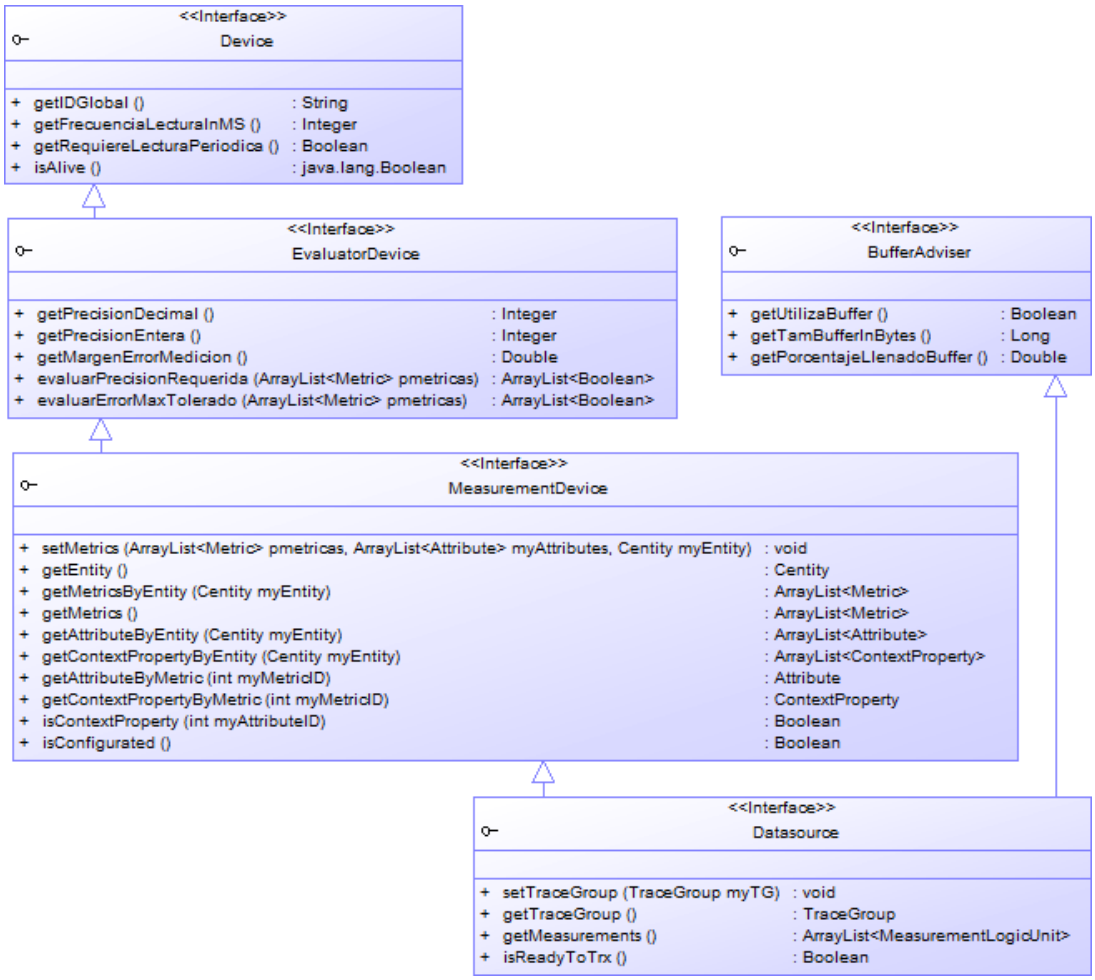


Figura 22. Principales Relaciones de la Interface Datasource

Así, la presente sub-sección analiza la interface *Datasource* (ver Figura 22), con el objeto de que cualquier dispositivo que desee formar parte del enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, implemente la misma a los efectos de comunicarse en un lenguaje homogéneo con el adaptador de mediciones (MA, ver sub-sección 3.2). De este modo, las fuentes de datos informarán las medidas a MA través de dicha interface, mediante una serie de métodos cuyos objetivos se encuentran claramente establecidos.

La interface Datasource tiene asociada una jerarquía de interfaces (ver Figura 22), las cuales abstraen conceptualmente diferentes estadios de un dispositivo de medición. La interface *Device,* representa cualquier dispositivo capaz de identificarse en forma única e informar sus mecanismos de lectura de datos. La interface *EvaluatorDevice*, representa un dispositivo capaz de verificar su precisión decimal y entera, junto con el error máximo tolerado, a los efectos de determinar si está en condiciones o no de implementar una determinada métrica. La interface *MeasurementDevice*, corresponde a un dispositivo capaz de evaluar si puede implementar un subconjunto de métricas dadas, y adicionalmente, configurar en base al proyecto de M&E basado en C-INCAMI, qué métricas implementará (sean asociadas a atributos de entidad o propiedades de contexto), para una determinada entidad bajo estudio. La interface *BufferAdviser*, determina un subconjunto mínimo de métodos de notificación sobre estado del buffer, que sirven para monitorear el estado de los mismos en aquellas fuentes de datos que los implementen. Finalmente, la interface *Datasource*, hereda de *MeasurementDevice* y *BufferAdviser*, incorporando la capacidad específica de recolectar las mediciones, en base a la configuración en el proyecto de M&E basado en C-INCAMI, como así también de asociarse e informar a qué grupo de seguimiento (ver sub-sección 4.3.1) se vincula.

Retomando el caso de aplicación presentado en la sub-sección 3.5 (ver Figura 15), un dispositivo capaz de identificarse es la Palm Treo 750 (por ejemplo, a través de su International Mobile Equipment Identity -IMEI-) por lo que podría implementar la interface *Device*. No obstante, éste no es capaz de verificar la precisión decimal o entera por cuanto es un Smartphone y su finalidad no es medir, por ende no es un dispositivo evaluador. Por otro lado, el dispositivo Garmin Forerunner 405 GPS, permite medir frecuencia cardíaca y determinar el posicionamiento georeferenciado (latitud, longitud y altura), de este modo podría implementar la interface *EvaluatorDevice*. Si este último dispositivo, pudiese adicionalmente implementar un subconjunto de métricas dadas en función de C-INCAMI, sería considerado un dispositivo de medición (interface *MeasurementDevice*) en nuestra jerarquía, para finalmente, convertirse en una fuente de datos (interface *Datasource*) al momento de asociarse con un grupo de seguimiento, que para el caso de aplicación, es el paciente que porta dicho dispositivo.

El método *getIdGlobal* de la interface *Device,* permite definir el identificador único del dispositivo para con el gestor de fuentes de datos, por ejemplo, el número de serie del dispositivo. A través de dicho método, el gestor de fuentes de datos puede identificar el origen de las medidas y por carácter transitivo, a qué entidad, atributo y métrica se asocia la misma.

Los métodos *getPrecisionDecimal/getPrecisionEntera/getMargenErrorMedicion* de la interface *EvaluatorDevice*, informan la precisión entera y decimal del dispositivo responsable de tomar la medida, como así también el margen de error tolerado por el mismo. Para el caso del Garmin Forerunner 405 GPS (GF-405GPS), serían la precisión y errores asociados a la medición de la frecuencia cardíaca, como el vinculado a la determinación de latitud, longitud y altura del paciente. Estos aspectos son fundamentales en el procedimiento de configuración del dispositivo, dado que el mismo debe soportar la precisión y error exigido en la definición de la métrica en el proyecto de medición y evaluación. De este modo, los métodos *evaluarPrecisionRequerida*

/evaluarErrorMaxTolerado de la misma interface*,* permiten pasar como parámetros, objetos asociados a la definición de la métrica del proyecto de M&E, y concluir sobre si los mismos pueden o no ser implementados por el dispositivo de medición que implementa la interface Datasource y por carácter transitivo, la interface *EvaluatorDevice*.

Notar que C-INCAMI incluye el nombre *Entity* para el concepto de entidad bajo análisis. A los efectos de implementación, éste fue cambiado por *Centity* para evitar inconvenientes con la clase *Entity* asociada a los gestores de persistencia basados en Java Beans.

El método *setMetrics* en la interface *MeasurementDevice*, permite asociar al dispositivo que implementa la interface Datasource, y por carácter transitivo la interface *MeasurmentDevice*, las métricas con las que se vincula, con qué atributo se asocia cada métrica y la entidad a la que corresponden dichos atributos. Por ejemplo el dispositivo GF-405GPS implementaría la métrica asociada al atributo frecuencia cardíaca de la entidad paciente, como así también la propiedad de contexto vinculada al posicionamiento del mismo. De este modo, al momento de recibir una medida desde el dispositivo, el mismo puede informar a qué métrica, de qué atributo y a qué entidad bajo análisis corresponde, evitando inconsistencias o errores de asociación en la medida. Dado que el concepto *ContextProperty* (concepto asociado a la propiedad de contexto) de C-INCAMI es un atributo, este método puede asociar en forma transparente métricas tanto a atributos de la entidad bajo análisis, como propiedades de contexto que se asocien a la misma.

Los métodos *setTraceGroup/getTraceGroup* de la interface *Datasource*, permiten asociar al dispositivo que implementa la interface con un grupo de seguimiento dado. Por ejemplo, si bien el GF-405GPS se asocia a la entidad paciente, puede existir ‘n’ pacientes siendo monitoreados en simultáneo, de este modo el dispositivo GF-405GPS identificado mediante su número de serie, puede asociarse a la entidad paciente cuyo grupo de seguimiento es “Juan Perez”. Esto, agrupa a las medidas asociadas a las métricas de los atributos/propiedades de contexto de la entidad bajo análisis, a dicho grupo de seguimiento en particular al momento de informarse a la función de reunión.

Los métodos *isAlive/isConfigurated/isReadyToTrx* tienen que ver con el estado en que se encuentra el dispositivo dentro del EIPFDcMM. El método *isAlive* de la interface *Device*, indica si el dispositivo se encuentra activo, esto es encendido. El método *isConfigurated* de la interface *MeasurementDevice,* indica si el dispositivo se ha registrado dentro del EIPFDcMM, se le ha asociado al menos una métrica a un atributo de una entidad bajo análisis de un proyecto de medición y evaluación y posee un grupo de seguimiento asignado. Finalmente, el método *isReadyToTrx* de la interface *Datasource*, indica si el dispositivo está configurado y contiene medidas que informar a la función de reunión.

El método *getMeasurements* de la interface *Datasource,* permite implementar la recolección de mediciones por arrastre (POP) desde el dispositivo para las métricas que implemente, asociándolas a los atributos de la entidad bajo análisis correspondientes, retornándose bajo la forma de unidad lógica de medición introducida en la sub-sección 4.3.3.

De este modo, los dispositivos conectados al paciente trasplantado ambulatorio del caso de aplicación (ver Figura 15), que debieran implementar la interface *Datasource* serían: a) El Omron ECO-TEMP, para poder informar la temperatura axilar, b) El termómetro digital, para poder informar la temperatura y humedad ambiente, c) el GF-405GPS, que proveerá la frecuencia cardíaca del paciente, junto con su posicionamiento (latitud, longitud y altura), d) El Omron HEM-711AC, proveerá la presión arterial del paciente y finalmente, e) El Fischer Series 3312, informará la presión atmosférica del hábitat donde reside el paciente. Cada uno de los dispositivos enunciados, al implementar la interface *Datasource*, informarán al adaptador de mediciones, a través de tecnología inalámbrica (por ejemplo: bluetooth, entre otras), cada una de sus mediciones. Como puede observarse a simple vista, cada uno de los dispositivos enunciados, corresponden a diferentes fabricantes, cada uno posee un objetivo específico, el modo de exportar sus datos difiere entre ellos, pero sin embargo, gracias a la implementación de la interface *Datasource*, son capaces de informar sus mediciones e interactuar con EIPFDcMM, a los efectos de automatizar el proceso de medición asociado a cada paciente trasplantado ambulatorio.

Resumiendo la sub-sección, el hecho de implementar la interface Datasource, garantiza que dicho dispositivo de medición, satisface un conjunto de operaciones básicas necesarias para la interacción y transmisión de las mediciones.

5.2 Servicios de Configuración

Previo a transmitir medición alguna, el dispositivo que implementa la interface DataSource (en adelante fuente de datos), debe ser configurada, por ejemplo, el dispositivo GF-405GPS, por nombrar uno de los enunciados en el caso de aplicación. Dicha configuración, implica registrar mínimamente el identificador de la fuente de datos (por ejemplo, su número de serie), asociar las métricas que implementará con el o los atributos y/o propiedades de contexto de una entidad bajo análisis (por ejemplo, la métrica *valor de la frecuencia cardíaca* para el atributo *frecuencia cardíaca* de la entidad *paciente trasplantado ambulatorio*) y asociarle un grupo de seguimiento (por ejemplo: Juan Perez).

Para poder llevar adelante la tarea de configuración, se ha planteado una interface base denominada *configurationServices*, la cual es especializada en *configurationServiceConsumer* y *configurationServiceProvider* según si el rol a desempeñar es el de consumidor del servicio o proveedor del mismo (ver Figura 23).

Al igual que la interface *Datasource*, la interfaces asociadas a configuración están basadas en C-INCAMI, por tal motivo, emplean conceptos de dicho marco para llevar adelante la tarea de configuración. En la Figura 23 puede apreciarse cada una de las interfaces de configuración junto con su dependencia de los conceptos del marco C-INCAMI.

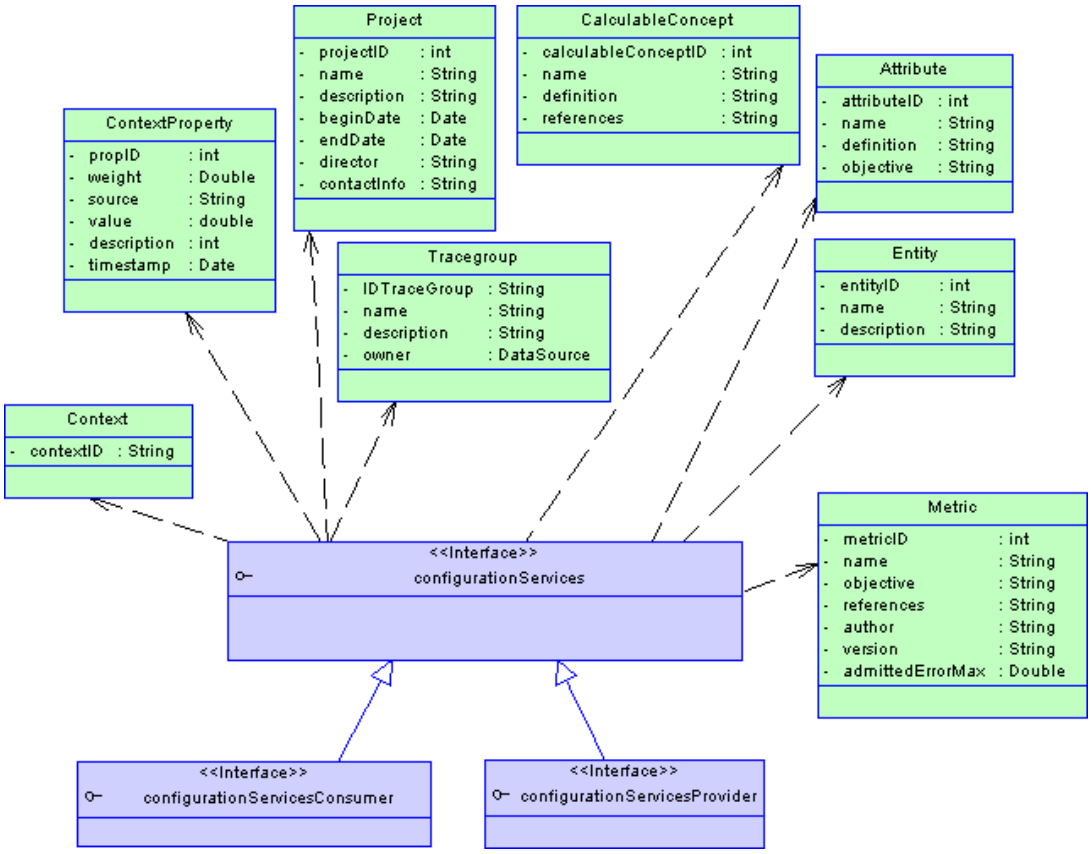


Figura 23. Interfaces de Configuración y sus Dependencias

La interface *configurationServices* (ver Figura 24) contiene una serie de métodos comunes tanto a quien consume el servicio de configuración como para quien lo provee y a continuación se discuten los principales métodos asociados.

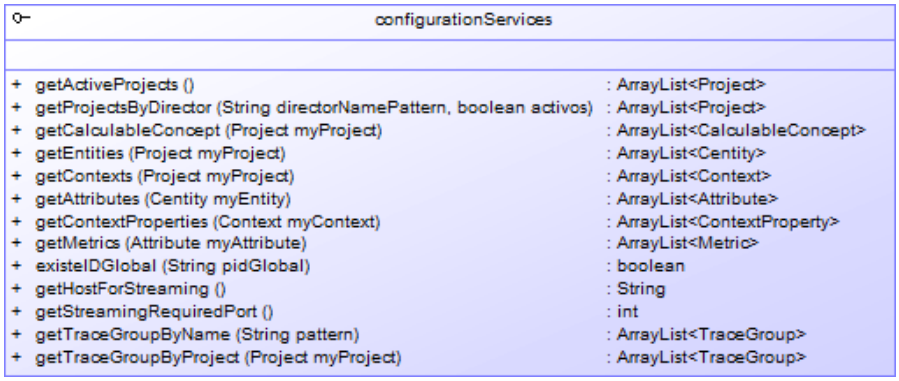


Figura 24. Interface configurationServices

Los métodos *getActiveProjects/getProjectsByDirector* permiten localizar uno o más objetos *Project* correspondientes a un proyecto de medición y evaluación definido dentro del repositorio de proyectos (ver C-INCAMI DB en sub-sección 3.2). Por ejemplo, el método podría retornar el proyecto del caso de aplicación asociado al *Monitoreo de Pacientes Trasplantados Ambulatorios*.

Los métodos *getCalculableConcept/getEntities/getContexts* permiten obtener, a partir de un proyecto dado, los conceptos calculables, las entidades o los contextos definidos y asociados al mismo. El método *getTraceGroupByProject* permite recuperar un conjunto de grupos de seguimiento definidos para el proyecto indicado o cuya denominación, coincida con el patrón indicado. Por ejemplo, en el caso del proyecto *Monitoreo de Pacientes Trasplantados Ambulatorios* podría retornarse grupos de seguimientos tales como *Juan Perez*, *Luis García*, *Esteban Quito*, entre otros.

Los métodos *getAttributes*/*getContexts* permiten obtener los atributos definidos para una entidad dada y las propiedades de contexto definidas para un contexto dado respectivamente, a partir de indicar una entidad o un contexto determinado. En este sentido, para la entidad paciente trasplantado ambulatorio, si se solicitan los atributos se retornaría por ejemplo la frecuencia cardíaca, la temperatura axilar, entre otros. Por otro lado, si se indicase el contexto del paciente trasplantado ambulatorio, se retornarían como propiedades contextuales a la humedad, el posicionamiento del paciente, la temperatura ambiental, entre otras.

El método *getMetrics* permite recuperar el conjunto de métricas definidas en el proyecto de medición y evaluación para un atributo dado. En este sentido, recordar que una propiedad de contexto es un atributo, por lo que se puede indicar como parámetro tanto a un atributo como a una propiedad contextual a los efectos de obtener sus métricas asociadas. De este modo, si se indicase la propiedad de contexto asociada con el posicionamiento del paciente, se retornaría entre sus métricas, la métrica *valor del posicionamiento del paciente*, mientras que si se indicase el atributo frecuencia cardíaca de la entidad paciente trasplantado ambulatorio, se retornaría entre sus métricas, la métrica *valor de la frecuencia cardíaca*.

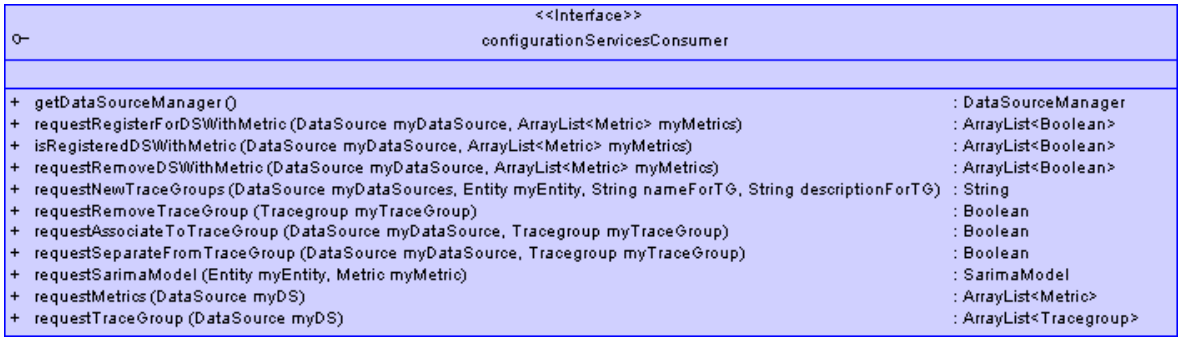


Figura 25. Interface configurationServicesConsumer

La interface *configurationServicesConsumer* (ver Figura 25) hereda la funcionalidad de configurationServices, e incorpora métodos específicos a ser utilizados por el cliente del servicio de configuración. A continuación se discuten los principales métodos asociados a la interface del cliente.

El método *getDataSourceManager* permite obtener una instancia del gestor de fuentes de datos en caso de no estar creado o bien, de estar creado, retorna una referencia al mismo. El gestor de fuentes de datos (*DataSourceManager*), el que será discutido en la siguiente sub-

sección, tiene por objetivo la administración de las fuentes de datos (por ejemplo, el GF-405GPS) conectadas al adaptador de mediciones y la implementación de la interface de configuración cliente.

El método *requestRegisterForDSWithMetric* permite efectuar la asociación ante el EIPFDcMM entre la fuente de datos y las métricas que implementa. Por ejemplo, mediante este método, se indicaría que el GF-405GPS implementa la métrica *valor de la frecuencia cardíaca* asociada al atributo *frecuencia cardíaca* de la entidad *paciente trasplantado ambulatorio*. En forma complementaria, los métodos *isResiteredDSWithMetric* y *requestRemoveDSWithMetric* permiten verificar si una fuente de datos se encuentra previamente registrada con una métrica dada o bien, disolver tal asociación respectivamente.

Los métodos *requestNewTraceGroups* y *requestRemoveTraceGroups* permiten la creación o eliminación respectivamente de un grupo de seguimiento a partir de los servicios de configuración clientes. Esto posibilita a una fuente de datos crear su propio grupo de seguimiento y asociar a otras fuentes de datos, en forma dinámica y compartida al resto de los integrantes del proyecto de medición y evaluación. Por ejemplo, si el GF-405GPS es la primera fuente de datos que se registra para el paciente "Juan Perez", la misma puede crear el grupo de seguimiento con dicho nombre y luego asociar a los restantes dispositivos del paciente (ver Figura 15) al grupo. En forma complementaria, los métodos *requestAssociateToTraceGroup* y *requestSeparateFromTraceGroup* permiten a una fuente de datos asociarse o separarse de grupos de seguimientos predefinidos.

El método *requestSarimaModel* permite solicitar al EIPFDcMM a través de los servicios de configuración los parámetros del modelo SARIMA (***S**easonal – **A**uto **R**egressive **I**ntegrated **M**oving **A**verage*) (Box & Jenkins, 1991) para una métrica dada. La aplicación de modelos SARIMA en la recolección de mediciones, tiene como antecedente a PRESTO (Li, Ganesan, et al., 2006) y será discutido en la sub-sección 5.4.

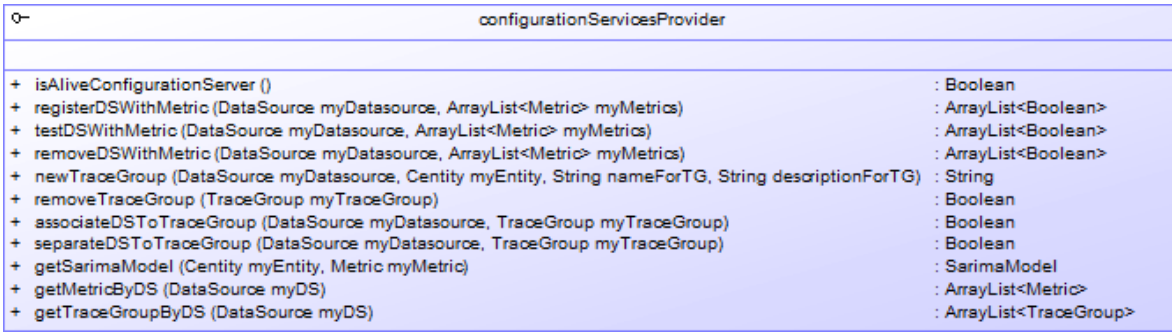


Figura 26. InterfaceConfigurationServicesProvider

Los métodos *requestMetrics* y *requestTraceGroups* tienen por objeto recuperar las métricas implementadas y el grupo de seguimiento respectivamente a partir del identificador de la fuente de datos, como forma de resguardo ante una eventual pérdida de información o falla en los datos de configuración local al adaptador de mediciones. De este modo, si el GF-405GPS solicita

sus métricas, obtendría como respuesta las métricas *valor de la frecuencia cardíaca* y *valor del posicionamiento del paciente*. Por otro lado, si solicita el grupo de seguimiento al que se asocia, se retornaría, en base al número de serie de la fuente de datos, un grupo determinado como ser "Juan Perez".

La interface *configurationServicesProvider* (ver Figura 26) hereda la funcionalidad de configurationServices, e incorpora métodos específicos a ser utilizados por el proveedor del servicio de configuración. Para no ser reiterativo, los conceptos asociados a los métodos del proveedor del servicio de configuración, son similares conceptualmente a los planteados por el cliente, sólo que el proveedor tiene la responsabilidad de registrar persistentemente las asociaciones o separaciones de fuentes de datos para con métricas o grupos de seguimiento, como así también el hecho de aprovisionar la información requerida a partir del repositorio donde se encuentra la definición de los proyectos de medición y evaluación (ver C-INCAMI DB en sub-sección 3.2).

5.3 Gestor de Fuentes de Datos (DataSourceManager)

Cada dispositivo que implementa la interface *Datasource* en EIPFDcMM, por ejemplo el GF-405GPS, es gestionado en el adaptador de mediciones mediante el gestor de fuentes de datos (DataSourceManager –DSM-). Dentro de las principales responsabilidades del gestor de fuentes de datos se encuentra el agregado y remoción de las fuentes de datos dentro del proceso de recolección de mediciones que éste coordina, la verificación de su estado y la implementación de la interface de configuración *configurationServicesConsumer.*

Mediante la implementación de la interface *configurationServicesConsumer*, el gestor de fuentes de datos permite llevar adelante la totalidad de las tareas de configuración, abstrayendo a las fuentes de datos de dicha problemática, independientemente de la tecnología con la que esta cuente.

El gestor de fuentes de datos (ver Figura 27) guarda referencia de las diferentes fuentes de datos que administra, mediante una tabla hash a la cual accede mediante el identificador de la fuente, el cual es único y puede verificarse a través de los servicios de configuración al momento de su inicialización. También el gestor permite incorporar una tabla hash con parámetros de diferentes modelos predictivos SARIMA, los cuales pueden ser empleados al momento de la recolección de mediciones, como se expondrá en la sub-sección 5.4.

Adicionalmente, el gestor almacena copia local de la información de conectividad solicitada al servidor de configuración con respecto al sitio donde transmitir (esto es el puerto y host de destino) las mediciones. De este modo, se evita consultar reiteradamente dicha información en forma previa a la transmisión y no representa un riesgo significativo desde el punto de vista de su desactualización, ya que no posee características de volatilidad o situaciones que produzcan cambios bruscos de la misma.

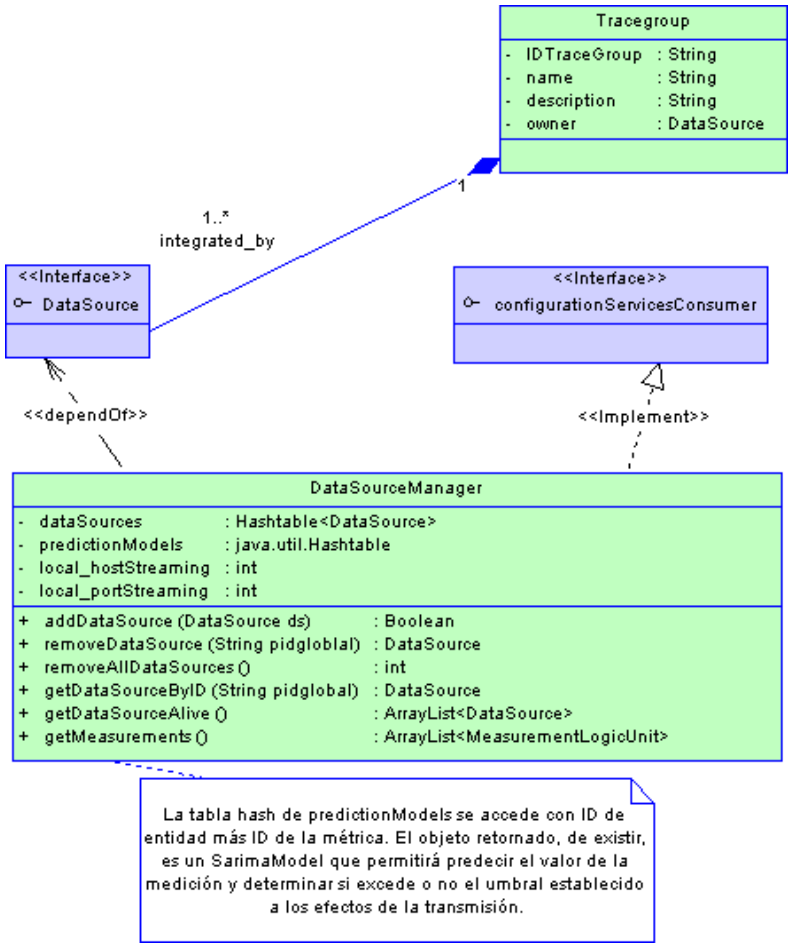


Figura 27. Clase DataSourceManager y sus Principales Relaciones

El DSM es el responsable de recolectar las unidades lógicas de medición (ver sub-sección 4.3.3) desde cada una de las fuentes de datos que gestiona, en la medida que se encuentren disponibles. Si bien el gestor se responsabiliza por la recolección, no es su responsabilidad determinar la periodicidad o el momento en que se efectuará la misma, ello será administrado por la clase *MeasurementAdapter,* la cual será abordada en el siguiente capítulo. El mecanismo de recolección de mediciones empleado por defecto es de arrastre (POP), dejando abierta la posibilidad a futuras especializaciones, teniendo en cuenta otros tipos de técnicas de recolección, tal como mecanismos de empuje (PUSH) a partir de las fuentes de datos. A continuación se discuten los principales métodos asociados a DSM.

Los métodos *addDataSource* y *removeDataSource* permiten agregar o remover una fuente de datos al DSM. En forma complementaria, el método *removeAllDataSources* permite eliminar la totalidad de las fuentes de datos que se encuentren gestionadas en ese momento por DSM.

El método *getDataSourceAlive* retorna un conjunto de fuentes de datos vivas. Por fuente de datos vivas se entiende a aquella fuente de datos cuyo dispositivo se encuentra encendido y escuchando peticiones, pero no garantiza que se encuentre listo para transmitir mediciones.

Recordar que para resolver este último aspecto, la interface *configurationServicesConsumer* proponía el método *isReadyToTrx*.

El método *getMeasurements* recorre cada una de las fuentes de datos (por ejemplo, el GF-405GPS, Fischer Series 3312, Omron HEM-711AC, entre otros expuestos en la Figura 15) administradas por DSM que se encuentren lista para transmitir, recupera sus mediciones y retorna una lista de unidades lógicas de medición (ver sub-sección 4.3.3).

5.4 Modelos Predictivos en la Recolección de Mediciones

La idea de aplicar modelos predictivos del tipo SARIMA al proceso de medición, ha sido discutida previamente en PRESTO (Li, Ganesan, et al., 2006). En tal prototipo, si el valor predicho por el recolector no difería, en valor absoluto, de un margen de error preestablecido, el mismo no era informado dado que no aportaba más información que la predicción. De este modo, ante la ausencia de valor, el procesador central tomaría la predicción para el instante de tiempo ausente en el flujo informado.

Esta idea fue incorporada en DSM como un aspecto opcional. Opcional en el sentido que posiblemente no toda métrica para un atributo de una entidad dada, ante la ausencia de valor, requiera *suplantar* la medición por una predicción. Esto último, se sustenta en que la ausencia de valor en sí misma, es información y puede exponer una situación en particular. De este modo se planteó una tabla hash dentro de DSM, la cual es accedida mediante la concatenación del identificador de la entidad, un punto y el identificador de la métrica. Si en el proceso de recolección de mediciones, el DSM desea emplear un modelo predictivo, primero verificará que cuente con los parámetros u objeto que implemente el modelo predictivo dentro de la tabla hash. De contar con el mismo, DSM aplica el modelo y obtiene la predicción, la compara contra la medición si estuviese disponible y determina, en función del margen de error establecido para el modelo predictivo, si informar la misma o realizar el reemplazo del valor por la predicción (en caso de que la fuente de datos no estuviese activa o lista para transmitir). Por otro lado, si DSM detecta que no existe modelo predictivo para el par métrica – entidad, informará las mediciones disponibles sin alterarlas.

La familia de modelos SARIMA (Box & Jenkins, 1991) se aplican a series temporales univariadas, en donde el período de variabilidad puede ser determinado y el seguimiento se efectúa sobre un individuo. La estimación del modelo no es simple y su cálculo está bien documentado en (Box & Jenkins, 1991). De este modo, si la idea fuese aplicar series temporales sobre datos multivariados, SARIMA posiblemente el mejor enfoque por los motivos enunciados y en su lugar, deberán aplicarse posiblemente otras alternativas tales como análisis de datos longitudinales (Diggle, Heagerty, et al., 2002).

Los modelos SARIMA son una buena alternativa asociado a propiedades contextuales tales como la temperatura o humedad ambiente, dado que independientemente del grupo de seguimiento al que se asocia la entidad bajo análisis, si el contexto es el mismo, el modelo tiene sustento, puede establecerse un período bien definido y la variable bajo análisis es única (por

ejemplo: humedad o temperatura). Claro que si lo que se desea es predecir ambas series, se tendrá un modelo SARIMA para temperatura y otro modelo posiblemente diferente para humedad.

Para la construcción de modelos SARIMA, primero debe identificarse el modelo (p,d,q) x $(P,D,Q)_S$, donde *p o P* es el orden del proceso auto-regresivo, *d o D* es la cantidad de veces que se diferenció una serie de observaciones, *q o Q* es el orden del proceso de medias móviles, (p,d,q) se refiere a la tendencia regular o parte regular de la serie, mientras que (P,D,Q) se refiere a las variaciones estacionales. Identificado el modelo, en segundo lugar se debe determinar sus parámetros, a partir de un conjunto de datos asociado a la entidad y métrica en cuestión. Para modelar dicho conjunto de datos, se ha incorporado la clase *MeasurementWindow* (ver Figura 28), la cual desempeña el rol de *Sliding Window o ventana corrediza* (ver 2.1.2).

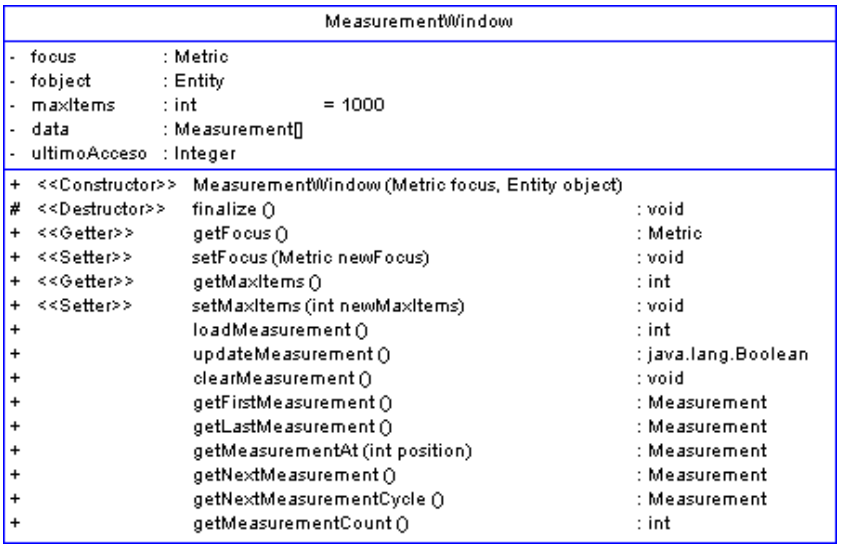


Figura 28. Clase MeasurementWindow

La clase *MeasurementWindow* brinda las funciones básicas de una lista, con la posibilidad de manejar dinámicamente el tamaño o longitud de la ventana. Las mediciones de las que se nutre MeasurementWindow no tienen el carácter de on-line con respecto a las fuentes de datos, sino que forman parte de un subconjunto de mediciones seleccionadas y almacenadas en la base de datos C-INCAMI (ver sub-sección 3.2) que integran la base de entrenamiento empleadas para la formulación de modelos predictivos.

Dada la utilidad de los modelos SARIMA con respecto a las propiedades contextuales, se incorpora la Clase *SarimaModel* (ver Figura 29) la cual contiene la métrica y entidad a la que se aplica, los parámetros que definen el modelo SARIMA, el umbral de tolerancia de error (para comparar lo medido versus lo predicho) y los parámetros *theta* del modelo. Los parámetros theta, empleados para el cómputo de la predicción, son modelados mediante la clase *SarimaParameter* la cual está asociada indefectiblemente al modelo. Los parámetros son estimados una vez determinado el modelo e informado el grupo de entrenamiento que se empleará para su construcción a través del objeto del tipo MeasurementWindow.

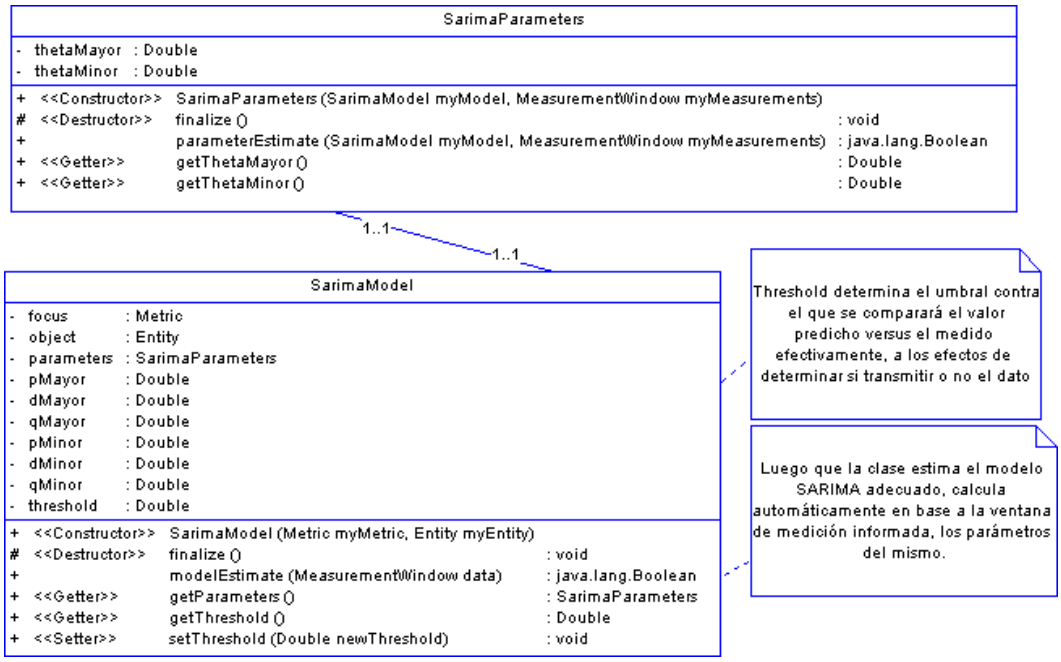


Figura 29. Clases SarimaModel y SarimaParameters

Existen dos operaciones sustancialmente importantes en las clases *SarimaModel* y *SarimaParameter*. En SarimaModel la operación *modelEstimate* recibe el conjunto de datos mediante el objeto del tipo *MeasurementWindow* e intenta determinar el modelo en forma iterativa. Por otro lado, una vez determinado el modelo, la operación *parameterEstimate* de la clase *SarimaParameters*, calculará el valor de los parámetros theta para poder efectuar el cómputo de las predicciones. Lo que es transmitido a DSM, es el modelo y sus parámetros ya computados, dado que éste debe utilizar el modelo predictivo y no calcularlo. El cálculo de los modelos predictivos, forma parte del proceso de inicialización del EIPFDcMM y se nutre a partir de la base de datos C-INCAMI, para poder computar los mismos.

5.5 Procedimientos de Configuración

Dentro de los procedimientos básicos de configuración de una fuente de datos en el EIPFDcMM, se encuentran: a) la asociación de cada fuente de datos con una o más métricas asociadas a un atributo de entidad, por ejemplo, asociar el dispositivo GF-405GPS con la métrica *valor de la frecuencia cardíaca* del atributo *frecuencia cardíaca* asociado a la entidad *paciente trasplantado ambulatorio*; b) la asociación de cada fuente de datos con una o más propiedades de contexto de un contexto definido en el proyecto de medición y evaluación, por ejemplo, asociar el GF-405GPS con la métrica *valor del posicionamiento del paciente* de la propiedad contextual *posicionamiento del paciente* para el contexto de la entidad *paciente trasplantado ambulatorio*; c) la incorporación o creación de grupos de seguimiento, por ejemplo, “Juan Perez”; y d) la actualización de modelos predictivos.

Los aspectos que intervienen a lo largo del proceso de configuración, están dados por: 1) la fuente de datos que implementa la interface *Datasource*; 2) el gestor de fuentes de datos (DSM)

que implementa la interface *configurationServicesConsumer*; y 3) la clase que implemente la interface *configurationServicesProvider,* en este caso *dsipsProvider.*

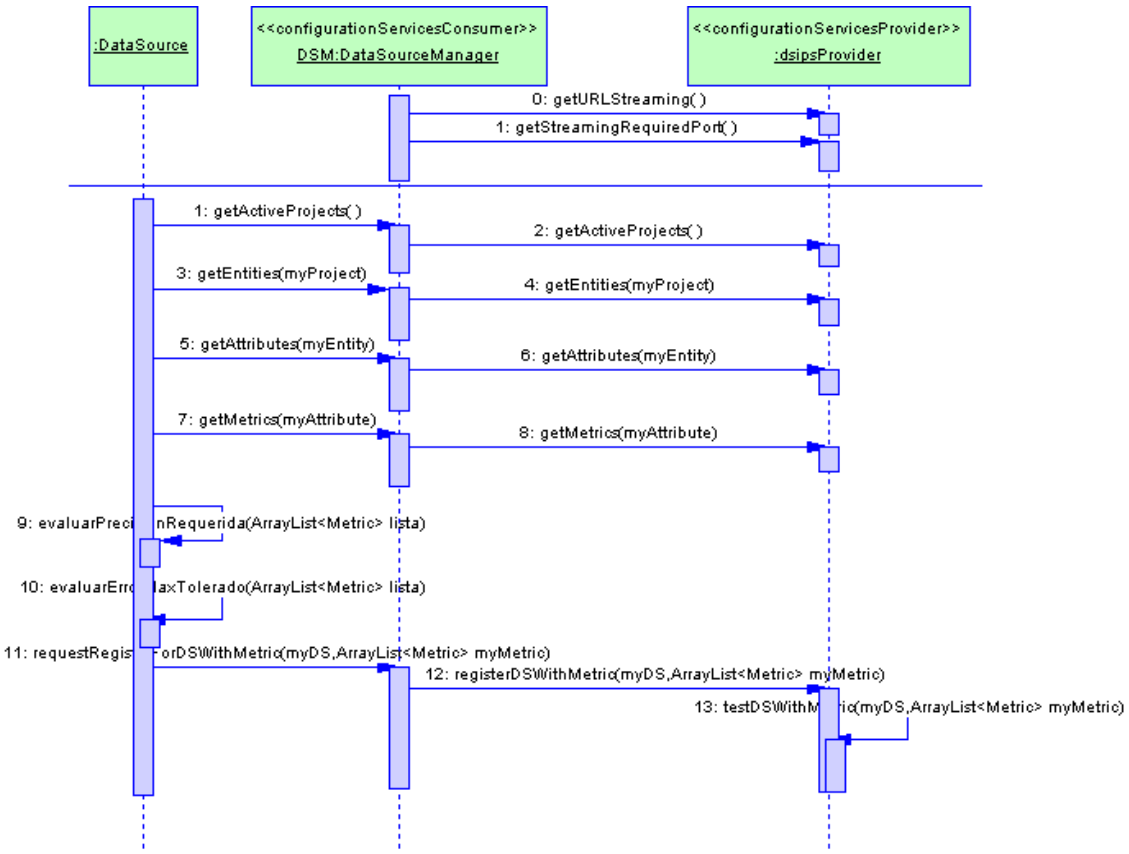


Figura 30. Diagrama de Secuencia de la Asociación de una Fuente de Datos con una Métrica asociada a un Atributo de Entidad

La Figura 30 muestra un diagrama de secuencia dividido en una sección superior y otra inferior separada por una línea continua. La parte superior del diagrama, expone la operatoria básica que el DSM efectúa durante su inicialización y la que consiste, en solicitar el nombre del host y puerto al cual se transmitirá posteriormente las mediciones. La parte inferior por otro lado, expone el procedimiento típico para asociar una métrica de un atributo de la entidad interviniente a la fuente de datos. Éste último, comienza solicitando los proyectos activos (por ejemplo: *Monitoreo de Pacientes Trasplantados Ambulatorios*), de los cuales la fuente mediante la interface de configuración deberá solicitar uno. A partir del proyecto seleccionado, solicita se le remitan las entidades vinculadas (por ejemplo, *Paciente Trasplantado Ambulatorio*). Al recibir las entidades, selecciona sobre cuál de ellas desea solicitar los atributos. De este modo, indica la entidad para obtener sus atributos, de ellos elige sobre qué atributo desea consultar las métricas previamente definidas. Por ejemplo, recibida la entidad *paciente trasplantado ambulatorio*, selecciona el atributo *frecuencia cardíaca*, que retornará como métrica *valor de la frecuencia cardíaca*. Una vez mostradas las métricas para el atributo seleccionado, debe evaluar si la fuente de datos está en condiciones de implementarla, en base a la precisión y error exigido por la métrica versus la soportada por la fuente de datos. Si la fuente de datos posee la precisión y tolerancia de error

exigida, puede solicitar se registre la misma con la métrica indicada (por ejemplo, aquí se registraría el dispositivo GF-405GPS junto con la métrica *valor de la frecuencia cardíaca*). Ante tal petición, la proveedora de servicios de configuración, recepta el pedido desde la fuente de datos y evalúa por su cuenta, si la fuente puede implementar realmente las métricas solicitadas. Finalmente, retornará un arreglo con el estado de asociación de la fuente de datos para con cada métrica que intentó implementar. De las métricas implementadas, la fuente de datos *debe almacenar* a qué proyecto, entidad y atributo pertenecen las mismas, ya que en futuras operatorias (dentro de las que se incluyen disociar a la fuente de datos de un subconjunto de métricas dadas) se requerirá dicha información.

Complementariamente, en caso de que una fuente de datos desee desvincularse de las métricas que ha implementado, el procedimiento es simple. Se invoca a través de DSM el método *requestRemoveDSWithMetric,* pasándose como parámetro la propia fuente de datos más las métricas de las cuales se desea disociar. Eventualmente, si previo a la disociación, la fuente de datos requiere verificar si el proveedor de servicios de configuración la vinculó a las métricas, puede realizarlo a través de DSM mediante el método *isRegisteredDSWithMetric*.

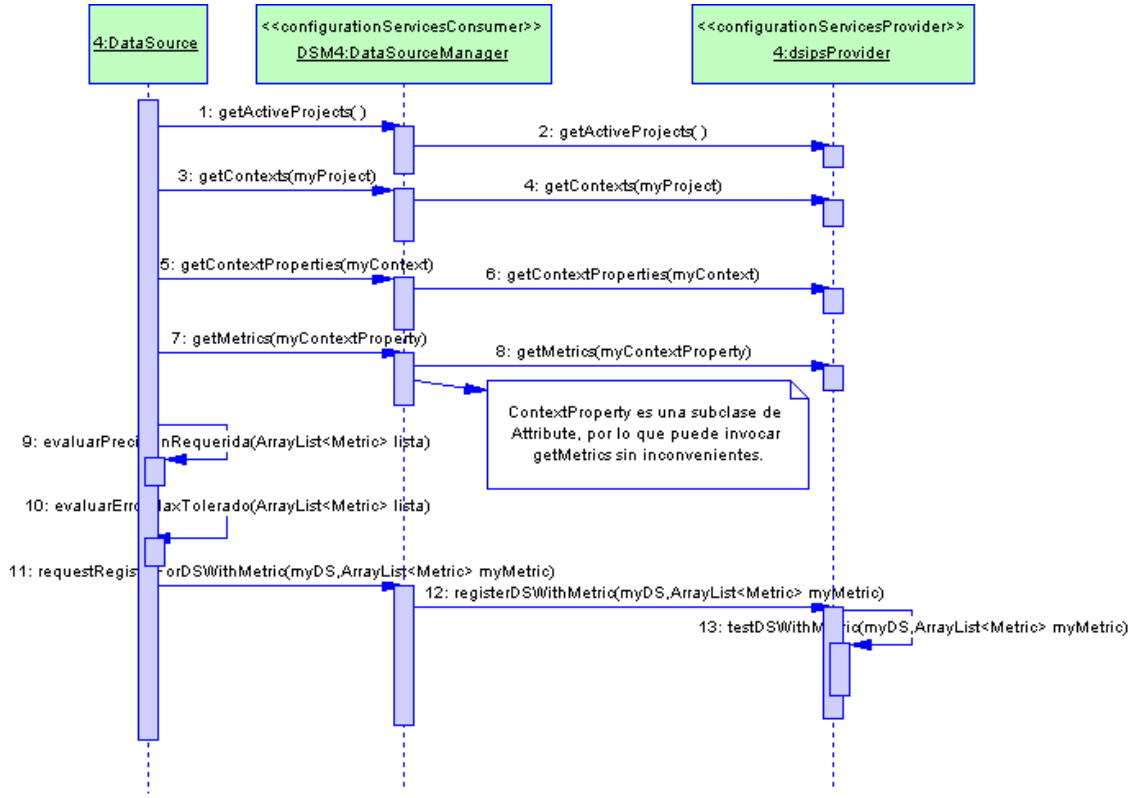


Figura 31. Diagrama de Secuencia de la Asociación de una Fuente de Datos a una Métrica asociada a una Propiedad de Contexto de un Contexto dado

En la Figura 31, puede observase el diagrama de secuencia asociado a la configuración de una fuente de datos, con métricas que se relacionan a propiedades de contexto de un contexto de medición definido para un proyecto de M&E determinado. El procedimiento típico, comienza

solicitando los proyectos activos (por ejemplo, *Monitoreo de Pacientes Trasplantados Ambulatorios*). De los proyectos activos, la fuente de datos a través de la interface de configuración seleccionará uno, a partir del cual solicitará se le remitan todos los contextos de medición definidos sobre el mismo (por ejemplo, el *contexto del paciente trasplantado ambulatorio*). Una vez recibidos los contextos, se debe escoger sobre cuál de ellos se desea solicitar las propiedades de contexto para poder realizar la petición (por ejemplo, para el contexto del paciente trasplantado ambulatorio, se retornará la *temperatura ambiental*, la *humedad ambiente*, la *presión ambiental* y el *posicionamiento del paciente*). A partir de la selección de una de las propiedades de contexto en particular, como por ejemplo el *posicionamiento del paciente*, se solicita que se informe qué métricas tiene asociadas dicha propiedad de contexto (por ejemplo, la métrica *valor del posicionamiento del paciente*). Recibidas las métricas, debe evaluar si la fuente de dato está en condiciones de implementarla, en base a la precisión y error exigido por la métrica versus la soportada por la fuente de datos. De estar en condiciones de implementarla, la fuente de datos a través de DSM, solicita se registre a la misma junto con las métricas indicadas (por ejemplo, aquí se registraría el dispositivo GF-405GPS junto con la métrica *valor del posicionamiento del paciente*). Ante tal petición, la proveedora de servicios de configuración recepta el pedido desde la fuente de datos, y evalúa por su cuenta si la fuente puede implementar realmente las métricas solicitadas. Finalmente, retornará un arreglo con el estado de asociación de la fuente de datos para con cada métrica que deseaba implementar. De las métricas implementadas, la fuente de datos *debe almacenar* a qué proyecto, contexto y propiedad de contexto pertenecen las mismas, ya que en futuras operatorias (dentro de las que se incluyen, disociar a la fuente de datos de un subconjunto de métricas dadas) se requerirá dicha información.

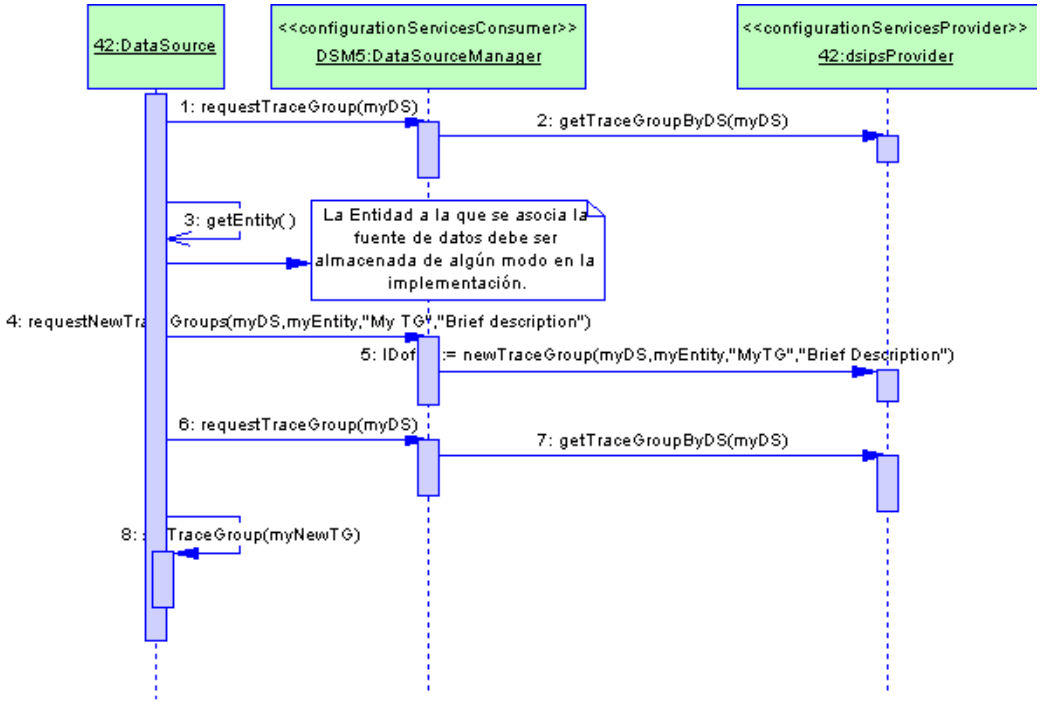


Figura 32. Diagrama de Secuencia para la Creación de un Grupo de Seguimiento

La Figura 32 muestra el procedimiento típico llevado adelante para la creación de un grupo de seguimiento (por ejemplo, “Juan Perez”), el cual es iniciado por una fuente de datos (por ejemplo, el GF-405GPS). La fuente de datos verifica inicialmente cuál es el grupo de seguimiento que tiene asociado, en caso de poseer uno, y luego la fuente de datos corrobora que tenga una entidad asociada sobre la cual esté implementando métricas, sea para atributos o para propiedades contextuales. En caso de tener una entidad asociada, puede solicitar la creación de un nuevo grupo de seguimiento (por ejemplo, “Juan Perez”), indicando sobre qué entidad se aplicará (por ejemplo, la entidad paciente trasplantado ambulatorio), qué fuente de datos solicita la creación (por ejemplo, el GF-405GPS), un nombre descriptivo del grupo más una breve descripción del mismo (por ejemplo, *Nombre*: “Juan Perez”, *Descripción*: Grupo de seguimiento para el paciente trasplantado ambulatorio Juan Perez). Tal petición, es efectuada a través del DSM y éste se encarga de traducirla en términos del proveedor de servicios, el cual verificará, que el grupo no exista previamente para luego, crear efectivamente el mismo. Una vez que el proveedor ha creado el grupo de seguimiento, retorna su identificador. Luego, la fuente de datos solicita se le informe, en forma actualizada, los grupos de seguimiento con los cuales se asocia y verifica efectivamente estar asociado al grupo creado, además de actualizar internamente los grupos de seguimiento con los que se vincula.

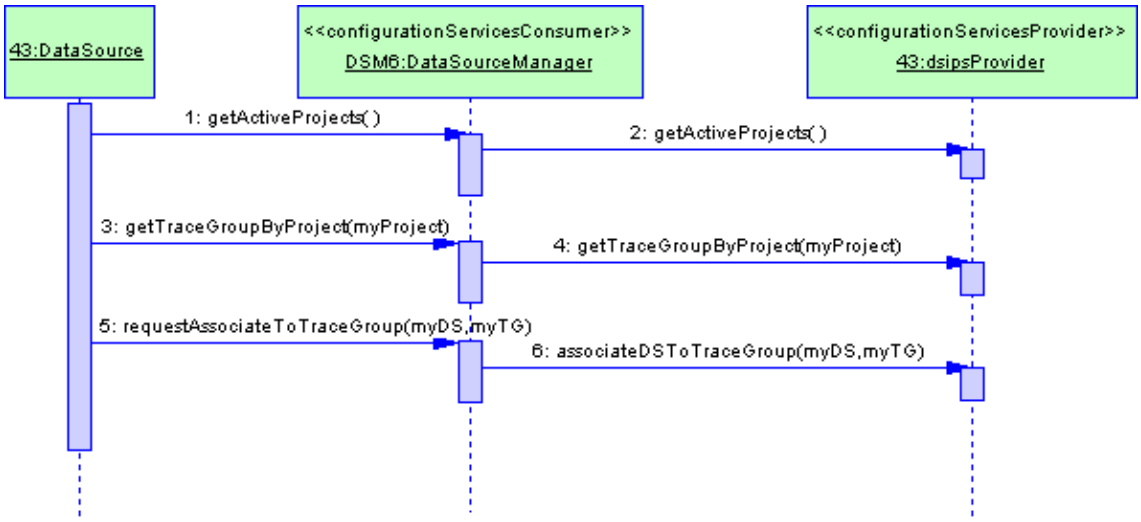


Figura 33. Diagrama de Secuencia para la Asociación de una Fuente de Datos a un Grupo de Seguimiento

Una fuente de datos puede asociarse a un grupo de seguimiento previamente creado por otra fuente de datos, tal y como ilustra el procedimiento de la Figura 33. El procedimiento comienza con la fuente de datos solicitando los proyectos activos. Seleccionado uno de los proyectos activos (por ejemplo, *Monitoreo de Pacientes Trasplantados Ambulatorios*), pide se le informe qué grupos de seguimiento tiene asociado dicho proyecto (por ejemplo, “Juan Perez”, “Luis García”, entre otros). De los grupos informados, la fuente de datos selecciona uno y mediante el método *requestAssociateToTraceGroup* de DSM solicita se vincule la fuente de datos a tal grupo.

Si una fuente de datos tuviese ‘n’ asociaciones a grupos de seguimiento, ello provocaría que cuando la medición arribe al proveedor de servicios de transmisión, como se profundizará en el capítulo siguiente, se genere una repetición de tal medición tantas veces como grupos de seguimiento asociados lo requieran dentro del buffer de mediciones. Esto último, es común en propiedades contextuales de pacientes situados en una misma región. Por ejemplo, si el paciente trasplantado “Juan Perez” está ubicado en el mismo barrio que “Luis García”, solo basta un dispositivo para medir la temperatura y humedad ambiente sobre uno de los pacientes (ver Figura 15), supongamos “Juan Perez”, e indicar que tales mediciones, sean también informadas al grupo de seguimiento de “Luis García” ya que corresponden a contextos análogos.

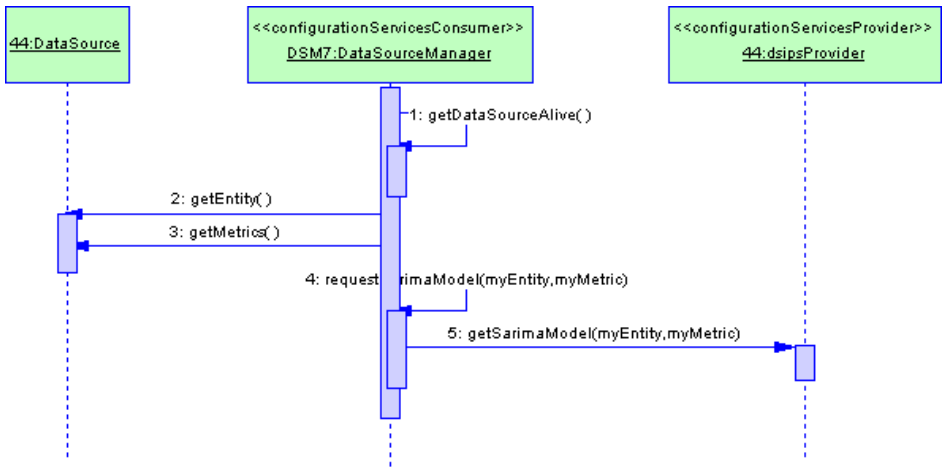


Figura 34. Diagrama de Secuencia Asociado a la Actualización de Modelos Predictivos

La actualización de modelos predictivos es administrada por DSM y provoca que el mismo recorra las fuentes de datos ‘vivas’ (ver Figura 34), le solicita que informen las métricas implementadas como así también a qué entidad se asocian. Efectuado el recorrido, se sintetiza dicha información para evitar redundancia, ya que dos fuentes de datos pueden implementar la misma métrica asociada a igual entidad. Una vez que posee un listado no redundante de métricas y entidades vinculadas, solicita al proveedor de servicios de configuración, le informe el modelo SARIMA con sus parámetros para cada par métrica-entidad. Es posible que no existan parámetros definidos para el modelo SARIMA de una métrica-entidad por motivos varios, por ejemplo, desde que no ha sido computado aún, hasta que la métrica no ajusta a los requerimientos de los modelos SARIMA para su aplicación.

6 Transmisión de Mediciones

El capítulo 5 ha discutido el modo en que las fuentes de datos heterogéneas son integradas al EIPFDcMM. Se han analizado los servicios de configuración intervinientes para enlazar la entidad bajo medición, sus atributos, las métricas definidas en un proyecto de medición y evaluación y los sensores/dispositivos utilizados para obtener las medidas. Adicionalmente, se ha discutido el gestor de fuentes de datos y su rol en la recolección de mediciones a partir de las diferentes fuentes de datos. Se ha analizado el rol de los modelos predictivos SARIMA dentro de la recolección de mediciones como así también, se han discutido los procesos de configuración típicos entre las fuentes de datos y el proveedor de servicios de configuración.

En este capítulo, se analiza el esquema XML (e**X**tensible **M**arkup **L**anguage) para el intercambio de mediciones sustentado en la base conceptual de C-INCAMI, denominado C-INCAMI/MIS (**M**easurement **I**nterchange **S**chema). Luego, se discute las interfaces de transmisión de mediciones y su relación con el procesador C-INCAMI/MIS, el cual permite la conversión entre el esquema C-INCAMI/MIS y los datos ralos desde las fuentes de datos.

A seguir, se analiza el adaptador de mediciones y su rol ante la recolección y transmisión de mediciones. Luego, se discute la gestión de mediciones dentro de un buffer multinivel dentro de la función de reunión, a los efectos de la organización de las medidas para su posterior análisis.

Finalmente, se discuten los mecanismos de load shedding y su incorporación en el buffer multinivel de mediciones, junto con el mecanismo de recepción de mediciones implementado desde la óptica de la función de reunión.

6.1 Esquema C-INCAMI / MIS

El gestor de fuentes de datos recolecta las mediciones desde las fuentes de datos en forma rala, es decir, la medida pura sin metadato alguno. A través de la configuración previa y mediante la interface Datasource (ver Figura 22), la cual debe implementar el dispositivo de medición, el gestor de fuente de datos tiene conocimiento de la asociación de la medida para con la métrica, el atributo y la entidad a la que corresponde. Ello permite, que dichas medidas puedan ser informadas conjuntamente con los metadatos asociados a los conceptos que están midiendo. Para efectivamente informarlos, se requiere de una forma específica de estructurar los datos y sus metadatos, para poder diferenciar claramente cada uno de los conceptos y sus valores. En este último aspecto, es en donde es central el rol de C-INCAMI/MIS con respecto al EIPFDcMM (Diván & Olsina, 2009b; Diván, Olsina, et al., 2011b).

El esquema utilizado para el intercambio de mediciones se denomina CINCAMI/MIS. El mismo, se sustenta en la base conceptual de C-INCAMI, con el objeto de incorporar en forma conjunta a las mediciones, metadatos que permitan a posteriori, hacer más robusto y consistente el análisis de los datos.

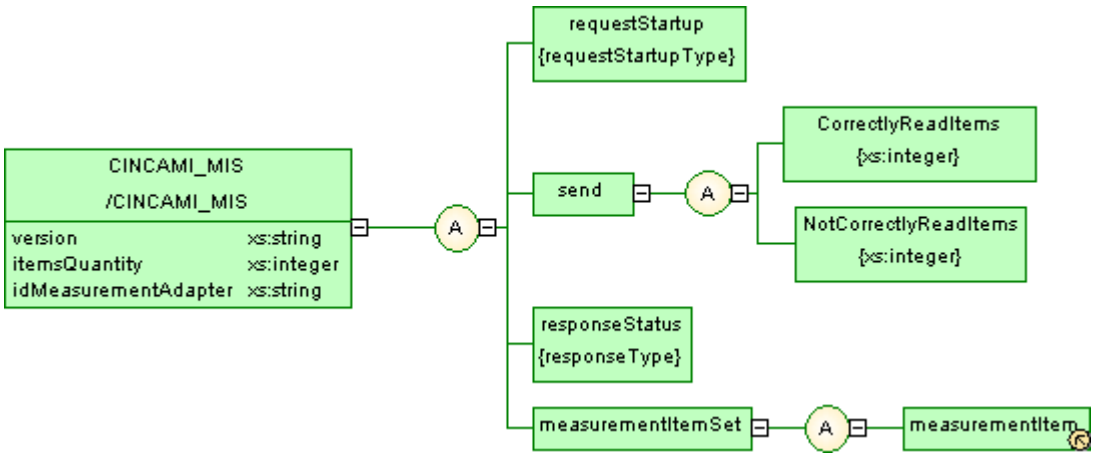


Figura 35. Nivel Superior del Esquema C-INCAMI/MIS.

La etiqueta principal del esquema es *CINCAMI_MIS* (ver Figura 35), y tiene asociada los atributos *version,* el cual señala la versión del esquema, *itemsQuantity* el cual indica la cantidad de unidades lógicas de medición a transmitir (ver sub-sección 4.3.3) y *idMeasurementAdapter* que es el identificador único del adaptador de mediciones del cual proviene el mensaje, el cual se discutirá en la sub-sección 6.4.

La letra "A" dentro del círculo en la Figura 35, que une a la etiqueta principal *CINAMI_MIS* con sus cuatro sub-etiquetas, representa que el orden en que se informen las cuatro sub-etiquetas no está definido, por lo que al momento de serializarse el mensaje es indistinto si primero se informa una que otra. Las cuatro sub-etiquetas que componen *CINCAMI/MIS* son opcionales. Esto implica, que el mismo esquema puede ser empleado para mensajes de control como para la transmisión de mediciones propiamente dichas. Dentro de las subetiquetas, *requestStartup* es una etiqueta opcional que representa un mensaje de control, la cual puede asumir solo los valores

startup o *reverse. Startup* puede enviarse previo al envío de mediciones, para conocer el estado del proveedor del servicio de transmisión, el cual será discutido en la próxima sub-sección. *Reverse* es un mensaje de control que se envía porque ha ocurrido un timeout durante un mensaje anterior y no se posee confirmación de estado del receptor. La sub-etiqueta *responseStatus* , se vincula solo con respuestas del proveedor e indica el estado del mismo, pudiendo solo asumir los valores *online* u *offline*. La sub-etiqueta *send* se asocia con mensajes de respuesta e indica cuantas unidades lógicas de medición han podido ser leídas correctamente y cuantas han sido descartadas. Finalmente, la sub-etiqueta *measurementItemSet* es la etiqueta que engloba las unidades lógicas de medición, agrupando un conjunto de sub-etiquetas *measurementItem* (ver Figura 36).

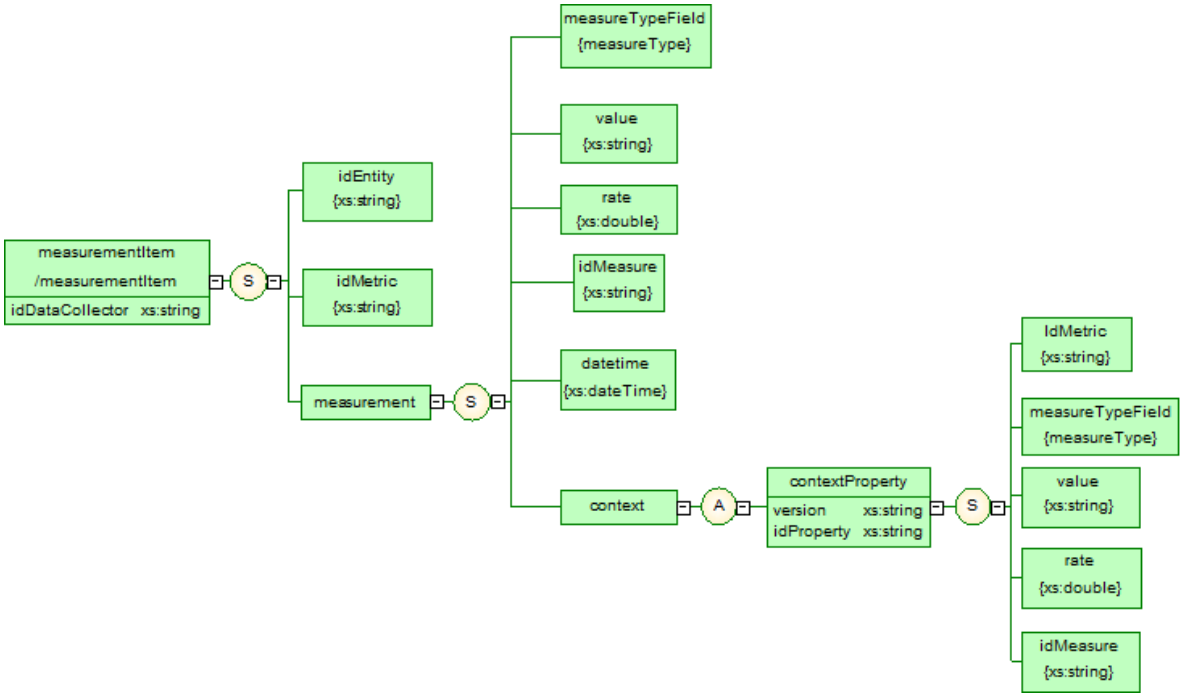


Figura 36. Composición de la Etiqueta *measurementItem* en CINCAMI/MIS

La letra "S" dentro del círculo en la Figura 36, que une a la etiqueta *measurementItem* con sus tres sub-etiquetas, representa que el orden en que se informen las mismas está definido y es secuencial, por lo que al momento de serializarse el mensaje primero se informará *idEntity*, luego *idMetric* y finalmente *measurement*. La letra "A" dentro del círculo en la Figura 36, posee el mismo significado que el indicado para la Figura 35. La etiqueta *measurementItemSet* posee un único atributo, *idDataCollector*. Este atributo es de singular importancia dado que informa el identificador de la fuente de datos según la unidad lógica de medición. De este modo, el esquema es apto para informar a través de un único mensaje, mediciones provenientes de diferentes fuentes de datos. La etiqueta *measurementItem*, se compone de tres sub-etiquetas obligatorias: a) La sub-etiqueta *idEntity,* representa el identificador de la entidad con la que se asocia la métrica, b) La sub-etiqueta *idMetric,* indica el identificador de la métrica sobre la que se está informando una medición c) La sub-etiqueta *measurement*, representa la medición propiamente dicha.

La etiqueta *measurement,* brinda la posibilidad de informar en forma conjunta la medición con respecto a las propiedades de contexto. Las sub-etiquetas obligatorias son *measureTypeField, value y datetime.* La sub-etiqueta *measureTypeField* puede asumir sólo los valores: *actual* y *estimated,* los cuales indican si la medición fue determinista o no determinista (probabilística) respectivamente. La sub-etiqueta *value* representa un valor concreto, que se asume determinista o no en función del valor que asuma *measureTypeField*. La etiqueta *datetime* informa fecha y hora en que se tomó la medición en la fuente de datos.

En caso de que el valor asumido por *measureTypeField* fuese *estimated*, se informarán las subetiquetas *rate* y *idMeasure.* La subetiqueta *rate* indica la probabilidad asociada al valor indicado en *value,* mientras que *idMeasure* permite agrupar la distribución de probabilidad entre diferentes etiquetas *measurement,* considerándolas fruto de una misma medición y no como mediciones separadas.

La etiqueta *context* (ver Figura 36) es opcional ya que no existe obligación que todo proyecto deba tener asociado un contexto, con lo cual puede ocurrir que deban informarse mediciones y las mismas no tener contexto asociado. En caso de poseer contexto asociado, *context* se compone de un conjunto de etiquetas *contextProperty.* Esta última etiqueta contiene un único atributo denominado *version,* el cual representa la versión de propiedad de contexto empleada. Cada etiqueta *contextProperty* representa una medición efectuada sobre una propiedad de contexto y tendrá como sub-etiquetas obligatorias a *idProperty*, *measureTypeField* y *value*. Además, *idProperty* es el identificador de la propiedad de contexto con la que se asocia, *measureTypeField* indica el tipo de valor informado con la propiedad de contexto *y value* especifica un valor dado. La sub-etiqueta *measureTypeField* puede solo asumir *actual* o *estimated* como valor. Si asumiese el valor *actual* entonces el valor de la sub-etiqueta *value* es determinista, pero si por el contrario fuese *estimated*, deberán informarse las sub-etiquetas *rate* y *idMeasure.* La sub-etiqueta *rate* indica la probabilidad asociada al valor indicado en *value,* mientras que *idMeasure* permite agrupar la distribución de probabilidad entre diferentes etiquetas *contextProperty,* considerándolas fruto de una misma medición y no como mediciones separadas.

A los efectos de esquematizar el empleo de C-INCAMI/MIS en base al caso de aplicación presentado en la sub-sección 3.5, supongamos que:

a) El adaptador de mediciones, situado en el Smartphone Palm Treo 750 del caso de aplicación, posee como identificador 12345 (*idMeasurementAdapter*)
b) Solo se transmite 1 medición determinista junto con una propiedad contextual no determinista
c) La entidad *paciente trasplantado ambulatorio* posee el identificador 1 (*idEntity*)
d) Se informa solo el atributo frecuencia cardíaca de la entidad *paciente trasplantado ambulatorio*. Su métrica asociada, *valor de la frecuencia cardíaca*, posee el identificador 2 (*idMetric*)
e) La fuente de datos que mide la frecuencia cardíaca, el GF-405GPS (ver Figura 15), posee el identificador 5 (*idDataCollector*)

f) Solo se considera una propiedad contextual para facilitar la lectura del XML comentado

g) La propiedad contextual a considerar será la humedad ambiente, cuyo identificador es 5 (*idProperty*)

h) El termómetro digital, presentado en el caso de aplicación (ver Figura 15), mide temperatura y humedad ambiente. Vamos a suponer que el mismo informa valores probabilísticos con respecto al valor de la humedad para poder visualizar el modo en que se transmitirían las mediciones

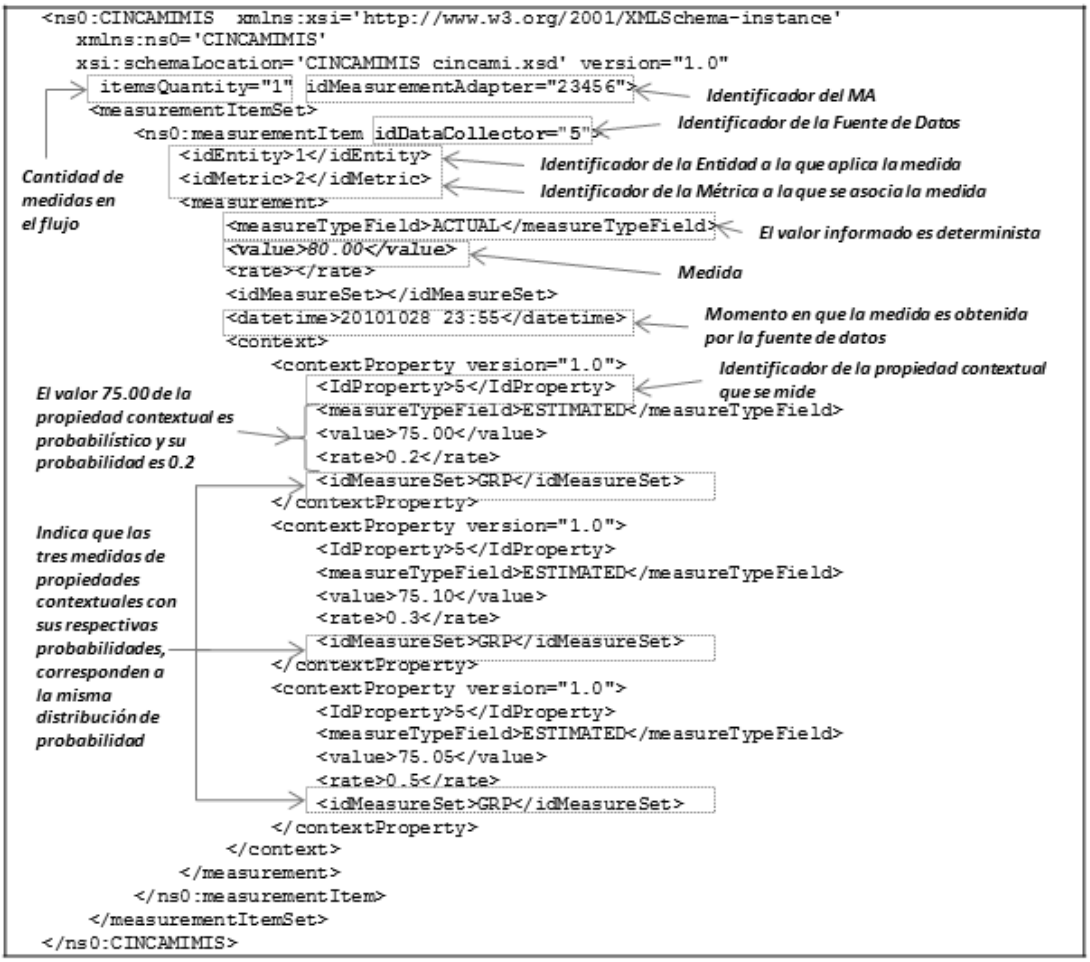


Figura 37. Esquema del flujo C-INCAMI/MIS comentado

La implicancia de los metadatos en el flujo de mediciones es relevante en nuestro trabajo, tal y como puede observarse en la Figura 37. De este modo, cada vez que se transmiten mediciones, se incorpora al dato en bruto –el valor 80 del ejemplo- la estructura del esquema C-INCAMI/MIS indicando a qué corresponde cada medida informada. Por ejemplo, en el mensaje mostrado en la Figura 37, IDEntity=1 representa a la entidad *paciente trasplantado ambulatorio*, IDMetric=2 a la métrica *valor de la frecuencia cardíaca* e IDProperty=5 a la métrica *valor del porcentaje de humedad ambiente* en el sitio donde se encuentra el paciente. Dicho mensaje, claramente

incorpora a través de sus metadatos, una serie de información que permiten guardar relación con el origen del dato, identificar la fuente de datos y su MA informante, el momento en que se obtiene la medida como la situación de las propiedades de contexto, a qué métrica y entidad se asocia una medida y el tipo de valor obtenido. Estos son algunos de los aspectos fundamentales que se desprenden de los metadatos del mensaje, los que permiten luego incrementar la consistencia en el modelo de procesamiento a partir del soporte ontológico de las definiciones. En efecto, supongamos por el contrario, que en un instante de tiempo, se recibe una medida asociada con la frecuencia cardíaca de un paciente trasplantado ambulatorio que indica sólo 80, la cuestión es: ¿Qué representa?, ¿Qué unidad de medida posee?, ¿Es bueno o malo?, ¿Qué es bueno?, ¿Qué es malo?, ¿En qué escala se expresa?, ¿Cualquier software puede procesar la medida? Si no fuese por la incorporación de los metadatos en el flujo y la base conceptual ontológica de C-INCAMI, estas preguntas no podrían ser resueltas en forma consistente, lo que incrementaría la complejidad de las aplicaciones asociadas derivando posiblemente, en la incorporación de la lógica de procesamiento en las mismas, haciéndolas demasiado específicas a cada contexto, complicando la reutilización y las tareas de mantenimiento.

6.2 Interface de Transmisión

Una vez que las diferentes fuentes de datos han sido configuradas (por ejemplo, el dispositivo GF-405GPS de la Figura 15, ha indicado que implementará las métricas *valor de la frecuencia cardíaca*, asociada al atributo *frecuencia cardíaca* de la entidad *paciente trasplantado ambulatorio*, y el *valor de posicionamiento del paciente*, asociado a la propiedad de contexto *posición del paciente* del *contexto asociado al paciente trasplantado ambulatorio*), es necesario definir cómo se procederá a los efectos de que el gestor de fuentes de datos pueda materializar la transmisión de las mediciones y éstas sean receptadas dentro del buffer multinivel (a discutir en la sub-sección 6.5). Con dicha finalidad, se plantea una interface base denominada *TransmitionService* y a partir de ella, se especializa en las interfaces *TransmitionServiceConsumer* y *TransmitionServiceProvider* según si el rol a desempeñar sea consumidor o proveedor del servicio respectivamente (ver Figura 38).

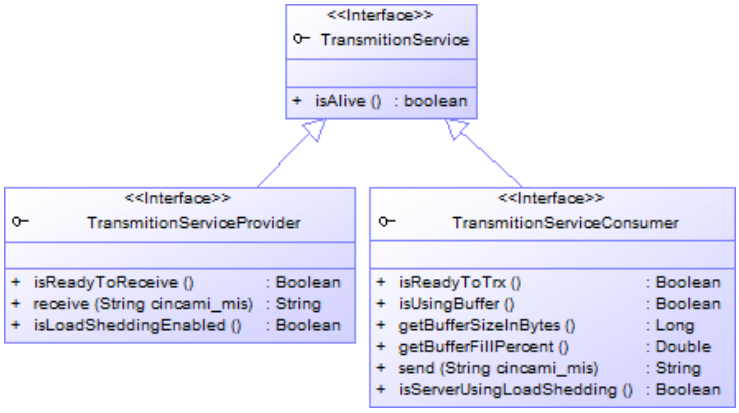


Figura 38. Interfaces Asociadas a la Transmisión de Mediciones

La interface *TransmitionService* incorpora los métodos comunes tanto al consumidor como al proveedor de servicios de transmisión. De este modo, el único método compartido es *isAlive*, el cual indica si el consumidor se encuentra activo a los efectos de realizar futuras transmisiones y, desde la óptica del proveedor, si éste está activo y escuchando el medio por donde se receptan las mediciones.

La interface *TransmitionServiceConsumer* representa la interface que *debe* ser implementada por quien desee transmitir las mediciones recolectadas desde diferentes fuentes de datos hacia la función de reunión (ver sub-sección 3.2). A continuación, se discuten los principales métodos asociados a dicha interface.

Los métodos *isUsingBuffer*, *getBufferSizeInBytes* y *getBufferFillPercent* permiten consultar sobre si quien implementa la interface de consumidor de servicios de transmisión, implementa un buffer para almacenar localmente las mediciones, en forma previa a su transmisión. De estar implementado el buffer, los restantes métodos permiten consultar sobre el tamaño en bytes ocupado por el mismo como así también la proporción relativa de llenado.

El método *isReadyToTrx* indica que existen mediciones que pueden ser informadas y el consumidor del servicio se encuentra en condiciones de transmitirlas. El método isServerUsingLoadShedding permite consultar si el proveedor de servicios de transmisión, esto es quien implementa la funcionalidad de la función de reunión analizada en la sub-sección 3.2, está empleando mecanismos de descarte selectivo ante la situación en que ocurriese un desborde de la tasa de procesamiento del servidor.

El método *send* es quien transmite las mediciones a la función de reunión. Dicho método, recibe una cadena de caracteres que corresponde a un documento bajo el esquema C-INCAMI/MIS, por lo que la interface tendrá dependencia directa con respecto al responsable de convertir las mediciones ralas a formato C-INCAMI/MIS y viceversa. Dicha responsabilidad recae sobre el denominado procesador C-INCAMI/MIS, el cual es abordado en la sub-sección 6.3.

La interface *TransmitionServiceProvider* representa la interface que *debe* ser implementada por quien receptará las mediciones recolectadas desde diferentes fuentes de datos, por ende, la misma corresponde a la denominada función de reunión (ver sub-sección 3.2). A continuación, se discuten los principales métodos asociados a dicha interface.

El método *isReadyToReceive* indica si el proveedor de servicios de transmisión (unidad de procesamiento o función de reunión) se encuentra disponible para iniciar la recepción de las mediciones desde el adaptador de mediciones, el cual será analizado en la sub-sección 6.4.

El método *isLoadSheddingEnabled* indica si el proveedor de servicios está empleando en ese preciso momento alguna política de descarte de mediciones. El descarte selectivo o load shedding, como sus políticas asociadas, serán discutidos en la sub-sección 6.6.

El método *receive* recibe una cadena de caracteres que corresponde a un documento bajo el esquema C-INCAMI/MIS para su deserialización y procesamiento de las mediciones. De este

modo, al igual que el método *send* del consumidor de servicios, el método receive depende directamente del procesador C-INCAMI/MIS.

6.3 Procesador C-INCAMI / MIS

La clase *CINCAMIMISProcessor* implementa la funcionalidad del procesador de mediciones, el cual traduce flujos XML bajo el esquema C - INCAMI / MIS a objetos y viceversa. Previo a ingresar en el análisis funcional de dicha clase, es prioritario presentar las clases y relaciones que se emplean para efectuar el mapeo. Dichos objetos se sustentan en la base conceptual de C-INCAMI, ya que responden al esquema C-INCAMI / MIS para efectuar la traducción. El proceso de traducción desde esquema a objetos, se ha implementado mediante JAXB (Java Architecture for XML Binding) y su conceptualización del funcionamiento puede apreciarse en la Figura 39.

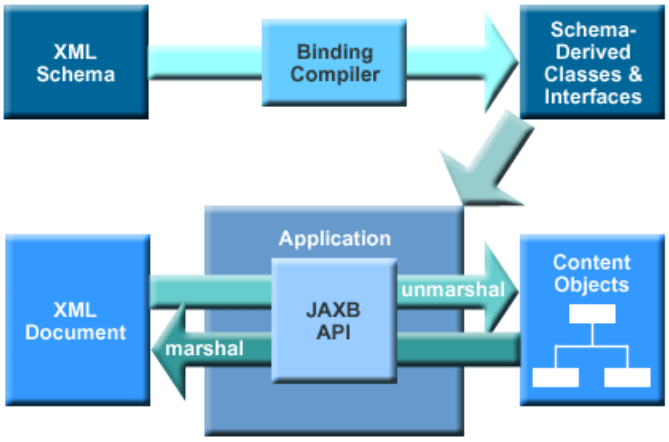


Figura 39. Conceptualización del Funcionamiento de JAXB (Ort & Mehta, 2003)

Conceptualmente, JAXB genera clases e interfaces derivadas a partir de un esquema XML validado. Luego, a partir de dichas clases e interfaces, es posible enlazar los objetos a aquellos documentos XML que respeten el esquema indicado, como así también, a partir de documentos XML basados en el esquema, es posible obtener el conjunto de objetos y sus relaciones, pudiendo efectuar la traducción en ambos sentidos de un modo transparente. De este modo y partiendo del esquema C-INCAMI/MIS, el conjunto de clases derivadas empleado JAXB, es expuesto en la Figura 40.

El diagrama de clases asociado a las clases derivadas de C-INCAMI/MIS (ver Figura 40), omite los métodos para facilitar la lectura del mismo. Puede observarse, comparado con C-INCAMI, que las clases expuestas coinciden y respetan las clases de la base conceptual de C-INCAMI, analizado en 2.2.3 y 4.3. Debe mencionarse a los efectos de evitar incurrir en inconsistencia, que dichas clases son solo empleadas para el mapeo entre XML y sus correspondientes objetos y relaciones, pero no son bajo ningún punto de vista persistentes.

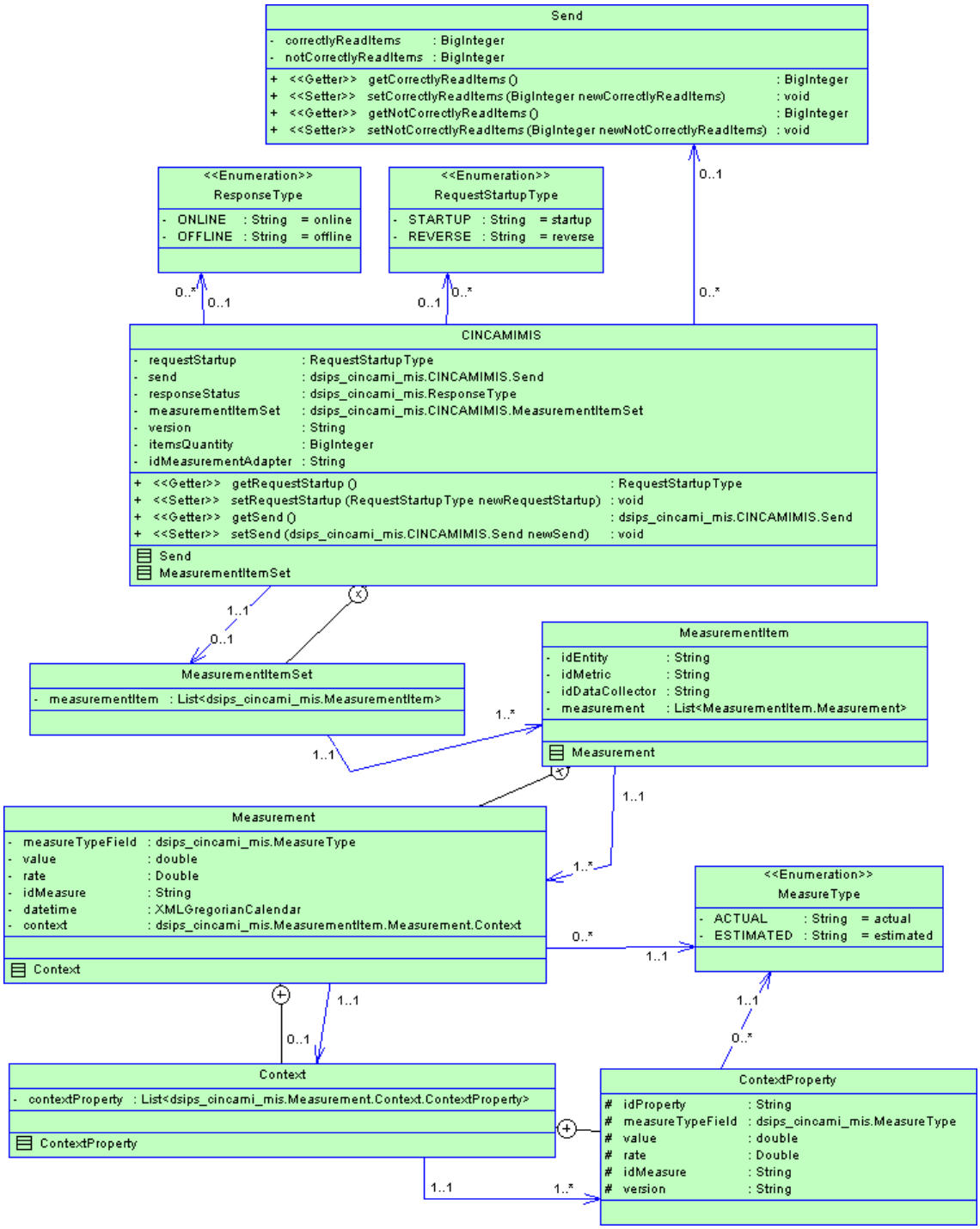


Figura 40. Clases Derivadas del Esquema C-INCAMI/MIS

Hay algunas cuestiones interesantes de mencionar, a los efectos de resaltar cómo la incorporación de metadatos permite incrementar la consistencia en las mediciones:

- La clase *CINCAMIMIS* representa el encabezado del mensaje y ella contiene toda información interna asociada a dicho mensaje. De este modo, cada objeto CINCAMIMIS

representa una ventana de procesamiento en términos de flujos de datos (data stream). Por ejemplo y continuando con el ejemplo de la Figura 37, la clase a través del encabezado del mensaje, sabría entre otras cuestiones, que las mediciones provienen del adaptador de mediciones 23456 (*idMeasurementAdapter*) asociado al Smartphone Palm Treo 750 del caso de aplicación.

- Si se analiza cada una de las relaciones de la clase *CINCAMIMIS*, podrá observarse que en ningún caso presenta obligatoriedad de existencia, por lo que se traduce esto en la aptitud de administrar mensajes de control y mediciones en forma totalmente transparente.

- La clase *MeasurementItem*, en lugar de tener un valor directamente, posee un conjunto de objetos del tipo *Measurement*. Ello explicita su aptitud para gestionar mediciones deterministas o no deterministas (probabilísticas). Esto último, fue ejemplificado en la Figura 37 mediante la métrica *valor de la frecuencia cardíaca* (cuyo idMetric es 2), para mediciones deterministas.

- Cada objeto del tipo *Measurement*, es capaz de informar opcionalmente su situación contextual, con lo que no se obliga a presentar información de contexto si una medición no la posee. No obstante, en caso de estar disponible, la propia medición es la que informa su situación de contexto, lo que en términos de procesamiento estadístico, permite efectuar un análisis conjunto de las mediciones y su contexto global o bien, permite analizar las mediciones de una métrica dada en su contexto particular. Esto último, también fue ejemplificado en la Figura 37, mediante la propiedad de contexto *valor del porcentaje de la humedad ambiente* (cuyo idProperty es 5), para mediciones no deterministas.

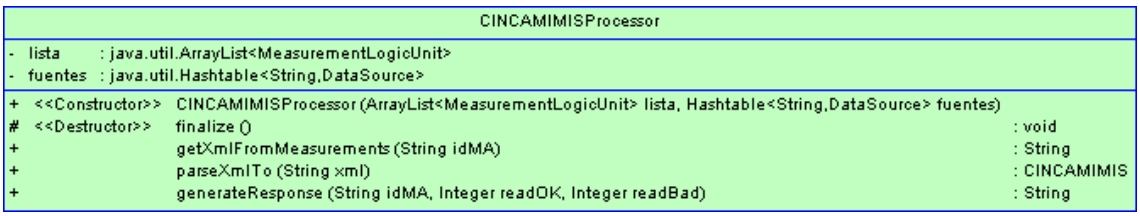


Figura 41. Clase CINCAMIMISProcessor

La clase CINCAMIMISProcessor (ver Figura 41) asume la responsabilidad de convertir un conjunto de mediciones ralas en un documento XML en base al esquema C-INCAMI/MIS. De igual modo, es responsable de la conversión de un documento XML bajo el mismo esquema, hacia un conjunto de objetos derivados del esquema, para el posterior procesamiento de las mediciones.

EL procesador C-INCAMI / MIS (CINCAMIMISProcessor), es instanciado solo cuando se desea convertir las unidades lógicas de medición a XML. En tal caso, se debe inicializar la instancia informando la lista de mediciones a transmitir como así también el conjunto de fuentes de datos que han intervenido en su generación. Por otro lado, la obtención de los objetos y relaciones a

partir de un documento XML bajo el esquema de intercambio de mediciones, está asociado con métodos estáticos de la clase, con lo que no se requiere instanciación alguna del procesador.

De este modo, el único momento en que se instancia el procesador es cuando quien implementa los servicios de consumo de transmisión (*transmisionServiceConsumer*, ver sub-sección 6.2) desea informar un conjunto de mediciones a la función de reunión. Así, el consumidor de servicios de transmisión empleará el constructor del procesador y el método *getXmlFromMeasurements,* para obtener el documento XML en base al esquema C-INCAMI/MIS. Por otro lado, la función de reunión, empleará los métodos estáticos *parseXmlTo* y *generateResponse* para obtener a partir del documento XML el conjunto de objetos y sus relaciones, como así también generar una respuesta al consumidor de servicios de transmisión respectivamente.

6.4 Adaptador de Mediciones

La clase *MeasurementAdapter* implementa la funcionalidad del Adaptador de Mediciones (Measurement Adapter -MA-) (ver sub-sección 3.2), cuyo objeto es administrar al gestor de fuentes de datos y a partir de las mediciones recolectadas por éste, es responsable de convertir las mismas al formato adecuado para concretar su transmisión.

De este modo, el MA implementa la interface *TransmitionServiceConsumer* (ver Figura 42) para poder llevar adelante la tarea de transmisión de las mediciones. Adicionalmente, presenta dependencia de las clases *DataSourceManager* y *CINCAMIMISProcessor.* La clase *DataSourceManager* es responsable de la gestión y configuración de las diferentes fuentes de datos (ver sub-sección 5.3), y la clase CINCAMIMISProcessor (ver sub-sección 6.3), es responsable de implementar el mapeo entre los documentos XML estructurados bajo el esquema C-INCAMI / MIS y los objetos derivados del mismo.

Si bien *DataSourceManager* es el responsable de llevar adelante la recolección de mediciones, ante un mecanismo de recolección por arrastre (POP), no es quien determinará el momento en el cual recolectar las mismas, sino que dicha orden vendrá dada por el propio Adaptador de Mediciones. De este modo, el MA se constituye en una clase activa, dado que es posible que varios hilos dentro del mismo puedan coordinar la recolección a partir del DSM.

El MA, implementa un buffer local mediante una tabla de dispersión, donde la función de dispersión es una función del ID de la fuente de datos. De este modo el buffer puede ser visto como un buffer jerárquico con dos niveles (ver Figura 43): a) El primer nivel permite el acceso al buffer particular de la fuente datos (por ejemplo, seleccionar entre los buffer del GF-405GPS, el termómetro digital, el Fischer Series 3312, entre otros expuestos en la Figura 15), mientras que b) el segundo nivel es el buffer particular a una fuente de datos en sí misma (por ejemplo, el buffer de mediciones exclusivamente del GF-405GPS). El buffer particular de cada fuente de datos, el segundo nivel, es implementado mediante una lista ordenada con las restricciones de una cola.

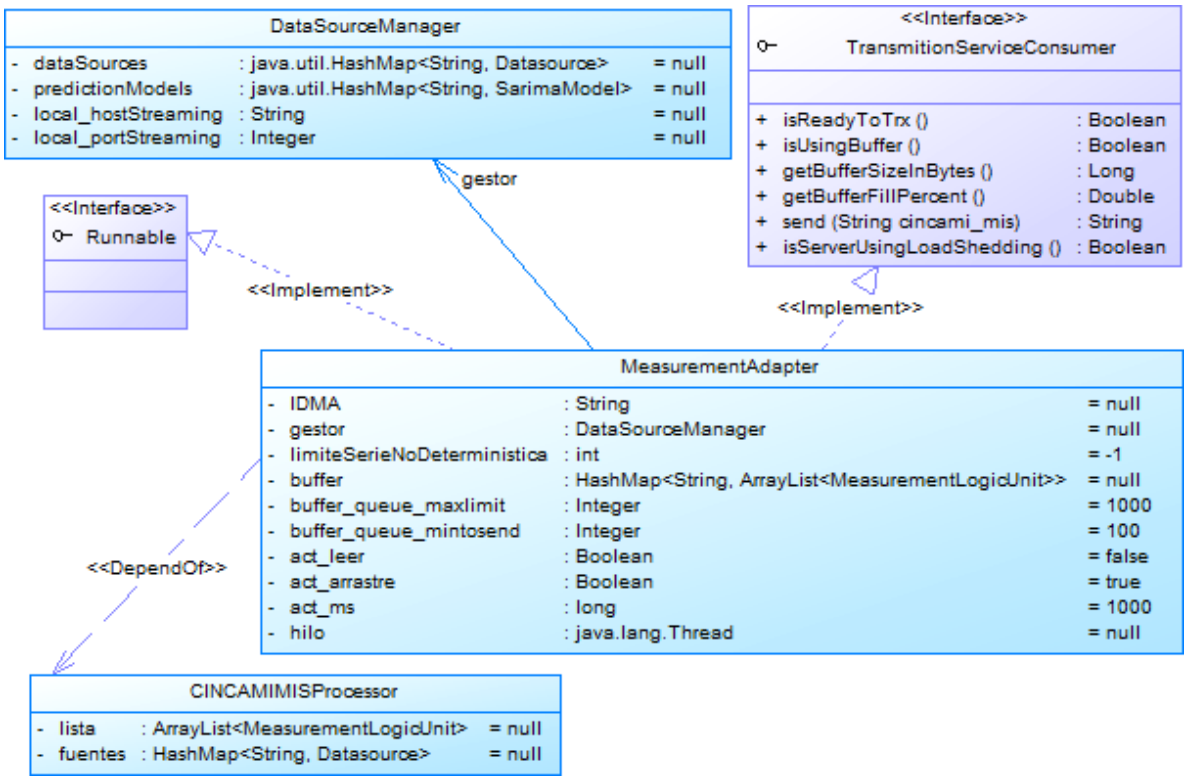


Figura 42. Principales Dependencias de la Clase *MeasurementAdapter*

Una vez dispersado el ID de la fuente de datos, se obtiene la raíz de la lista asociada con esa fuente de datos para poder acceder al buffer particular de la misma, en ella sólo existen mediciones de dicha fuente.

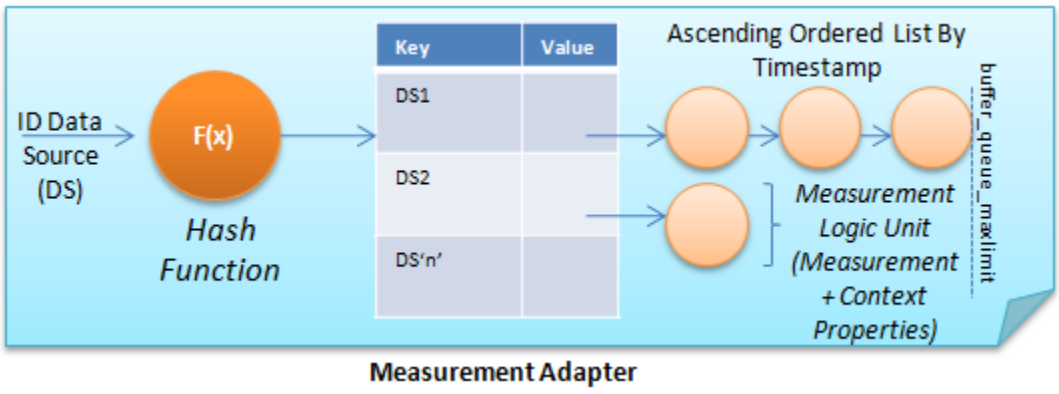


Figura 43. Conceptualización del Buffer Jerárquico del Adaptador de Mediciones

La clase *MeasurementAdapter* (ver Figura 44) posee dos propiedades de particular importancia para la gestión del buffer, *buffer_queue_maxlimit* y *buffer_queue_mintosend.* La propiedad *buffer_queue_maxlimit* representa el límite máximo de mediciones que es posible almacenar para el buffer de una fuente de datos mientras que *buffer_queue_mintosend,* determina la cantidad de mediciones mínimas a transmitir por cada fuente de datos. De este

modo, estas propiedades administran el volumen de mediciones a transmitir por el adaptador de mediciones a la función de reunión.

Si un buffer particular estuviese en el límite indicado por *buffer_queue_maxlimit,* al tratar de ingresar una nueva medición primero se redirecciona la raíz de la lista al segundo elemento, se descarta el primero y finalmente se incorpora la medición. Así, ante un potencial exceso del buffer local, *siempre* se descartan las mediciones más viejas no transmitidas, como se ha esquematizado en la Figura 43.

Adicionalmente, es posible adaptar dinámicamente el parámetro *buffer_queue_maxlimit.* De incrementarse el mismo, no existirá inconveniente en cuanto a las mediciones no transmitidas salvo por la verificación previa de suficiencia de memoria. No obstante, de decrementarse, provocará que el adaptador de mediciones efectúe una revisión de cada buffer particular para cada fuente de datos, de modo de analizar si la cola excede o no el nuevo límite. De exceder la cola al nuevo límite, se descartarán tantas mediciones viejas como sea necesario hasta sincronizar su tamaño.

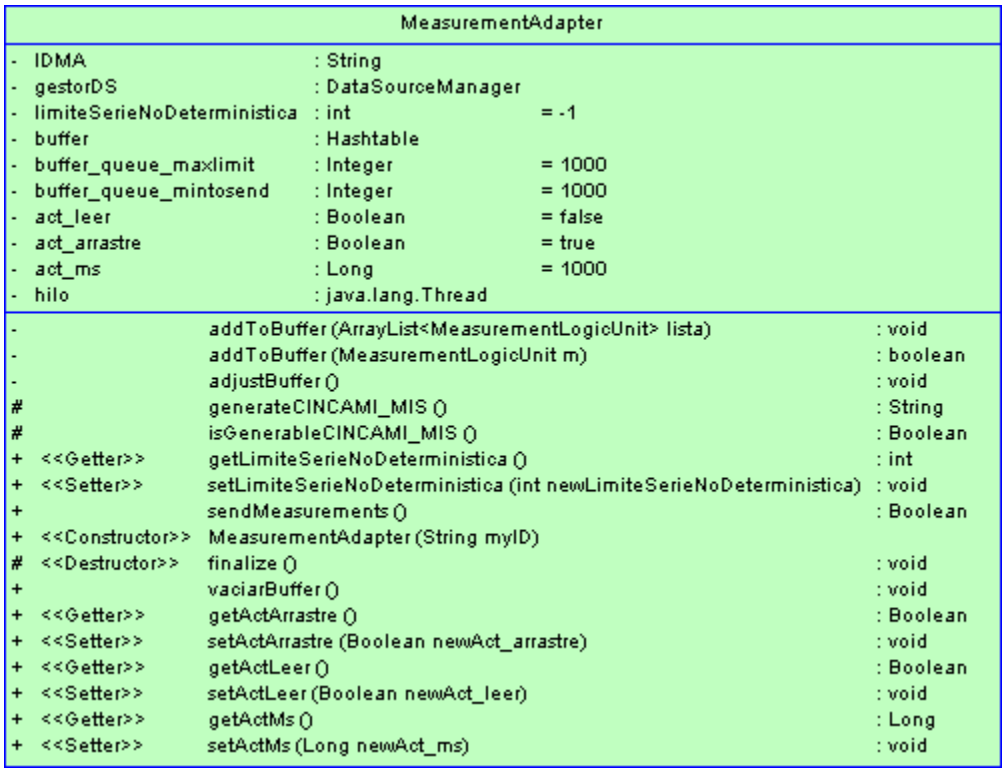


Figura 44. Clase MeasurementAdapter

La Tabla 4, analiza cada uno de los atributos de la clase MeasurementAdapter, dado que en base a ellos, se coordina la transmisión de las mediciones provenientes desde las fuentes de datos.

Tipo	Propiedad	Significado
String	IDMA	Identificador Global del Adaptador de Mediciones
DataSourceManager	gestorDS	Gestor de las fuentes de datos a través de la cual es posible configuración y gestión.
Integer	limiteSerieNoDeterministica	Límite de valores para series no deterministas. Si una fuente de datos arroja una serie de valores no deterministas 'n', estos serán ajustados a los primeros '*limiteSerieNoDeterministica*'. Un valor -1 indica que no existe límite establecido por lo que se receptará la serie completa como válida.
Hashtable	Buffer	Tabla de dispersión cuya clave de acceso es el ID de la fuente de datos, mediante la cual se accede a la lista de las mediciones más recientes. El tamaño de dicha lista está regido por buffer_queue_maxlimit.
Integer	buffer_queue_maxlimit	Límite máximo de la lista de mediciones para cada fuente de datos en el buffer.
Integer	buffer_queue_mintosend	Cantidad mínima de mediciones a transmitir a la función de reunión. Debe ser menor o igual a buffer_queue_maxlimit. El límite aplica a cada cola, no confundir con cantidad de mediciones globales del buffer.
Boolean	act_leer	Indica si el MA leerá continuamente mediante mecanismo de arrastre (POP) o empuje (PUSH). Si act_arrastre está activado, entonces solicitará las mediciones a las fuentes de datos y las incorporará en el buffer (POP), caso contrario la responsabilidad de llenar el buffer es de las fuentes. Mientras act_leer sea TRUE, se verificará si existen mediciones que transmitir, indistintamente si es por arrastre o empuje la recolección. De existir las mismas, se transmiten para luego vaciar el buffer solo de las mediciones enviadas.
Boolean	act_arrastre	Variable de control de la interfaz java.lang.Runnable de JAVA, que determina si culminar o no el ciclo de recolección de mediciones por arrastre. Si es FALSE el ciclo continúa, mientras act_leer sea TRUE, esperando mediciones en el buffer que sean informadas por las fuentes para su transmisión.
Long	act_ms	Tiempo en milisegundos cada el cual se producirá la lectura de mediciones mediante arrastre.
java.lang.Thread	Hilo	Hilo de control para lectura de mediciones

Tabla 4. Propiedades de la Clase MeasurementAdapter

Dentro de los principales métodos de la clase MeasurementAdapter, se encuentra *addToBuffer*, *adjustBuffer* y *vaciarBuffer* los cuales se asocian con la incorporación de mediciones al buffer junto con su ajuste dinámico y vaciado. Los métodos *isGenerableCINCAMI_MIS* y *generateCINCAMI_MIS,* permiten verificar si se está en condiciones de utilizar el procesador de mediciones para generar el documento en base al esquema C-INCAMI/MIS y generarlo de ser posible en forma respectiva.

6.5 Gestión de Mediciones Mediante Buffer Multinivel

De momento, se ha analizado cómo se pueden recolectar las mediciones, almacenarlas localmente en un buffer para lograr una cantidad mínima que luego, serán mapeadas en un documento XML basado en el esquema C-INCAMI/MIS y transmitidas. Se ha mencionado en la sub-sección 3.1, que tales mediciones eran recibidas por la función de reunión y organizadas en base a los metadatos incorporados en el propio flujo.

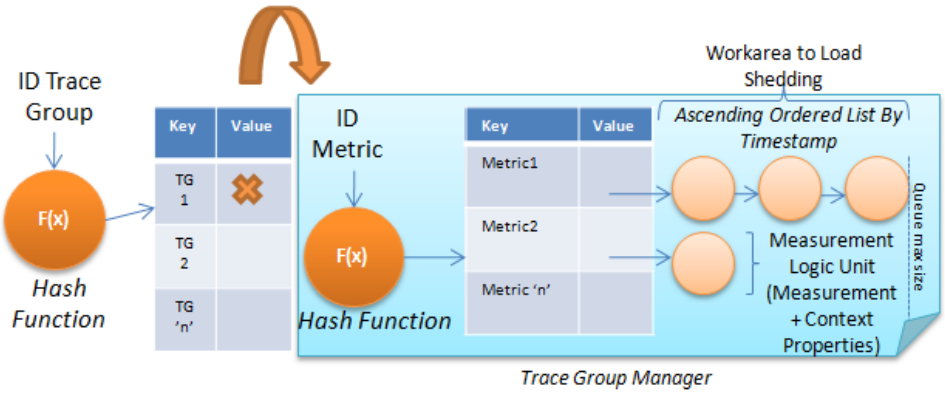


Figura 45. Conceptualización del Buffer Multinivel de la Función de Reunión

El buffer empleado en la función de reunión (ver Figura 45), posee tres niveles anidados mediante tablas de dispersión y es implementado mediante la clase *BufferTG* (ver Figura 46). El primer nivel, dispersa el ID de grupo de seguimiento para localizar un objeto del tipo *BufferMetric*. La clase *BufferMetric* administra las mediciones vinculadas a cada métrica para ese grupo de seguimiento (por ejemplo, aquí estarán todos los grupos de seguimientos administrados, como pueden ser el de "Juan Perez", "Luis García", entre otros). El segundo nivel, dispersa el identificador de la métrica para localizar la raíz de la lista que contiene las mediciones asociadas a ésta (por ejemplo, el idMetric 2 para la frecuencia cardíaca, según la Figura 37 dentro del grupo de seguimiento de "Juan Perez"). El tercer nivel, es el conjunto de mediciones específicas a la métrica y un grupo de seguimiento dado (por ejemplo, son todas las mediciones de las frecuencias cardíacas informadas para el grupo de seguimiento "Juan Perez"). Al igual que el buffer del MA (ver Figura 43), el buffer de la función de reunión posee parámetros que regulan su crecimiento, entre ellos *queue_max_limit* que trabaja en forma análoga a *buffer_queue_maxlimit* de MA.

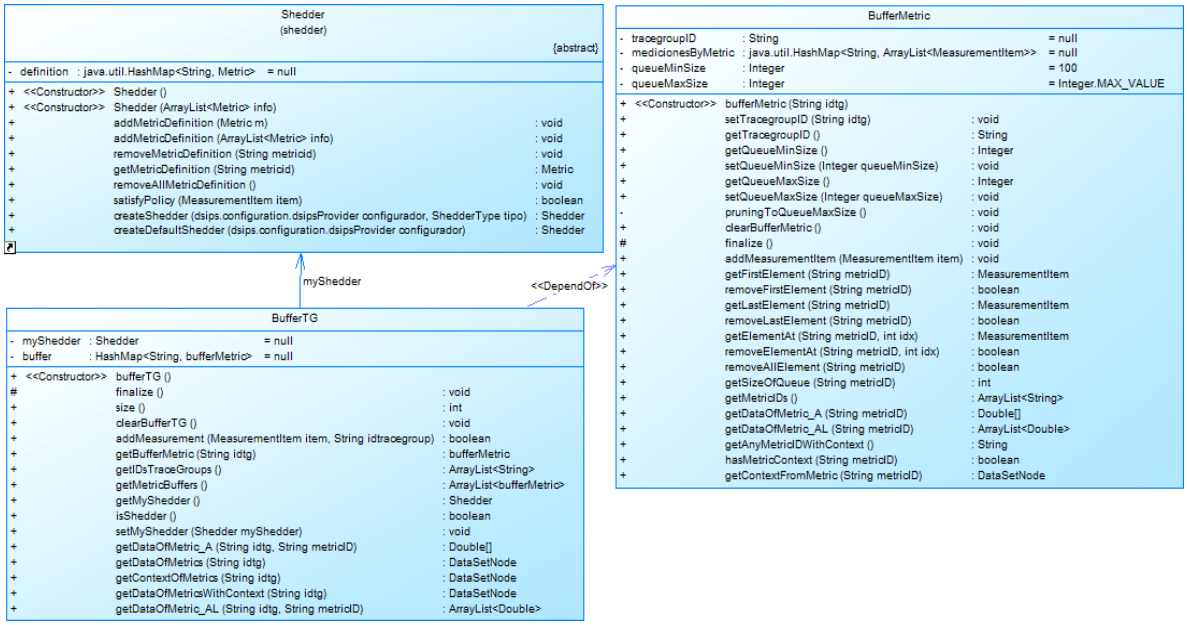


Figura 46. Implementación del Buffer Multinivel. Clases BufferTG, BufferMetric y Shedder.

La clase *BufferTG* será la responsable de ordenar las mediciones de acuerdo a los grupos de seguimiento en los que participe la fuente de datos que los originó. De este modo, si la fuente de datos participa en 'n' grupos de seguimiento (por ejemplo, el termómetro digital del caso de aplicación mostrado en la Figura 15, podría informar la humedad y temperatura ambiente para los grupos de pacientes "Juan Perez" y "Luis García" los cuales están situados en el mismo barrio), se dispersará tantas veces como grupos participe, para poder ubicar una copia de la medición en cada uno de ellos. Por otro lado, *BufferTG* contiene una colección de clases *BufferMetric* que administra las mediciones por grupo de seguimiento y las organiza por métricas. Así, cada grupo de seguimiento tendrá un único objeto del tipo *BufferMetric* que organizará la totalidad de las mediciones de acuerdo a cada una de las métricas involucradas en el mismo (por ejemplo, el grupo de seguimiento de "Juan Perez", según el caso de aplicación citado, tendrá las métricas asociadas con la temperatura axilar, frecuencia cardíaca y presión arterial, en forma conjunta con las propiedades de contexto temperatura ambiental, humedad ambiental, posicionamiento del paciente y presión ambiental).

La Tabla 5, analiza cada uno de los atributos de la clase *BufferMetric*, dado que en base a ellos, se gestionan las mediciones dentro de un grupo de seguimiento.

Tipo	Propiedad	Significado
String	tracegroupID	Identificador del grupo de seguimiento del cual administra las mediciones
Hashtable<String, ArrayList<MeasurementItem>>	medicionesByMetric	Tabla de dispersión con acceso mediante ID de métrica que contiene una lista con las mediciones organizadas por métrica.
static final Integer	queueMinSize	Longitud mínima reservada para las mediciones de una métrica dada en un grupo.
Integer	queueMaxSize	Longitud máxima permitida para las mediciones de una métrica dada en un grupo

Tabla 5. Propiedades de la Clase BufferMetric

La clase *BufferTG* presenta métodos asociados con la gestión del primer nivel del buffer (ver Figura 45), donde principalmente la operatoria radica en localizar el objeto *BufferMetric* que gestiona las mediciones por grupo de seguimiento, crearlo en caso de no estar presente o vaciar el buffer cuando corresponda. Adicionalmente, incorpora técnicas de descarte de carga de trabajo (load shedding) a través de la clase Shedder, la cual será discutida en la próxima sub-sección.

Gracias a la estructura de dispersión planteada en el primer nivel del buffer, si una fuente de datos integrase más de un grupo de seguimiento, sus mediciones serían replicadas con orden "K" (constante) en cada uno de ellos.

Por otro lado, la clase *BufferMetric* presenta métodos vinculados con la distribución de la medición en cada métrica correspondiente al grupo de seguimiento, junto con sus propiedades contextuales, administrando los niveles dos y tres del buffer global (ver Figura 45). La clase, gestiona directamente las mediciones por cada métrica dentro del grupo de seguimiento, contando con la posibilidad de ajustar dinámicamente el tamaño de la cola por métrica.

6.6 Técnicas de Descarte de Carga (Load Shedding)

Las técnicas de descarte de carga (Load Shedding), permiten abordar la problemática de cola de servicios ante eventuales desbordes. Ello podría producirse a partir de la volatilidad en las tasas de arribo de datos, desde los diferentes data streams, que nutren de mediciones al EIPFDcMM. Así, el objetivo del load shedding es actuar proactivamente sobre la cola de servicios, en situaciones donde la tasa de procesamiento de las mediciones es inferior a la tasa de arribo de datos, incorporando mecanismos de descarte selectivo con la finalidad de mantener el uso de recursos dentro de un marco limitado o finito, minimizando tanto como sea posible la pérdida de precisión (Chakravarthy & Jiang, 2009).

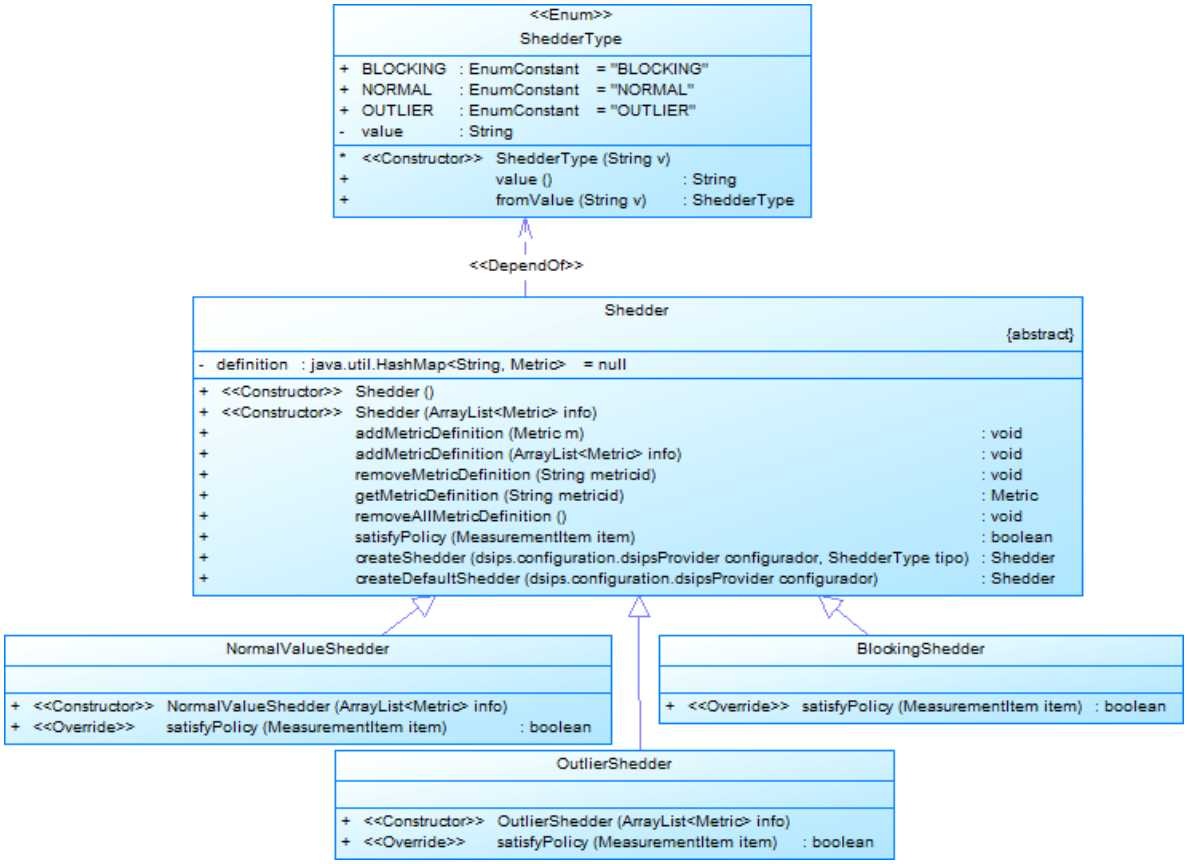


Figura 47. Principales Relaciones de la Clase Shedder

De este modo, uno de los principales temas a resolver dentro de los operadores de descarte es cómo eliminar el exceso de datos minimizando la pérdida de precisión, y dónde ubicar los mismos a los efectos de que la sobrecarga de procesamiento (overhead) que se incorpora sea mínimo en términos globales.

Dado que el buffer de gestión multinivel de EIPFDcMM, estableció que las mediciones en cada métrica dentro de un grupo de seguimiento (por ejemplo, la métrica *valor de la frecuencia* cardíaca del grupo de seguimiento “Juan Perez”), se comporten como una cola con tamaño finito, cualquier exceso repercutirá en el descarte de ‘n’ mediciones viejas para permitir la incorporación

de las ‘n’ mediciones nuevas. De este modo, el buffer mantiene siempre mediciones recientes y evita el desborde, pero ante una situación de arribo masivo y concurrente, se ha incorporado un operador de load shedding, como mecanismo adicional de control en el primer nivel del buffer.

El operador de load shedding, es modelado a través de la clase abstracta Shedder (ver Figura 47). Dicha clase, es aprovisionada con la especificación formal de la métrica dentro del proyecto de M&E durante la inicialización de la instancia, accediendo a la BD C-INCAMI (ver Figura 12) mediante una instancia de la clase *dsipsProvider.* La clase *dsipsProvider* implementa la funcionalidad asociada a la configuración dentro del EIPFDcMM y será discutida en la próxima sub-sección. El hecho de contar con la definición formal de las métricas sustentado en la base conceptual de C-INCAMI, permite interpretar las mediciones provenientes desde las diferentes fuentes de datos y determinar en principio, si los valores son consistentes a su definición (escala, tipo de escala, etc.), lo que permite detectar anomalías o ruido propio de la transmisión.

La clase Shedder plantea un método abstracto denominado *satisfyPolicy* el que recibe como parámetro una instancia de *MeasurementItem*. Este método, tiene por objetivo permitir que cualquiera de las clases derivadas, en base a los metadatos de la métrica obtenidos del proyecto de M&E y a la unidad lógica de medición recibida, determine cuál será la política de descarte que empleará.

La instancia de Shedder, es invocada desde el buffer de grupos de seguimiento (ver nivel 1 en Figura 45) en forma previa a la incorporación en los buffer a nivel de métrica. Ello permite, que el operador de descarte actúe en forma anterior al buffer, liberando de sobrecargas de eliminación e incorporación de mediciones si es que las mismas no se ajustan a la política vigente en el momento del arribo. La política de descarte, puede ser actualizada dinámicamente en tiempo de ejecución, informando a la instancia de *bufferTG* el objeto derivado de la clase Shedder con la nueva política mediante el método setMyShedder de *BufferTG*.

Actualmente el EIPFDcMM soporta tres políticas de descarte selectivo: BLOCKING, NORMAL y OUTLIER. La política BLOCKING, bloquea cualquier arribo de mediciones garantizando que el buffer permanecerá intacto hasta tanto se determine un cambio de política. La política NORMAL, deja pasar solo las mediciones que no son considerados como outliers, es decir aquellas que se rigen en forma similar al comportamiento general de la serie pre-existente, cuyo análisis es abordado en el siguiente capítulo. Por último, la política OUTLIER, permite solo el paso de mediciones cuyo comportamiento no se condicen con el comportamiento general de la serie.

Un aspecto fundamental a destacar en EIPFDcMM, es que la activación y desactivación del operador de descarte es dinámica, es decir que si la instancia de *BufferTG* tiene una referencia nula en su propiedad myShedder, las mediciones serán distribuidas dentro buffer multinivel con el control de cola limitado al valor predefinido en la misma, pero sin presencia alguna de política de descarte. Por otro lado, si la propiedad myShedder tiene referencia a alguna implementación derivada de la clase Shedder con su política asociada, ésta se aplica inmediatamente a la unidad lógica de medición que arribe al buffer de grupos de seguimiento, determinando si se descarta o continua hacia los buffer de mediciones asociados a las métricas.

Es importante destacar que independientemente del uso de la política de descarte, el control de colas de mediciones sobre las métricas en cada grupo de seguimiento siempre es efectuado, con el objeto de garantizar la longitud de mediciones pre-definidas.

6.7 Recepción de Mediciones

La función de reunión introducida en la sub-sección 3.2, tiene por responsabilidad principal, la recepción de las mediciones y su organización. La clase *DsipsProviderCommunication* implementa la interface *TransmitionServiceProvider* y *java.lang.Runnable,* de este modo, la clase se torna en la responsable primaria de brindar los servicios de transmisión, asumiendo bajo sus responsabilidades la recepción de mediciones junto con su organización. Así, la clase *DsipsProviderComunication* se constituye en la función de reunión (ver Figura 48).

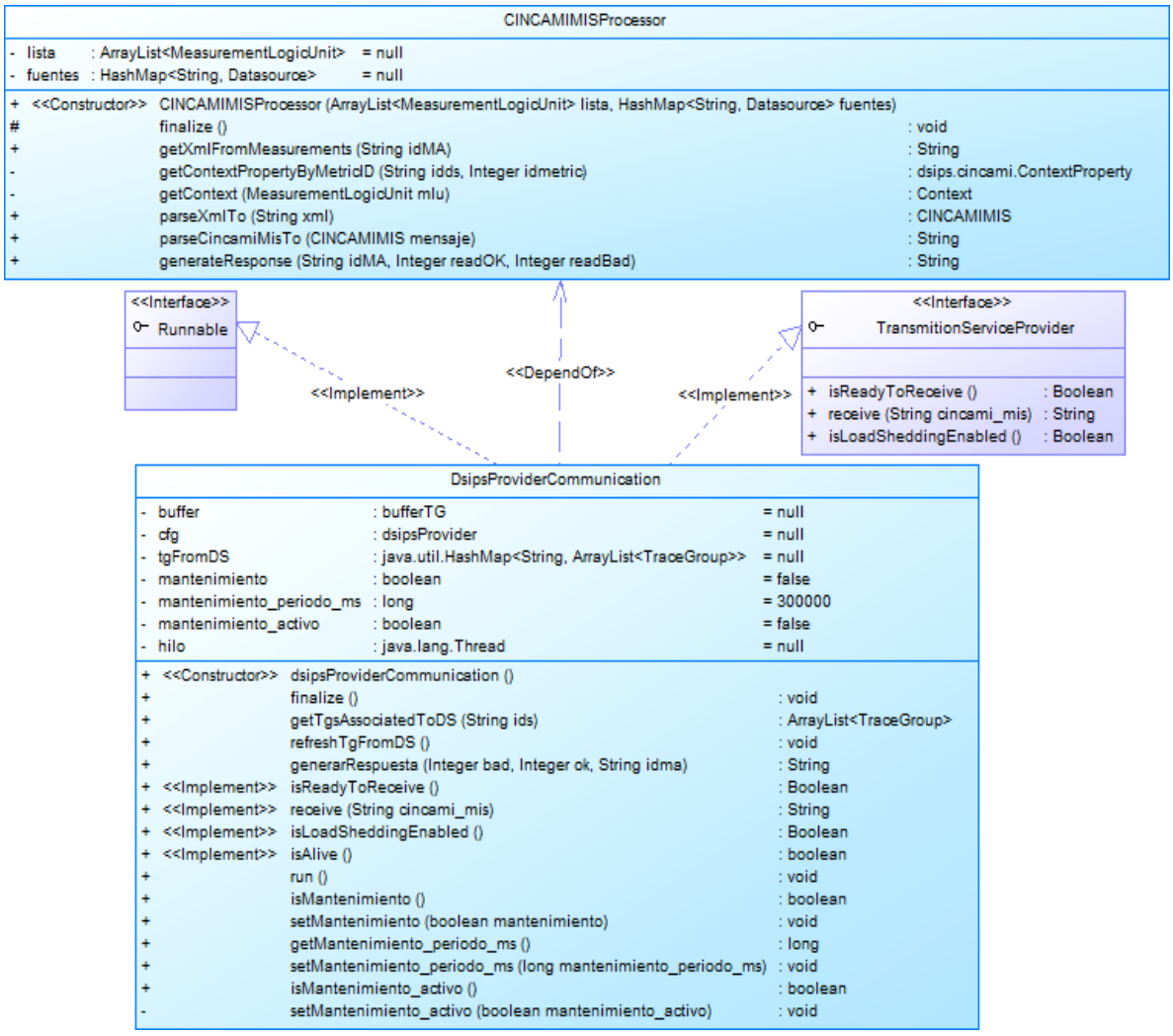


Figura 48. Dependencias e Interfaces Asociadas a la Clase *DsipsProviderCommunication*

La Tabla 5, analiza cada uno de los atributos de la clase *DsipsProviderCommunication*, dado que en base a ellos, se reciben y organizan las mediciones.

Tipo	Propiedad	Significado
BufferTG	buffer	Gestor del buffer de mediciones
dsipsProvider	Cfg	Gestor de configuración. Brinda acceso a la BD C-INCAMI para la recuperación de datos y metadatos.
Hashtable<String, ArrayList<Tracegroup>>	tgFromDS	Almacena en memoria los grupos de seguimiento asociados con cada fuente de datos
Boolean	mantenimiento	Bandera que indica si el proveedor de servicios de comunicación está en mantenimiento o no.
Integer	mantenimiento_periodo_ms	Ciclo en milisegundos cada cuanto efectuara mantenimiento.
Boolean	mantenimiento_activo	Bandera de control que indica si continua activo el Thread de mantenimiento o lo culmina.
java.lang.Thread	Hilo	Hilo al que se le pasará el actual objeto Runnable para que controle su ejecución mediante el método *run* y efectúe tareas de mantenimiento sobre el proveedor de servicios. El hilo es iniciado con el constructor y cada vez que se indica la finalización del mantenimiento.

Tabla 6. Propiedades de la Clase *DsipsProviderCommunication*

Los métodos de la clase *DsipsProviderCommunication* están orientados a recibir las mediciones, mediante la implementación de la interface *TransmitionServiceProvider*, como así también a organizar los mismos en base a sus metadatos mediante la clase *BufferTG*. Este proceso de recepción y organización se implementa en forma activa, a través de hilos de procesamiento paralelo de las mediciones, pudiendo encontrarse temporalmente en mantenimiento ante situaciones de actualización síncrona de los metadatos de los proyectos de M&E.

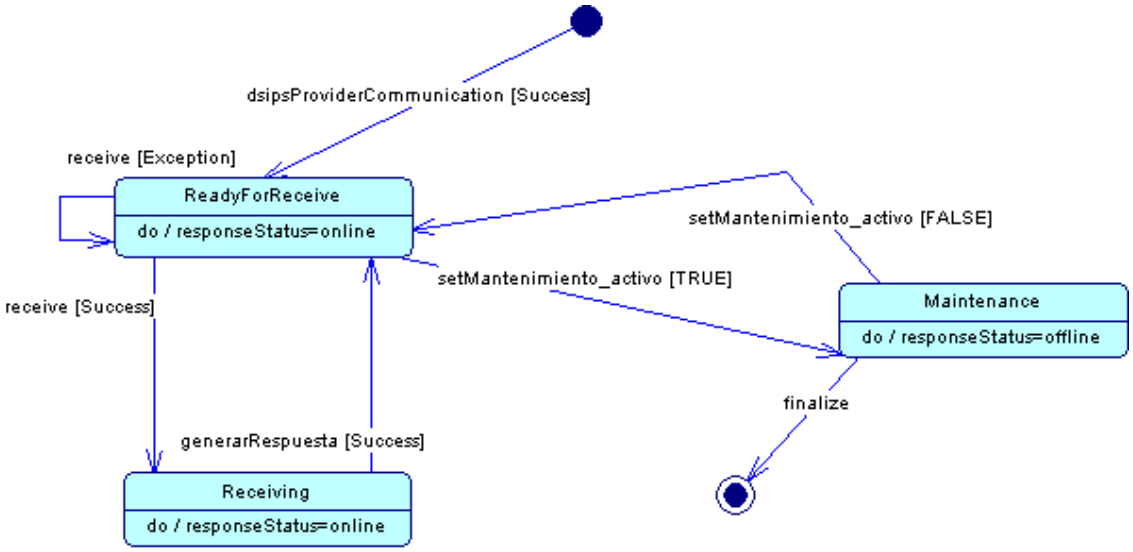


Figura 49. Diagrama de Estado de la Función de Reunión (Clase DsipsProviderCommunication)

La función de reunión puede estar básicamente en tres estados (ver Figura 49):

- ***ReadyForReceive***: La función de reunión se encuentra lista para recibir mediciones desde cualquier adaptador de mediciones. Desde aquí podrá recibir nuevas mediciones sin inconvenientes para lo que transita al estado '*Receiving*'. Sin embargo, si ocurriese algún inconveniente en el inicio de la recepción, se mantiene en el presente estado. Adicionalmente, puede iniciar tareas de mantenimiento *únicamente* desde este estado.

- ***Maintance***: La función de reunión, se encuentra actualmente en tareas de mantenimiento y por ende, imposibilitada de recibir mediciones. Al finalizar el mantenimiento, retornará al estado *ReadyForReceive* donde podrá retomar la recepción de mediciones. Únicamente desde el estado de mantenimiento el objeto puede ser destruido, ya que aquí se garantiza que no hay recepción de ninguna medición y, que por lo tanto no se interfiere con ningún flujo.

- ***Receiving***: La función de reunión se encuentra actualmente recibiendo mediciones, exista un error o se reciba correctamente, siempre generará una respuesta indicando el estado de la recepción.

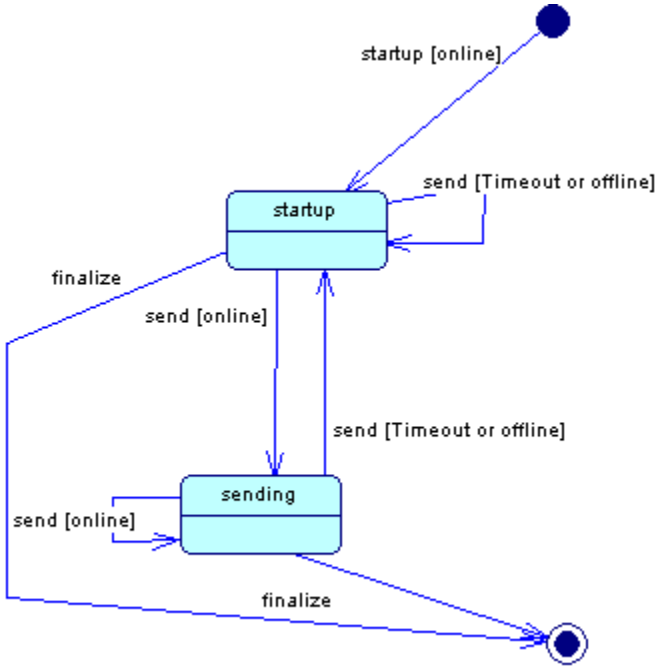


Figura 50. Diagrama de Estado para la Claser MeasurementAdapter

El adaptador de mediciones es el responsable de interactuar con la clase DsipsProviderCommunication (función de reunión), para transmitir las mediciones. Así, los estados en los que puede residir el adaptador de mediciones (ver Figura 50) son:

- ***Startup***: El MA se encuentra listo para transmitir, pero aún no ha comenzado ninguna transmisión. Si inicia una transmisión y falla, ya sea porque el proveedor de servicios se encuentra fuera de línea o porque efectúa un timeout, se mantiene en el estado.

- ***Sending***: El MA ha iniciado la transmisión del flujo sin inconvenientes y se mantendrá en el estado hasta culminar la misma. Puede suceder que al culminar un envío tenga cola de espera de mediciones por transmitir, con lo cual puede intentar desde el mismo estado

seguir transmitiendo. Si el inicio del nuevo envío procede correctamente se mantiene en el estado, caso contrario transita al estado *startup.*

En cualquiera de los dos estados puede recibir la señal de finalización del objeto. Esto es así, porque el MA se encuentra generalmente en un dispositivo móvil cercano a las fuentes de datos y podría ocurrir, entre otros tantos inconvenientes, que el dispositivo se quede sin energía, con lo cual *debe* garantizarse un estado de consistencia sea que se esté transmitiendo o no (por ejemplo, ver el escenario asociado al caso de aplicación, planteado en la Figura 15). En caso de estar transmitiendo y el MA cortar abruptamente el flujo, la función de reunión procesará solo hasta las mediciones recibidas correctamente y generará su respuesta con el resumen de la comunicación. Esto último, implica que la función de reunión garantiza la consistencia de los datos recibidos, descartando aquellos que son inconsistentes o no responden a la definición del esquema C-INCAMI/MIS.

7 Función de Suavización

En el capítulo 6 se ha analizado el esquema XML para el intercambio de mediciones sustentado en la base conceptual de C-INCAMI, denominado C-INCAMI/MIS. Luego, se ha discutido las interfaces de transmisión de mediciones y su relación con el procesador C-INCAMI/MIS, el cual permite la conversión entre el esquema C-INCAMI/MIS y los datos ralos desde las fuentes de datos. Se ha analizado el adaptador de mediciones y su rol ante la recolección y transmisión de mediciones. Posteriormente, se discutió la gestión de mediciones dentro de un buffer multinivel dentro de la función de reunión, a los efectos de la organización de las medidas para su posterior análisis y finalmente, se discutieron los mecanismos de load shedding y su incorporación en el buffer multinivel de mediciones, junto con el mecanismo de recepción de mediciones implementado desde la óptica de la función de reunión.

En este capítulo, se discute el motor de cálculo estadístico y su rol en la función de suavización. Por lo que, se analiza el concepto de operación estadística, como forma de abstraer el comportamiento homogéneo de las mismas, independientemente de cuál de ellas se realice. Y, se discute el análisis de correlación, el análisis de componentes principales, el análisis descriptivo y las pruebas paramétricas/no paramétricas sobre la distribución de probabilidad que pudiesen tener los flujos de datos.

A partir de esta base, se analiza el concepto de sinopsis, el cual permite responder a consultas sobre el comportamiento típico de una fuente de datos, incluso aunque la misma esté temporalmente fuera de servicio y/o presente inconvenientes de ausencia de datos. Luego, se discute la gestión de instantáneas, concepto similar a la idea de Data Warehousing (Dodge & Gorman, 2000), con respecto al hecho de tomar una fotografía de la situación de los datos en un instante de tiempo dado para su posterior análisis.

Luego, se discuten el rol de los suavizadores frente al flujo de datos y su relación con las instantáneas. Se analiza el rol del tomador de decisiones y los notificadores dentro del análisis estadístico.

Finalmente, se discute el analizador estadístico en forma conjunta con la parametría que rige su comportamiento, el cual

permite coordinar e implementar las responsabilidades propias de la función de suavización, presentada en la sub-sección 3.3.

7.1 Motor de Cálculos Estadísticos

Dentro del presente capítulo, se discutirán las diferentes clases que implementan las responsabilidades de la denominada función de suavización (sub-sección 3.3), como así también se analiza la clase responsable de la coordinación de las operaciones estadísticas. La clase responsable de la coordinación de las operaciones estadísticas, denominada *StatisticalAnalyzer*, será analizada en la sub-sección 7.12 y es quien implementa en definitiva, la función de suavización.

Las diferentes clases que integran la función de suavización, a discutir en el presente capítulo, utilizan el software R como motor de cálculos estadísticos. R es un proyecto GNU (Rosen, 2005; Hornik, 2011) de cómputo estadístico y gráfico, escrito inicialmente por Ihaka y Gentleman (Ihaka & Gentleman, 1996), del Departamento de Estadística de la Universidad de Auckland (Nueva Zelanda).

El software R actúa como servidor dentro de una arquitectura cliente/servidor. La interconexión de la función de suavización con el motor de cálculos estadísticos R, puede efectuarse embebiendo librerías escritas en C con el código de un programa Java, por ejemplo vía JNI de Java o bien, mediante TCP/IP a través del CRAN (Comprehensive R Archive Network) denominado Rserve (Urbanek, 2003). La ventaja que permite este último modo de interconexión a R sobre el primero, es que permite paralelizar la recepción de peticiones mediante su procesamiento a través de múltiples hilos. El paquete Rserve, permite a cualquier programa cliente, indistintamente de su lenguaje de desarrollo, conectarse a R mediante TCP/IP evitando tener que depender de otras librerías de R, o bien del enlace a ellas por medios estáticos o dinámicos. El módulo dentro del paquete Rserve actúa como servidor, permaneciendo a la escucha de solicitudes de los clientes. Recibida una solicitud, la envía al motor de cálculos estadísticos R para el cómputo respectivo, retornando al cliente el resultado. Rserve soporta conexiones remotas, autenticación y transferencia de archivos por lo que el lugar de residencia del mismo, al igual que R, podría ser otro nodo de procesamiento diferente de donde resida la función de suavización.

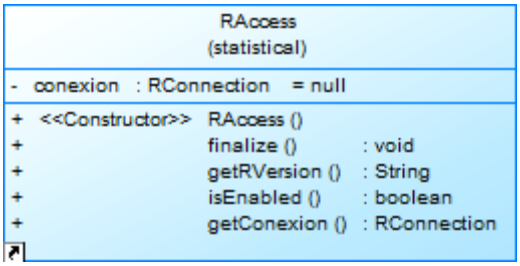


Figura 51. Clase RAccess

Para poder utilizar Rserve debe contarse con R y Rserve instalado en una computadora y el software cliente, debe tener referencia a los JARs de utilidades de Rserve (REngine.jar y

RserveEngine.jar), disponibles en la URL http://www.rforge.net/Rserve/files/. Estos JARs de Rserve, permiten abstraer a la aplicación cliente de la conexión con R, como de efectuar las conversiones de tipos de datos en forma automática, al igual que la autenticación y transferencia de archivos.

La clase RAccess (Ver Figura 51) es la responsable de gestionar la conexión al motor de cálculo estadístico R, a través del CRAN Rserve. De este modo, su principal responsabilidad consiste en la apertura, verificación y cierre de las conexiones con el software R mediante Rserve.

Al crearse una instancia de la clase RAccess, se intenta abrir una conexión al software R mediante R. De ser posible la apertura de la conexión, se almacena su referencia en la propiedad *conexion* de la clase RAccess. De este modo, el método *isEnabled* indicará TRUE si existe una conexión abierta y disponible a R, de lo contrario retornará FALSE.

Si alguna de las clases de la función de suavización requiriesen hacer uso del software R, deberán solicitar la conexión a través del método *getConexion* de la clase RAccess. Si existe una conexión abierta, se retorna un objeto RConnection, de lo contrario se retornará null. La clase *RConnection* es una abstracción que modela la conexión vía TCP/IP entre un aplicativo y el motor de cálculos R, y se encuentra situada en el paquete org.rosuda.REngine.Rserve. Una utilidad interesante que incorpora RAccess a través de los servicios brindados por Rserve, es la consulta sobre la versión de R que se está ejecutando, a los efectos de verificar la compatibilidad de los scripts o comandos que se intenten ejecutar en el mismo.

7.2 Operaciones Estadísticas

El conjunto de operaciones estadísticas que aplican cálculo multivariado sobre flujos de datos dentro de la función de suavización, presentan una serie de comportamientos similares a la hora de enviar los datos y requerir sus resultados. Así, se ha abstraído dicha similitud a través de la clase abstracta *StatisticalOperation* (ver Figura 52), de modo que cualquier operación estadística multivariada que se desee implementar, lo hará a partir de la especialización de esta clase.

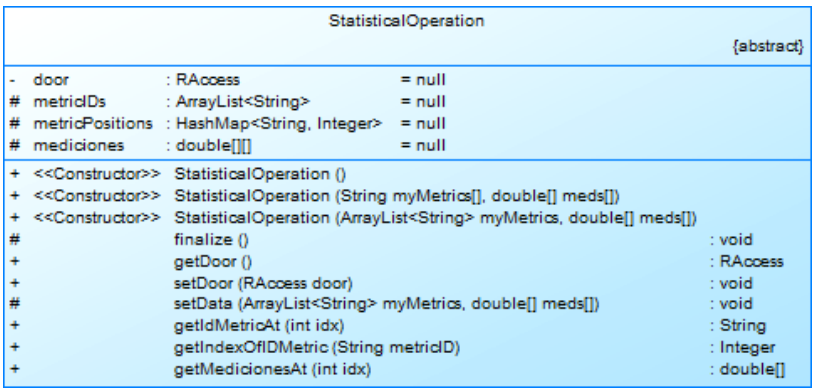


Figura 52. Clase StatisticalOperation

El análisis estadístico del paciente trasplantado ambulatorio, presentado en la sub-sección 3.5, es por naturaleza multivariado, por cuanto es necesario efectuar el análisis de la frecuencia

cardíaca, la temperatura axilar, la presión del paciente, la temperatura ambiental, la humedad ambiente, entre otros aspectos de monitoreo, en forma simultánea. Es decir, el análisis estadístico no sólo abordará una a una las variables, sino que también realiza el estudio de su interacción y evolución simultánea para determinar, entre otras cosas, si el comportamiento de una variable o métrica (sea de un atributo de entidad o bien, una propiedad de contexto) se encuentra arrastrando a otra. Por ejemplo, en este último sentido, si un paciente trasplantado está situado en una zona calurosa, es altamente probable que la métrica asociada a la temperatura ambiental arrastre a aquella asociada con la temperatura axilar

Los constructores de la clase *StatisticalOperation*, permiten inicializar la operación estadística sin datos o bien, indicando los datos mediante nombre de variable y conjunto de mediciones asociadas. En este sentido, notar que el concepto de variable queda definido como el identificador de la métrica, por lo que cada métrica, sea que este aplicada a un atributo de entidad o a una propiedad de contexto, se torna en variable bajo análisis a los efectos del análisis estadístico. La clase puede inicializar su propia conexión al software R mediante una instancia de la clase RAccess o bien, puede recibir o cambiar una conexión preexistente, mediante los métodos *setDoor* y *getDoor* respectivamente.

Dado que el cómputo se aplica sobre una matriz numérica densa, es posible obtener la posición de una determinada variable en la misma mediante el método *getIndexOfIDMetric*. En forma análoga, es posible obtener el nombre de la variable a partir de su posición mediante el método *getIdMetricAt*. Dada la posición de una variable en la matriz de datos, es posible obtener el subconjunto de mediciones propias mediante el método *getMedicionesAt*.

7.3 Análisis de Correlación

El Análisis de Correlación es una técnica que estudia la presencia de relaciones lineales entre un conjunto de variables, por ejemplo, si la temperatura axilar del paciente trasplantado está en función de la temperatura ambiental. Tal análisis puede efectuarse basándose en la matriz de covarianza o bien de la matriz de correlación. La matriz de covarianza es susceptible a los valores extremos, por lo que un outlier afectaría su valor, mientras que en la matriz de correlación, los valores son expresados mediante el coeficiente de correlación siendo más robustos frente a outliers. Por ejemplo, si el paciente trasplantado accidentalmente se situase cerca de un calefactor, los valores de su temperatura ambiental provenientes de su sensor, comenzarían a dispararse comportándose en forma diferente a su serie histórica, de este modo, si se utilizase la matriz de covarianza, la magnitud de tal diferencia podría afectar el análisis con respecto a su temperatura corporal, provocando que se disparen falsas alarmas. La matriz de correlación es una matriz cuadrada, en donde su diagonal principal es 1 y existe simetría en los coeficientes C_{ij} y C_{ji}, ya que representan la correlación entre la variable ‘i’ y ‘j’ o bien, entre ‘j’ e ‘i’ y las mismas son conmutativas. El coeficiente de correlación varía dentro del intervalo [-1,1] y los valores próximos a 1 (en valor absoluto), estarían indicando una fuerte correlación lineal entre las variables bajo análisis. Ahora bien, un valor cercano a cero no indica necesariamente ausencia de relación entre las variables, sino que a rigor de verdad estaría señalando que no existiría relación *lineal* entre las

mismas, de este modo podría existir otro tipo de relación no lineal y el coeficiente de correlación ser cero. Para mayores detalles sobre Análisis de Correlación consultar (Johnson, 1998).

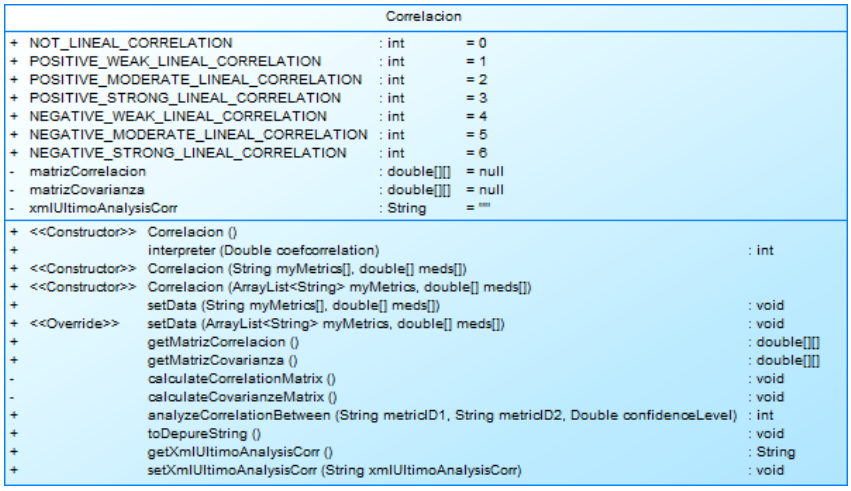


Figura 53. Clase Correlacion

La clase *Correlacion* (ver Figura 53) hereda la operatoria común al cálculo multivariado de *StatisticalOperation*. Además, implementa la funcionalidad para calcular la matriz de correlación e interpretar los coeficientes arrojados a partir del conjunto de datos indicados. Los datos y variables son indicados mediante métodos propios de *StatisticalOperation*, mientras que la clase especializa el método *setData* para el manejo de arreglos o bien listas dinámicas de datos, como así también el cómputo y evaluación de resultados propios del análisis.

La Tabla 7 expone el concepto asociado a cada una de las propiedades de la clase *Correlacion,* como así también su implicancia dentro del análisis de correlación.

Tipo	Propiedad	Significado
Integer	NOT_LINEAL_CORREATION	Constante que indica la no existencia de relación lineal entre variables
Integer	POSITIVE_WEAK_LINEAL_CORRELATION	Constante que indica la existencia de una débil relación lineal positiva entre variables
Integer	POSITIVE_MODERATE_LINEAL_CORRELATION	Constante que indica la existencia de una moderada relación lineal positiva entre variables
Integer	POSITIVE_STRONG_LINEAL_CORRELATION	Constante que indica la existencia de una fuerte relación lineal positiva entre variables
Integer	NEGATIVE_WEAK_LINEAL_CORRELATION	Constante que indica la existencia de una débil relación lineal negativa entre variables
Integer	NEGATIVE_MODERATE_LINEAL_CORRELATION	Constante que indica la existencia de una moderada relación lineal negativa entre variables
Integer	NEGATIVE_STRONG_LINEAL_CORRELATION	Constante que indica la existencia de una fuerte relación lineal negativa entre variables
Double[][]	matrizCorrelacion	Matriz de correlación densa calculada a partir de los datos informados
Double[][]	matrizCovarianza	Matriz de covarianza densa calculada a partir de los datos informados
String	xmlUltimoAnalysisCorr	Cadena XML que informa el resultado del último análisis de correlación. En él se informa variables involucradas, coeficiente de correlación, nivel de confianza y su intervalo de confianza asociado.

Tabla 7. Propiedades de la Clase *Correlacion*

A seguir se describen los principales métodos de la clase *Correlacion.* El constructor por defecto de la clase solo reserva espacio de memoria para la instancia, no abre conexión alguna al software R. De este modo, tanto los datos, como las variables y conexiones al motor de cálculo, deberán ser provistos mediante los métodos *setData* y *SetDoor* respectivamente. Sin embargo, las restantes alternativas para construir una instancia de *Correlacion*, intentan abrir su propia conexión al software R a la vez que reciben la información de las variables y sus valores asociados.

Los métodos *calculateCorrelationMatrix* y *calculateCovarianzeMatrix* permiten llevar adelante el cómputo de la matriz de correlación y de la matriz de covarianza respectivamente para la totalidad de las variables indicadas. Adicionalmente, el método *analyzeCorrelacionBetween* permite calcular el coeficiente de correlación entre dos variables y efectuar adicionalmente el cómputo del intervalo de confianza para el coeficiente obtenido. Dicho intervalo de confianza, puede ser obtenido mediante el método *getXmlUltimoAnalysisCorr,* luego de la ejecución de *analyzeCorrelacionBetween* y en base al nivel de confianza indicado en este último método. Estos métodos permiten por ejemplo, solicitar si para el paciente trasplantado “Juan Perez” (grupo de seguimiento) la temperatura axilar está en función de la temperatura ambiental. Obtenido el coeficiente de correlación para las métricas temperatura axilar y temperatura ambiental, es posible solicitar la realización de un intervalo de confianza sobre el mismo para determinar si éste se corresponde con un determinado nivel de significancia, con el objetivo de descartar situaciones de falsas correlaciones.

El método *interpreter* permite interpretar el coeficiente de correlación pasado como parámetro, respondiendo sobre el tipo de correlación al que correspondería el mismo. El tipo de correlación será uno de los indicados por las constantes públicas de la clase, por ejemplo: POSITIVE_WEAK_LINEAL_CORRELATION, etc.

El principal aporte de la clase *Correlacion* a la función de suavización, es que le permite efectuar sobre cada ventana de datos, un análisis sobre las potenciales relaciones lineales que podrían existir entre las diferentes métricas, indistintamente de cuáles sean estas e incluso, si las mismas estuvieran aplicadas a atributos de una entidad o bien, a propiedades del contexto. Por ejemplo, y tal como se mencionó anteriormente, es posible determinar si la temperatura axilar está en función de la temperatura ambiental para un paciente dado. De este modo, como se verá en la sub-sección 7.10, es posible disparar alarmas preventivas, por ejemplo ante situaciones de potencial arrastre de una variable sobre otra, para que el tomador de decisiones analice si corresponde o no darle curso a las mismas en base a la configuración del proyecto de M&E.

7.4 Análisis de Componentes Principales

“*El Análisis de Componentes Principales es un método estadístico multivariado de simplificación o reducción de la dimensión, que se aplica cuando se dispone de un volumen elevado de variables con datos cuantitativos correlacionados entre sí, persiguiendo obtener un menor número de variables, combinación lineal de las primitivas e incorrelacionadas, que se denominan componentes principales o factores (Principal Component - PC), que resuman lo mejor posible a las variables iniciales con la mínima pérdida de información y cuya posterior interpretación permitirá*

un análisis más simple del problema estudiado" (Pérez López, 2005). En otras palabras y yendo al caso de aplicación planteado en la sub-sección 3.5, dada la totalidad de las métricas que cuantifican distintos atributos de la entidad paciente trasplantado (valor de la temperatura axilar, valor de la presión, etc.), como así también aquellas que cuantifican sus propiedades de contexto (valor de la temperatura ambiental, valor de la humedad ambiente, etc.), deseamos saber cuál de todas ellas, contribuyen mayormente a la variabilidad del sistema. Su identificación es fundamental, por cuanto determinan los focos prioritarios, a partir de los cuales pueden originarse, entre otras cosas, las potenciales descompensaciones del paciente.

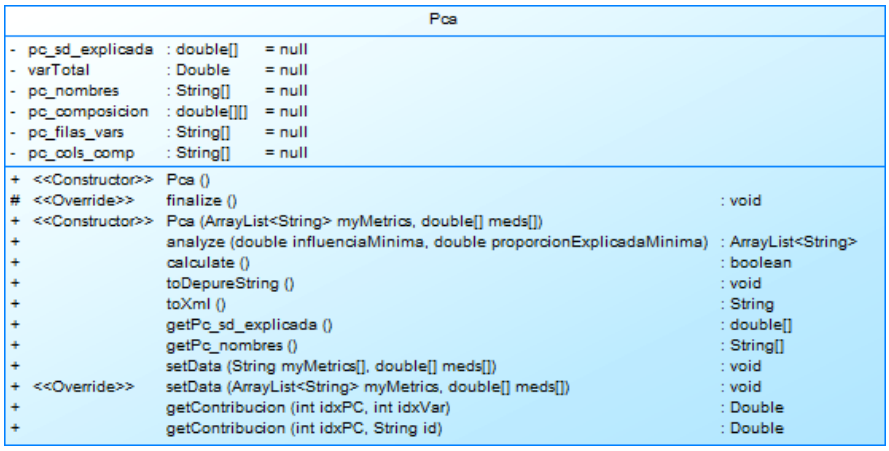


Figura 54. Clase Pca

Cada componente principal expresa una proporción de la variabilidad del sistema, por lo que si las variables primitivas están muy correlacionadas, puede detectarse qué conjunto de variables son aquellas que más inciden en la explicación de la varianza del sistema. De este modo el análisis de componentes principales permite complementar las alarmas que se emitan desde la función de suavización, por ejemplo al detectar correlaciones, dado permite detallar la estructura asociada a la variabilidad del sistema (en términos de identificadores de métricas), al momento de producirse la misma. Al informar en la estructura de variabilidad del sistema, los identificadores de las métricas, se conoce en forma indubitable para cada una de ellas los atributos de la entidad o propiedad de contexto asociada y de este modo, quienes afectan la estabilidad del sistema.

La clase *Pca* (ver Figura 54) hereda la operatoria común al cálculo multivariado de *StatisticalOperation*. Adicionalmente, implementa la funcionalidad para realizar el análisis de componentes principales e interpretar los resultados arrojados a partir del conjunto de datos indicados. Los datos y variables son indicados mediante métodos propios de *StatisticalOperation*, mientras que la clase especializa el método *setData* para el manejo de arreglos o bien listas dinámicas de datos, como así también el cómputo y evaluación de resultados propios del análisis.

La Tabla 8 expone el concepto asociado a cada una de las propiedades de la clase *Pca,* como así también su implicancia dentro del análisis de componentes principales.

Tipo	Propiedad	Significado
Double[]	pc_sd_explicada	Arreglo que contiene la desviación estándar explicada por cada componente principal
Double	varTotal	Varianza total del sistema
String[]	pc_nombres	Nombres de los componentes principales, Ejemplo: Comp.1. Este arreglo y *pc_sd_explicada* poseen la misma longitud y sus posiciones se corresponden, es decir que el componente principal denominado pc_nombres[0] explica una desviación de pc_sd_explicada[0] y así sucesivamente con los restantes componentes.
Double[][]	pc_composicion	Matriz de composición de los componentes principales. En las columnas reside cada uno de los componentes mientras que en las filas lo hacen las variables o métricas. De este modo la posición C_{ij} representa cuanto la variable o métrica 'i' aporta a la componente principal 'j'.
String[]	pc_filas_vars	Indica el ID de las variables o métricas. El orden del arreglo tiene relación directa con el orden de filas, por lo que la variable pc_filas_vars[0] está ubicada en pc_composicion[0] y el arreglo resultante es el aporte de dicha variable, a cada una de las componentes principales representadas como columnas.
String	pc_cols_comp	Indica los nombres de las componentes principales. El orden del arreglo tiene relación directa con el orden de columnas, por lo que la variable pc_cols_comp[0] está ubicada en pc_composicion[][0] y el arreglo resultante es el aporte que cada variable efectúa a esta componente principal en particular.

Tabla 8. Propiedades de la Clase Pca

A seguir se describen los principales métodos de la clase *Pca*. El constructor por defecto de la clase solo reserva espacio de memoria para la instancia, no abre conexión alguna al software R. De este modo, tanto los datos, como las variables y conexiones al motor de cálculo, deberán ser provistos mediante los métodos *setData* y *SetDoor* respectivamente. Sin embargo, las restantes alternativas para construir una instancia de *Pca*, intentan abrir su propia conexión al software R a la vez que reciben la información de las variables y sus valores asociados.

El método *calculate* lleva adelante el cómputo del análisis de componentes principales a partir del conjunto de datos informados. Los métodos *getPc_sd_explicada* y *getPc_nombres* permiten obtener la desviación estándar explicada por cada uno de los componentes principales, junto con el nombre de cada uno de los componentes respectivamente.

El método *getContribución* permite analizar cuánto aporta una variable o métrica (por ejemplo, valor de la temperatura axilar del paciente trasplantado) determinada a la variabilidad del sistema para un componente principal en particular. En forma más general, el método *analyze* retorna una lista con los identificadores de las variables, que contribuyen en una magnitud mayor o igual a la indicada como parámetro, y explican al menos una proporción mínima de la varianza la cual también se indica como parámetro. De esta manera por ejemplo, el análisis de componentes principales nos dirá cuál de las métricas del paciente (sean asociadas a un atributo del mismo o propiedad del contexto) aportan mayor variabilidad y por ende, representan los focos prioritarios de monitoreo por cuanto pueden ser causal de descompensación.

El principal aporte de la clase *Pca* a la función de suavización, es que le permite efectuar sobre cada ventana de datos, un análisis sobre la composición de la variabilidad global del sistema (por ejemplo, cada uno de los pacientes monitoreados) en términos de sus variables componentes (métricas asociadas a atributos de la entidad paciente o de sus propiedades contextuales), cuantificando las mismas y permitiendo establecer un orden de contribución de cada variable a dicha variabilidad global. De este modo, es posible informar la composición de la variabilidad del sistema al tomador de decisiones (capítulo 8), al momento de disparar las alarmas preventivas, con el objetivo de complementar el análisis sobre si corresponde o no darle curso a una alarma determinada, en base a la configuración del proyecto de M&E.

7.5 Análisis Descriptivo

El análisis descriptivo dentro del EIPFDcMM, consiste en calcular las medidas centrales, de tendencia y dispersión a partir de la serie de datos vinculada a cada variable (o métrica). Con dichas medidas, se puede caracterizar a una variable en un instante de tiempo dado e identificar incluso la presencia de outliers dentro de la serie. De este modo por ejemplo, si el paciente trasplantado se situase accidentalmente cerca de un calefactor, la magnitud de incremento de su temperatura ambiental sería detectada en principio, como un outlier de la métrica valor de la temperatura axilar.

Para caracterizar la serie de datos asociada a una variable, se toma la media, mediana, mínimo, máximo, cantidad de nulos, moda entera, primer cuartil, tercer cuartil, rango intercuartil, suma total de la serie, varianza, desviación estándar, coeficiente de variación, curtosis y asimetría.

Dado que las mediciones en las que se basa el EIPFDcMM son continuas, la moda entera se obtendrá a partir de la definición de un rango unitario y entero sobre el dominio de la variable, y del cálculo de la frecuencia de la parte entera de cada medición en cada uno de los rangos.

El Coeficiente de Variación es de particular importancia en el análisis descriptivo, por cuanto permite comparar la variabilidad de dos series de datos abstrayéndose de las unidades de medidas de cada una de ellas. Esto nos permite, entre otras cosas, determinar si la temperatura axilar del paciente presenta mayor variabilidad que la temperatura ambiental, como análisis complementario del de componentes principales.

Dichas medidas, constituyen las propiedades de la clase *DescriptiveResume* (ver Figura 55), la cual modela el concepto de instantánea de la serie de datos al momento de su arribo. El concepto de instantánea, tiene que ver con disponer de una descripción consistente, resumida y consolidada en un momento de tiempo dado para una determinada métrica. Ello permite, responder a consultas generales sobre cualquier variable (o métrica), ante la eventual ausencia de datos o interrupción del flujo de datos. Supongamos por ejemplo, que en un instante de tiempo el sensor de frecuencia cardíaca, el dispositivo GF-405GPS del caso de aplicación, sufre un desperfecto y deja de transmitir. El centro de salud si bien detecta la situación cuenta con la última información estadística con respecto a la frecuencia cardíaca del paciente, para poder determinar, entre otras cuestiones, si el paciente previo a que se produzca el desperfecto, se

encontraba cercano a una zona de riesgo. De este modo, la síntesis estadística, permite en cada paciente, brindar una respuesta aproximada con respecto al estado de la frecuencia cardíaca (u otra métrica) ante la ausencia de datos, salvando de este modo una situación de incertidumbre total con respecto a la misma.

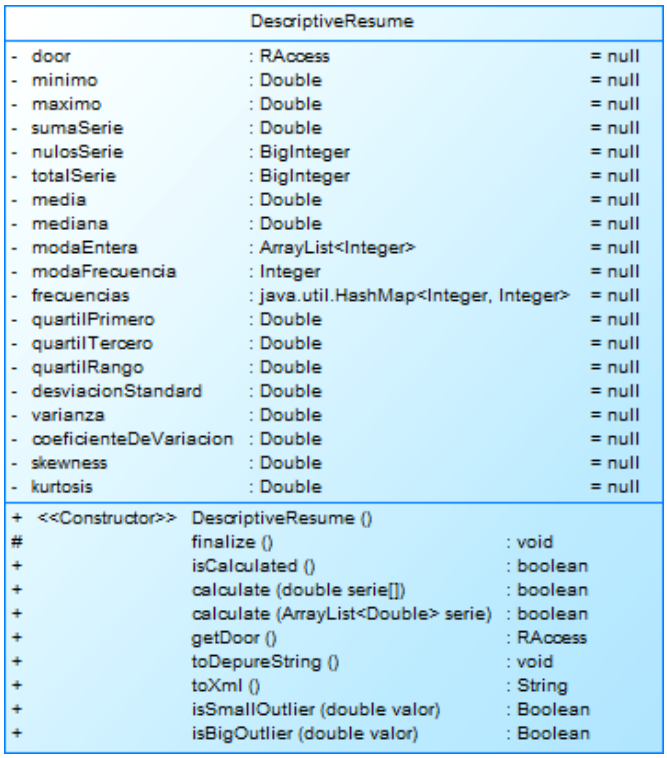


Figura 55. Clase DescriptiveResume

La clase *DescriptiveResume* (ver Figura 55) no hereda la operatoria común al cálculo multivariado de *StatisticalOperation,* dado que no corresponde a un cómputo multivariado. De este modo, los datos son indicados mediante el método *calculate* al momento de requerir que se genere las medidas resumen para una variable dentro de la ventana asociada a un flujo de datos.

Una vez que las medidas resumen han sido obtenidas para una variable (o métrica), la clase *DescriptiveResume* incorpora los métodos *isSmallOutlier* e *isBigOutlier* los cuales permiten, ante el arribo de un nuevo dato, determinar si el mismo se corresponde con un pequeño o gran outlier en forma instantánea. El outlier es un valor que presenta un comportamiento atípico al general de la serie, lo que no significa que sea inválido o incorrecto. A partir del rango intercuartil (RI), calculado como la diferencia en valor absoluto entre el primer y tercer cuartil, es posible definir intervalos, en función de ciertos límites (ver Tabla 9), para poder identificar outliers.

Límites Inferiores	Límites Superiores
L1= Primer Cuartil – 1.5*RI	U1= Tercer Cuartil + 1.5*RI
L2= Primer Cuartil – 3*RI	U2= Tercer Cuartil + 3*RI

Tabla 9. Cálculo de Puntos para la Determinación de Outliers

De este modo los valores que caen dentro del intervalo (L1,L2] se denominan pequeños outliers inferiores, mientras que aquellos valores inferiores a L2 se conocen como grandes outliers inferiores. En forma análoga, los valores que caen dentro del intervalo (U1,U2] se denominan pequeños outliers superiores, mientras que los valores superiores a U2 se conocen como grandes outliers superiores. Para mayor información sobre la identificación de outliers, remitirse a (Chambers, Cleveland, et al., 1983; Barnett & Lewis, 1994).

El método *isCalculated* indica si el cómputo de las medidas de resumen, a partir del conjunto de datos informado en el método *calculate*, ha concluido y se encuentra disponible para poder ser accedido.

El principal aporte de la clase *DescriptiveResume* a la función de suavización, es que le facilita la generación de instantáneas sobre cada métrica, permitiendo responder consultas generales sobre la misma (incluso ante ausencia de datos o interrupción del flujo de datos) e identificar en línea, outliers ante el arribo de nuevos datos.

7.6 Distribuciones de Probabilidad

Un aspecto fundamental de una serie de datos es poder conocer el modo en que se distribuye, o bien si la misma se corresponde con alguna distribución de probabilidad conocida. Para este aspecto en particular, se ha diseñado la clase *probDistribution,* cuya responsabilidad es determinar si dos series de datos corresponden a una misma distribución (indistintamente cual fuese ella) o bien, si se distribuyen en forma normal (caso particular del test anterior). Esto es útil por ejemplo, para determinar si la temperatura axilar de un paciente corresponde a la misma distribución empírica que la temperatura ambiental, a los efectos de complementar la determinación de arrastres de una métrica sobre otra.

Para verificar si dos series de datos corresponden a una misma distribución empírica, indistintamente de cual fuese ella, se emplea el test de Kolmogorov-Smirnov (Chakravarti, Laha, et al. , 1967). Ahora bien, existen casos en los que se desea verificar si la serie se distribuye en forma normal, dado que suele ser condición necesaria de muchos test estadísticos, para ello se emplea una modificación del test de Kolmogorov-Smirnov denominado test de Anderson-Darling (Stephens, 1974).

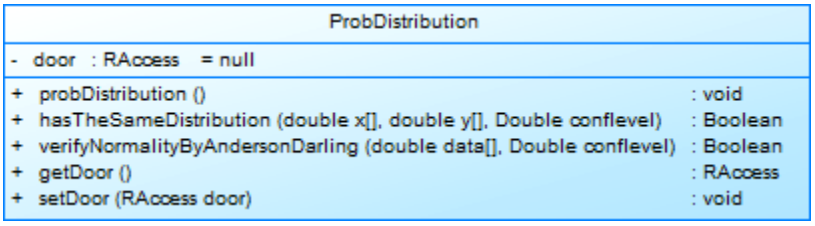


Figura 56. Clase ProbDistribution

La clase *ProbDistribution* (ver Figura 56), es empleada en forma interna y de modo indirecto por el analizador estadístico, a profundizar en la sub-sección 7.12, previo decidir si disparará o no una alarma determinada.

Al igual que las clases *Pca*, *Correlacion* y *DescriptiveResume*, la clase *ProbDistribution* obtiene acceso al motor de cálculo estadístico a través de RAccess, mediante los métodos setDoor y getDoor. Esto no implica que herede de StatisticalOperation, dado que no se trata de un cálculo multivariado sino de un test sobre la distribución de los datos.

El método *hasTheSameDistribution* recibe como parámetro dos series de datos y un nivel de confianza determinado, para determinar si las mismas corresponden o no a una misma distribución empírica. En forma análoga, pero acotado a la distribución normal, el método *verifyNormalityByAndersonDarling*, determina si la serie de datos pasada como parámetro, en función del nivel de confianza definido, corresponde o no a una distribución normal según la prueba de Anderson-Darling.

El principal aporte de la clase *ProbDistribution* a la función de suavización, es que incorpora las pruebas de normalidad y de correspondencia de distribución empírica entre dos series de datos, permitiendo complementar al analizador estadísticos al momento de determinar si da curso o no a una determinada alarma.

7.7 Sinopsis

Se entiende al concepto de sinopsis, dentro del EIPFDcMM, como un conjunto de datos sintetizados a partir de un flujo de datos, a través de los cuales se puede brindar una respuesta aproximada. Este conjunto de datos, ocupa una porción de memoria significativamente inferior a los datos originales y permiten responder consultas ante situaciones de interrupción o ausencia del flujo de datos. En base a esto, es fundamental la participación de las instantáneas asociadas a las medidas descriptivas (instancia de la clase *DescriptiveResume*) en el concepto de sinopsis. Para mayores detalles conceptuales sobre sinopsis se puede consultar (Chaudhry, Shaw, et al., 2005) o bien, desde el punto de vista de su implementación, se puede consultar la arquitectura de MavStream en (Chakravarthy & Jiang, 2009).

La clase S*inopsis*, organiza una síntesis descriptiva de la serie de datos original, agrupada por grupo de seguimiento. Cada agrupación, se organiza internamente por el identificador de métrica y tiene asociado a ella el resumen descriptivo de la serie de datos para dicha métrica en el grupo de seguimiento definido. La organización de la serie de datos por grupo de seguimiento es responsabilidad de la clase *Sinopsis*, mientras que la organización del resumen descriptivo para cada métrica en un grupo de seguimiento, es responsabilidad de la clase *MetricSynopsis* (ver Figura 57). Este aspecto es de suma utilidad en el monitoreo de pacientes trasplantados, por cuanto cada grupo de seguimiento, tiene a priori y en función de las mediciones provenientes de las fuentes de datos asociada al mismo, un resumen estadístico de cada métrica (sea que cuantifique un atributo de entidad o propiedad de contexto, por ejemplo: valor de la temperatura ambiental, valor de la humedad ambiente, valor de la temperatura axilar, etc.), que le permite responder en forma aproximada sobre la situación del paciente, ante la ausencia de datos y considerando la totalidad de las métricas vinculadas al grupo. De este modo, se evita una situación de incertidumbre total, posiblemente surgida de un desperfecto generalizado de los sensores del paciente.

Este modo de organizar los resúmenes estadísticos, permite acceder a cada uno de ellos en un tiempo constante, facilitando la comparación de las métricas entre diferentes grupos de seguimientos (por ejemplo, entre el paciente "Juan Perez" y "Luis García"), ya que los identificadores de la misma son homogéneos a lo largo de ellos. Esto implica, que un identificador de métrica no se ve alterado por participar en más de un grupo y ello permite, localizar a cada métrica en forma indubitable dentro de los grupos de seguimiento a los efectos de su análisis y/o comparación. Por ejemplo, tal como se presentó en la sub-sección 6.1 con el flujo comentado de C-INCAMI/MIS, la métrica *valor de la frecuencia cardíaca* cuyo identificador era 2 -*idMetric*- mantendrá dicho valor sea que transmita mediciones de "Juan Perez" o "Luis García". En igual sentido, el hecho de organizar el conjunto de resúmenes estadísticos bajo la clase MetricSynopsis, permite efectuar comparaciones de similitud entre objetos de dicho tipo en base a las métricas intervinientes y sus resúmenes estadísticos.

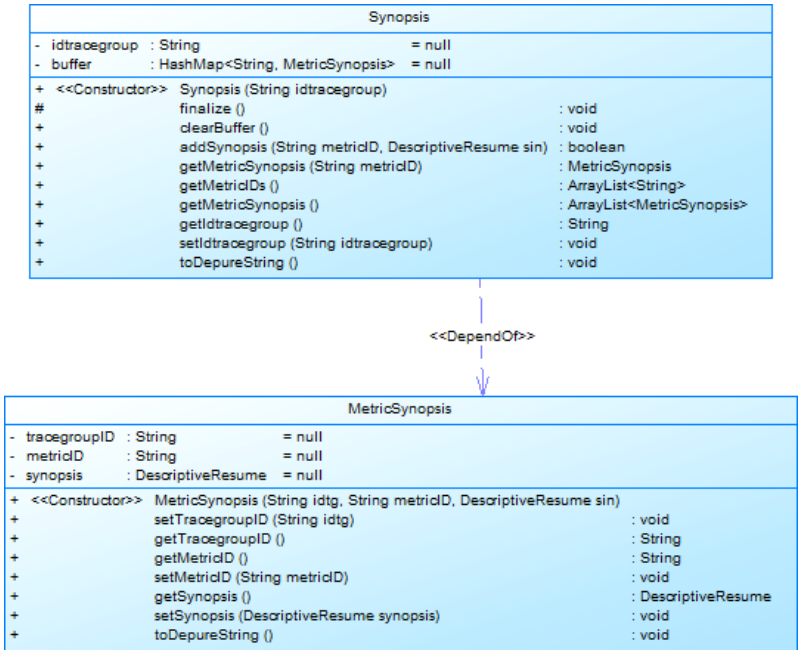


Figura 57. Clase Synopsis y MetricSynopsis

Los métodos de la clase *Synopsis* y *MetricSynopsis*, básicamente implementan las capacidades de dispersión para agregar y/o sustituir resúmenes estadísticos, como así también su esquema de agrupación por métrica y/o por grupo de seguimiento.

7.8 Gestión de Instantáneas de Datos

Las clases *DataSet* y *DataSetNode* (ver Figura 58), permiten abstraer al analizador estadístico del manejo de datos desde el buffer según se esquematizó en la Figura 45. El Analizador Estadístico, a profundizar en la sub-sección 7.12, obtiene una referencia al buffer (Clase *BufferTG*) de la función de reunión (Clase *DsipsProviderCommunication*, ver sub-sección 6.7) y a través de ella, extrae una instantánea de los datos más recientes. Dicha instantánea, es organizada mediante las clases *DataSet* y *DataSetNode* a los efectos de independizar el procesamiento

estadístico de la funcionalidad de la función de reunión, evitando interferencias y bloqueos ante accesos concurrentes y paralelos.

La clase *DataSet*, a través del identificador del grupo de seguimiento, permite organizar el conjunto de datos por grupo de seguimiento y establece una correspondencia entre cada identificador de grupo con una instancia de *DataSetNode* (ver Figura 58). Cada instancia *DataSetNode*, internamente almacenará las variables y sus valores asociados, dando como resultado una matriz *nxm*, donde '*n*' representa la cantidad de variables o métricas y '*m*' la cantidad de mediciones asociada a cada una de ellas.

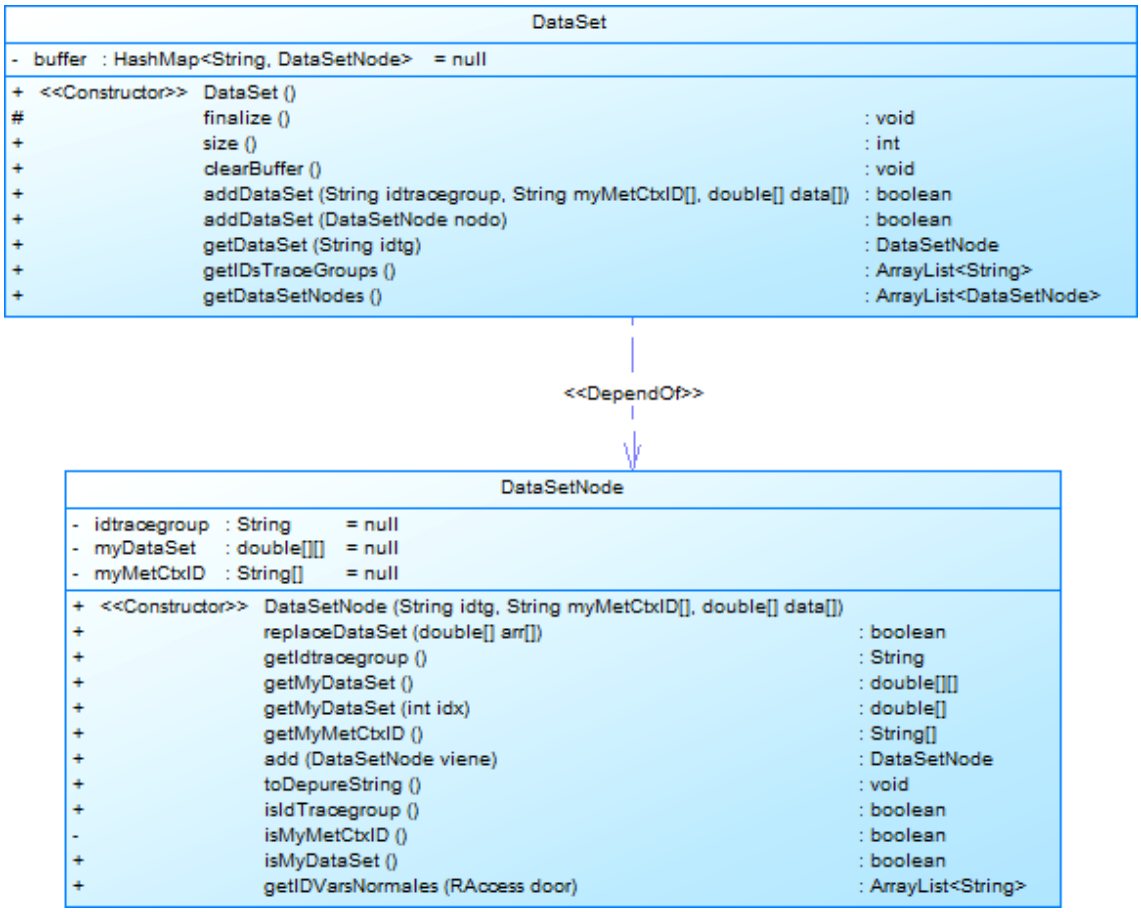


Figura 58. Clases DataSet y DataSetNode

Cada instancia de *DataSetNode* representa la instantánea de las mediciones para un grupo de seguimiento dado, de modo que al momento de efectuar los cálculos descriptivos, se tomarán los datos a partir de dicha instancia evitando incursionar en bloqueos innecesarios sobre el buffer de la función de reunión (Clase *BufferTG*). Por ejemplo, cada instancia de *DataSetNode* sería como una hoja de cálculo donde en cada columna dispondría de las métricas asociadas a un grupo de seguimiento (por ej.: valor de la temperatura axilar, valor de la temperatura ambiental, valor de la humedad ambiente, valor de la frecuencia cardíaca, entre otras, asociadas a "Juan Perez") y en cada fila, tendría un instante de tiempo determinado (el momento en que se toma cada

medición). La intersección entre cada fila y columna, representaría la medición de dicha métrica (por ej.: la frecuencia cardíaca) en ese instante de tiempo, para un grupo de seguimiento en particular (por ej.: "Juan Perez"). Debe destacarse, que es posible que una métrica para un instante de tiempo dado no posea valor, lo cual puede haberse originado por un desperfecto en el sensor.

No debe confundirse la instantánea asociada a las medidas estadísticas de resumen para una métrica (o variable) dada, con respecto a la instantánea de datos del buffer. La primera, contiene las medidas estadísticas centrales y de tendencia para una métrica particular; mientras que la segunda, representa una copia del buffer efectuada en un momento de tiempo dado, con respecto a la totalidad de los grupos de seguimiento, sus métricas y mediciones asociadas.

Cada instancia *DataSetNode* controlará la correspondencia entre las variables informadas y su conjunto de mediciones, es decir, que si existen 10 variables (o métricas) para un grupo de seguimiento, el arreglo de mediciones *deberá* poseer 10 filas con las mediciones vinculadas. De no respetarse esto último, se dispara una excepción y los datos no son inicializados en el nodo. Esto permite, tener la certeza que todo cálculo que se efectúe a partir de un conjunto de datos en el nodo, tiene correspondencia con una variable (o métrica) particular dentro el mismo.

De este modo, el único que posee acceso al buffer de la función de reunión es el analizador estadístico (ver sub-sección 7.12), el cual se responsabilizará de aprovisionar los mismos a cada una de las clases que lo requieran a los efectos de realizar los cálculos estadísticos. Dentro de tales clases, se encuentran *Correlacion*, *Pca* y *DescriptiveResume* analizados previamente en las sub-secciones 7.3, 7.4 y 7.5 respectivamente.

7.9 Suavizadores

El concepto de suavizado de una serie de datos tiene que ver, con realizar en forma previa a la toma de decisiones, la aplicación de un conjunto de políticas de ajuste de datos que satisfaga un objetivo específico. De este modo, la política de suavización podrá ser diferente de acuerdo al objetivo que se persiga.

El Analizador Estadístico, el cual se abordará en la sub-sección 7.12, puede utilizar suavizadores sobre la serie de mediciones informadas para una variable (o métrica). Tales suavizadores son de aplicación opcional, con lo que su empleo estará sujeto a la configuración del proyecto de M&E (dentro del marco del análisis estadístico), el cual es abordado en la siguiente sub-sección.

El EIPFDcMM propone el modelado de suavizadores a partir de una clase base abstracta, denominada *Smoother* (ver Figura 59), de la cual se derivarán los suavizadores específicos e implementarán la política de suavización que corresponda a los efectos de garantizar la extensibilidad.

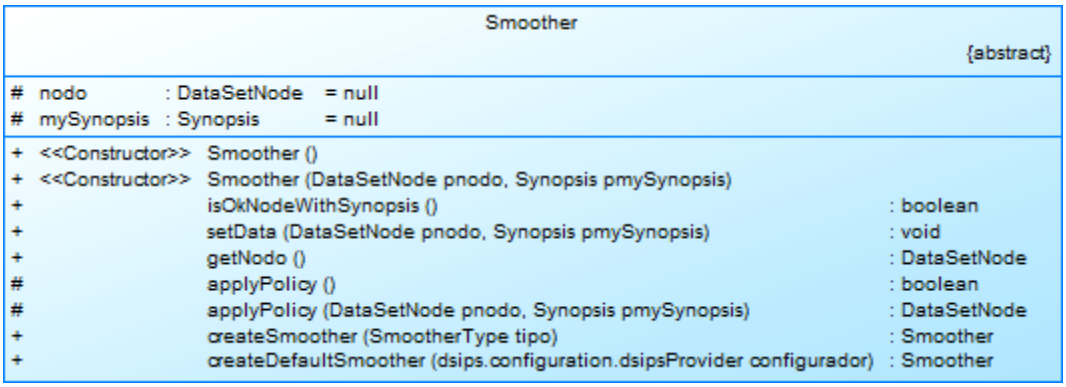


Figura 59. Clase Smoother

El método *isOKNodeWithSynopsis* verifica que la síntesis preexistente de los datos, sea consistente en término de cantidad de variables y volumen de mediciones asociadas para la aplicación de la política de suavizado.

La inicialización de los datos de referencia (sobre los que se aplicará la suavización) y el patrón de comparación (a ser utilizado para el ajuste de la serie de datos de acuerdo al objetivo que se defina), se efectúa mediante el método *setData*. Los métodos *applyPolicy* deberán ser implementados por las clases derivadas, indicando el tipo de ajuste a aplicar en su interior y retornando un *nuevo* conjunto de datos que satisface la política implementada. En otras palabras, el objetivo al cual debe satisfacer una política de suavizado dada, se implementa mediante el método *applyPolicy*, de allí que el mismo deba ser implementado por las clases derivadas de la clase *Smoother*.

Adicionalmente, la clase Smoother (ver Figura 59) posee dos métodos estáticos. El primero de ellos es *createSmoother(SmootherType tipo),* el cual recibe el tipo de suavizador a crear y retorna una instancia derivada de Smoother que implementa dicho suavizador. Los tipos de suavizadores disponibles, son indicados a través de la enumeración *SmootherType*. En EIPFDcMM, se proponen inicialmente los siguientes tipos de suavizadores: NONE, NULL_BY_MEDIA, NULL_BY_MEDIANA, NULL_BY_Q1, NULL_BY_Q3, NULL_BY_MIN y NULL_BY_MAX (ver Tabla 10). Estos tipos de suavizadores son los soportados actualmente, pudiéndose extender éstos o agregar otros, a partir de la modificación de *SmootherType* y de la clase *Smoother*.

Tipo	*Concepto*
NONE	*Es una implementación de un tipo de suavizador que mantiene la serie de datos original sin efectuar cambio alguno. Es un modo alternativo de indicarle al analizador estadístico que no se efectuarán modificaciones a la serie dato original y que la serie suavizada es la misma serie original.*
NULL_BY_MEDIA	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por la media aritmética. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún modo.*
NULL_BY_MEDIANA	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por la mediana. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún modo.*
NULL_BY_Q1	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por el valor asociado al primer cuartil. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún*

	modo.
NULL_BY_Q3	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por el valor asociado al tercer cuartil. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún modo.*
NULL_BY_MIN	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por el valor asociado al mínimo. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún modo.*
NULL_BY_MAX	*Esta implementación analiza la presencia de nulos en la serie y como política implementada, reemplaza cada nulo por el valor asociado al máximo. La serie de datos modificada es paralela a la original, es decir que ésta última no se altera de ningún modo.*

Tabla 10. Políticas de Suavizado Disponibles en SmootherType

Como se ha mencionado anteriormente y en forma adicional a las políticas de suavizado disponibles, podría plantearse un subconjunto de suavizadores derivados de la clase *Smoother,* que implementen distintas funcionalidades para el abordaje de outliers y su consiguiente reemplazo. En nuestra estrategia de procesamiento en particular, el análisis de outliers es de singular importancia por cuanto nos permite disparar alarmas preventivas y representan un aspecto importante de análisis sobre las mediciones, motivo por el cual se han mantenido. Para hacer más gráfica esta última decisión y, utilizando el caso de estudio presentado en la sub-sección 3.5, suponga que para un paciente dado, se recibe una temperatura corporal de 40 grados centígrados. Esto último, posiblemente sea un outlier pero seguramente me interesará disparar una alarma asociada al grupo de seguimiento del paciente y no efectuar su descarte.

El EIPFDcMM rige el comportamiento de su análisis estadístico, a través de la configuración de un subconjunto de parámetros dentro de la base de datos C-INCAMI (ver sub-sección 3.3), los cuales serán abordados en la siguiente sub-sección. En este sentido, cabe mencionar que es posible indicar al EIPFDcMM, cuál será el tipo de suavizador que deberá utilizarse mediante parametría y éste, al momento de su inicialización, será automáticamente creado y asociado al analizador estadístico (ver sub-sección 7.12).

7.10 Tomador de Decisiones y Notificadores Relativos al Análisis Estadístico

El EIPFDcMM, supone que ante la detección de determinado evento, se deberá informar del mismo al tomador de decisiones, y éste resolverá si procede o no a disparar una o más alarmas. De este modo, se discute en esta sub-sección, el rol del tomador de decisiones (representado por la clase *DecisionMaker*) circunscripto al análisis estadístico, junto con las formas de notificación disponibles con la que cuenta.

Este aspecto constituye la fase detectiva del modelo, el cual sustenta su comportamiento en la definición del proyecto de M&E, los análisis estadísticos y la base conceptual ontológica de C-INCAMI que permite en forma robusta y consistente, definir cada uno de los conceptos intervinientes dentro del proceso de medición. De este modo, si para el paciente trasplantado ambulatorio se recibe una temperatura axilar superior a los 38 grados centígrados, a través de los criterios de decisión del indicador elemental *Nivel de la Temperatura corporal* (ver Tabla 1), se interpreta una situación de “Fiebre Alta”, por lo que se informa al tomador de decisiones, quien en

base al criterio de decisión definido para el mencionado indicador, dará curso a la alarma al centro de salud.

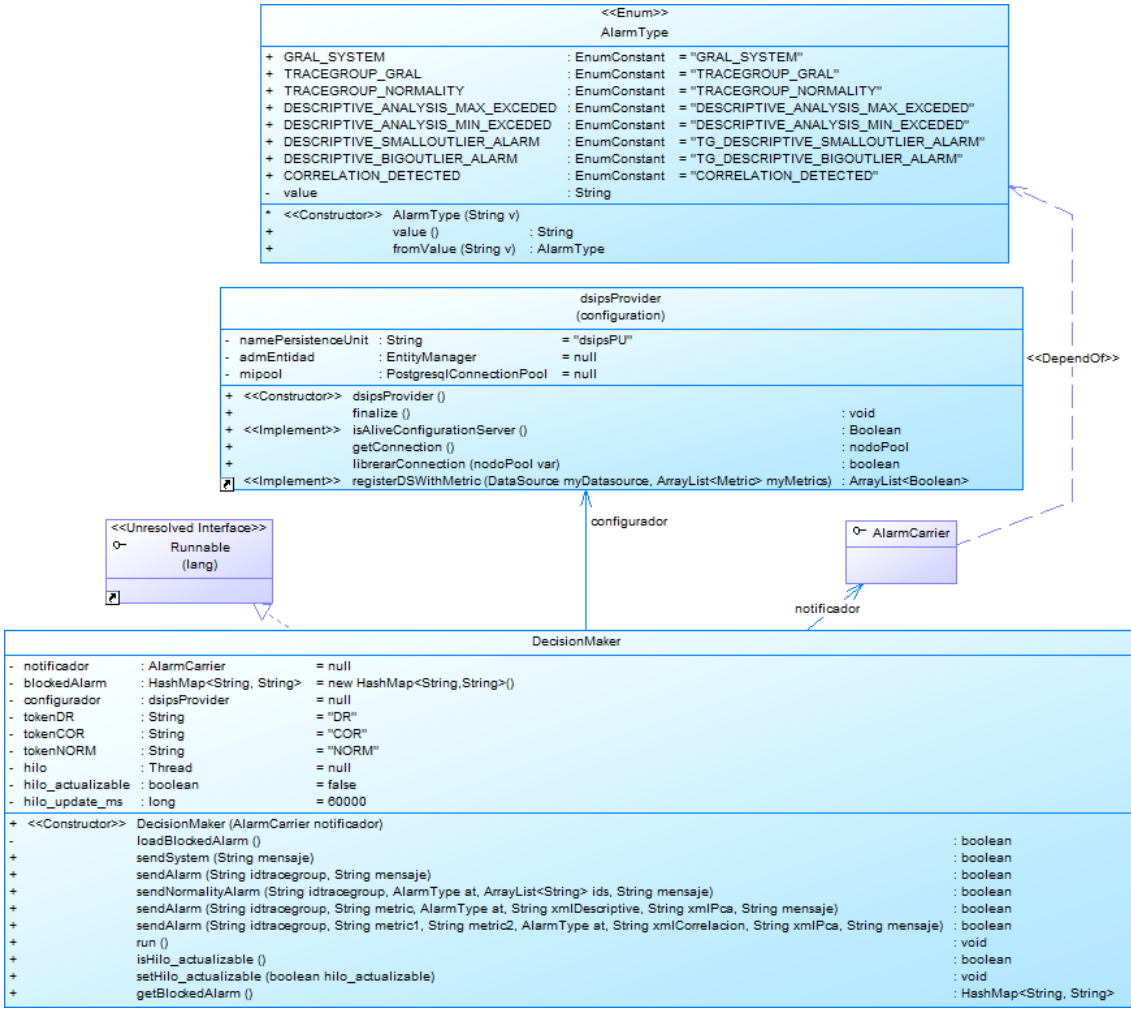


Figura 60. Clase DecisionMaker y sus Principales Relaciones

La clase *DecisionMaker* (ver Figura 60) tiene acceso a la información de configuración de los diferentes aspectos del proyecto de M&E, a través de la clase *dsipsProvider* (ver sub-secciones 5.5 y 6.6). Además, se asocia con un notificador de alarmas (o eventos), permite el bloqueo de determinados tipos de alarmas, y dispone de una actualización en segundo plano, de los parámetros de configuración de eventos desde la base de datos C-INCAMI.

Un aspecto a destacar, es que una instancia de *DecisionMaker* sólo puede ser inicializada en la medida que se disponga de una instancia que implemente la interface *AlarmCarrier* (ver Figura 61). Esto último es clave, dado que si el tomador de decisiones recibe un determinado tipo de alarma (o evento) y corresponde informarlo, ***debe*** disponer de un medio a través del cual canalizar el mismo. El medio por el cual canaliza las alarmas, está dado por la clase que implementa la interface *AlarmCarrier*. Este aspecto es esencial, por cuanto supongamos que recibe desde el paciente trasplantado una temperatura axilar superior a los 38 grados centígrados; según la

interpretación del indicador elemental *Nivel de la Temperatura corporal*, se debiera notificar al centro de salud en forma inmediata por cuanto constituye, según los expertos, un factor de riesgo para la salud del paciente. La no notificación o la imposibilidad de hacerlo, daría por tierra todo intento de análisis estadístico, definición de proyectos de M&E, ya que sería como querer gritar y no tener voz para hacerlo, se estaría impedido fácticamente de conseguir el objetivo "Prevenir o, en el peor de los casos, detectar".

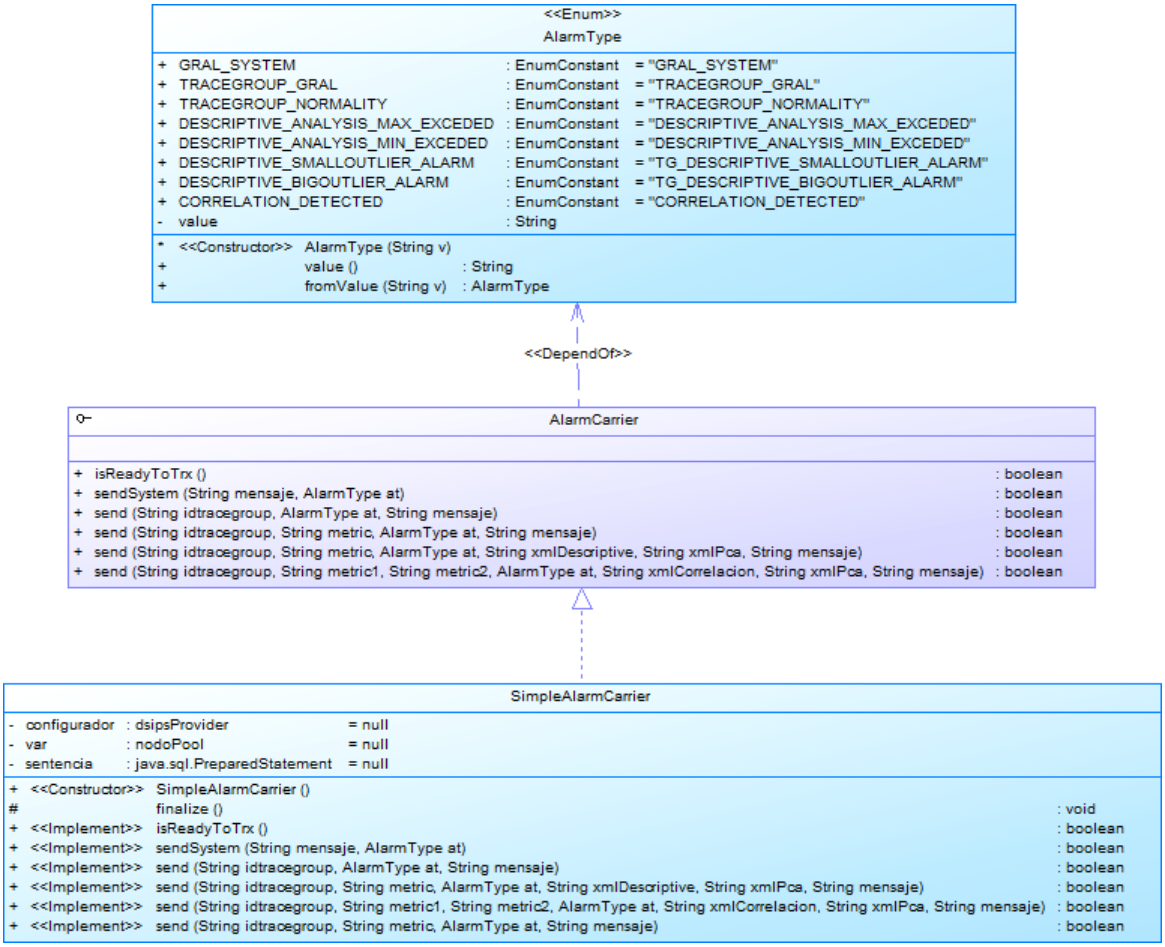


Figura 61. Interface AlarmCarrier y sus Principales Relaciones

El método *loadBlockedAlarm* de *DecisionMaker*, permite cargar y/o actualizar desde la parametría del EIPFDcMM, cuáles son los tipos de alarmas permitidas y cuales estarán bloqueadas indistintamente de que ocurra el evento. Esta información, es actualizada dinámicamente cada cierto período (dado por la propiedad *hilo_update_ms*) por lo que no requiere re inicialización de los servicios del EIPFDcMM.

Los métodos que comienzan con *send*, son métodos que implementan diferentes tipos de alarmas a disparar a través de la instancia de *DecisionMaker*. No obstante, el hecho de ejecutar un método de envío de alarma desde el tomador de decisiones, no implica necesariamente que ésta será exteriorizada, dado que el mismo posee un conjunto de parámetros que determinan, a partir de la base de datos C-INCAMI del proyecto de M&E, qué tipos de alarmas disparar y cuáles no.

El método *sendSystem* de *DecisionMaker,* permite notificar alarmas (o eventos) de sistema, como por ejemplo, cuestiones asociadas a la disponibilidad de memoria. El método *sendAlarm(String idtracegroup,String mensaje),* permite emitir una alarma asociada al grupo de seguimiento (por ej.: "Juan Perez"), por ejemplo cuando este presenta un lapso 'x' de tiempo sin informar mediciones. El método *sendNormalityAlarm,* permite informar que se ha detectado distribución normal en un conjunto de métricas dadas, dentro de un grupo de seguimiento. El método *sendAlarm(String idtracegroup,String metric,AlarmType at,String xmlDescriptive,String xmlPCA,String mensaje),* permite informar un evento originado en el análisis descriptivo, al cual se le adjunta información del mismo junto con el análisis de componentes principales, para contextualizar la variabilidad del sistema donde se origina la alarma. El método *sendAlarm(String idtracegroup,String metric1,String metric2, AlarmType at,String xmlCorrelacion,String xmlPCA,String mensaje),* permite informar un evento originado en el análisis de correlación al cual, en forma análoga al descriptivo, se le adjunta el análisis de componentes principales para contextualizar la variabilidad del sistema al momento en que detecta el vínculo o correlación entre las métricas indicadas. De los métodos de la clase *DecisionMaker*, el subconjunto expuesto en la Figura 60 y los aquí mencionados, son los que en definitiva emplea el analizador estadístico.

Los tipos de alarmas que pueden ser disparados, acotados a la funcionalidad del análisis estadístico, son modelados mediante la enumeración *AlarmType* y son explicados en la Tabla 11.

Tipo	*Concepto*
GRAL_SYSTEM	*Alarma General del Sistema*
TRACEGROUP_GRAL	*Alarma General Asociada a un Grupo de Seguimiento*
TRACEGROUP_NORMALITY	*Detección de distribuciones normales dentro de un grupo de seguimiento.*
DESCRIPTIVE_ANALYSIS_MAX_EXCEDED	*Se ha detectado que la serie de datos de una métrica dentro de un grupo de seguimiento, ha excedido el máximo definido dentro del proyecto de medición.*
DESCRIPTIVE_ANALYSIS_MIN_EXCEDED	*Se ha detectado que la serie de datos de una métrica dentro de un grupo de seguimiento, ha excedido el mínimo definido dentro del proyecto de medición.*
DESCRIPTIVE_SMALLOUTLIER_ALARM	*Informa la presencia de pequeños outliers en una métrica dentro de un grupo de seguimiento.*
DESCRIPTIVE_BIGOUTLIER_ALARM	*Informa la presencia de grandes outliers en una métrica dentro de un grupo de seguimiento.*
CORRELATION_DETECTED	*Informa que se ha detectado correlación entre métricas dentro de un grupo de seguimiento*

Tabla 11. Tipos de Alarmas Disponibles para el Análisis Estadístico

Los tipos de alarmas son empleados tanto por el usuario de los métodos de *DecisionMaker,* como así también por cada instancia de la clase, a través de la notificación cuando corresponda realizarla. De este modo, la clase que implemente la interface *AlarmCarrier* dependerá directamente de la enumeración propuesta en *AlarmType.* Debe destacarse en este último sentido, que la presente sub-sección está circunscripta al análisis estadístico, por lo que la funcionalidad analizada es parcial al uso que el mismo requiere y puede ser extendida sin inconvenientes.

La interface *AlarmCarrier* (ver Figura 61) propone diferentes métodos send (circunscriptos a la funcionalidad requerida para el análisis estadístico), requeridos para satisfacer las necesidades

de comunicación del tomador de decisiones, ante la necesidad de informar un evento o disparar una alarma. Como ejemplo de ello, el EIPFDcMM plantea una clase *SimpleAlarmCarrier* (ver Figura 61), que implementa la interface y registra todas las alarmas dentro de una tabla denominada *alarmevent,* dentro de la BD C-INCAMI. Ello tiene por objetivo, disponer de un registro de las mismas en forma persistente y con fines de prueba. No obstante, es posible que la clase que implemente la interface *AlarmCarrier*, además de dejar registro en el sistema, desee enviar un e-mail y/o un SMS al director del proyecto al mismo tiempo, lo cual es absolutamente posible. Por tal motivo se planteó de este modo el esquema de notificación. La idea desde este último punto de vista, es que quedase expresada la idea del notificador y su objetivo, independientemente de cómo hacer llegar la notificación al o los destinatarios, lo cual fue pensado no sólo como un aspecto de implementación sino también de extensibilidad.

7.11 Parametría Asociada al Analizador Estadístico

El analizador estadístico, a detallar en la sub-sección 7.12, puede configurar su comportamiento a partir de un subconjunto de parámetros que son almacenados en forma persistente dentro de la base de datos C-INCAMI. Es importante mencionar, que no toda la parametría se aplica en términos generales sobre el análisis estadístico, sino que también, es posible indicar una parametría particular a un grupo de seguimiento (por ej.: a un paciente en particular, "Luis García") dentro de un proyecto de M&E. La Tabla 12, indica cada uno de los parámetros de configuración asociados al análisis estadístico, que rigen el comportamiento del tomador de decisiones (instancia de la clase *DecisionMaker*).

Parámetro	Tipo	Concepto
SYSTEM_ALARM	Boolean	Indica si DecisionMaker permitirá el tránsito y despacho de alarmas del sistema. Por defecto es TRUE.
NORMALITY_ALARM	Boolean	Indica si DecisionMaker permitirá el tránsito y despacho de alarmas, ante la detección de variables con distribución normal. Por defecto es TRUE.
STATISTICAL_ALARM	Boolean	Indica si DecisionMaker permitirá el tránsito y despacho de alarmas, surgidas del análisis estadístico. Por defecto es TRUE.
TG_GRAL_ALARM	Boolean	Indica si DecisionMaker permitirá el tránsito y despacho de alarmas generales, asociadas a grupos de seguimientos. Por defecto es TRUE.
TG_DESCRIPTIVE_RESUME_ALARM	Boolean	Indica si el DecisionMaker permitirá el tránsito y despacho de alarmas, surgidas del análisis descriptivo de la serie de datos asociada a una ventana (Window) del data stream para un Grupo de Seguimiento (TG) dado. Por defecto es TRUE.
TG_DESCRIPTIVE_SMALLOUTLIER_ALARM	Boolean	Indica si el Decision Maker permitirá el tránsito y despacho de alarmas, surgidas de la identificación de pequeños outliers en el análisis descriptivo a partir de la serie de datos asociada a una ventana (Window) del data stream para un Grupo de Seguimiento (TG). Por defecto es TRUE.
TG_DESCRIPTIVE_BIGOUTLIER_ALARM	Boolean	Indica si el Decision Maker permitirá el tránsito y despacho de alarmas, surgidas de la identificación de grandes outliers en el

		análisis descriptivo a partir de la serie de datos asociada a una ventana (Window) del data stream para un Grupo de Seguimiento (TG). Por defecto es TRUE.
TG_CORRELATION_ALARM	Boolean	Indica si el Decision Maker permitirá el tránsito y despacho de alarmas surgidas, del análisis de correlación de la serie de datos asociada a una ventana (Window) del data stream para un Grupo de Seguimiento (TG). Por defecto es TRUE.
STATISTICAL_USE_SMOOTHER	Boolean	Indica si el analizador estadístico suavizará los datos recibidos desde el buffer, posterior a su análisis y previo a la puesta a disposición de los mismos a los clasificadores. Por defecto es TRUE.
STATISTICAL_SMOOTHER_TYPE	String	El parámetro solo tiene validez si STATISTICAL_USE_SMOOTHER está activo. Actualmente este parámetro puede asumir los siguientes valores: NONE, NULL_BY_MEDIA, NULL_BY_MEDIANA, NULL_BY_Q1, NULL_BY_Q3, NULL_BY_MIN y NULL_BY_MAX. El valor por defecto es NONE. Los valores posibles del parámetro se corresponden con los tipos de suavizadores explicados en la Tabla 10

Tabla 12. Parámetros de configuración que rigen a la clase DecisionMaker y están asociados al Análisis Estadístico

El parámetro *TG_CORRELATION_ALARM,* indica si se permitirá el tránsito y despacho de alarmas vinculadas al análisis de correlación. Puede ocurrir, que posiblemente interese tener activo este tipo de alarma pero filtrar aquellas asociadas a determinados pares de métricas, que a priori se sabe, están correlacionadas (por ejemplo, si sobre el paciente se tomase, en forma adicional a la temperatura axilar, la temperatura lingual, debiera bloquearse el informe de correlación entre ambas, por lo que a priori se sabe que dispondrán de un comportamiento análogo, por ende correlacionado, y ello no aportaría mayor información sobre el estado del paciente). Esto mismo, es posible mediante la especificación de las mismas dentro de la tabla *blockedalarmtgcorrelation* de la base de datos C-INCAMI de los proyectos de M&E, en donde se debe indicar básicamente el identificador del grupo de seguimiento y los dos identificadores asociados a las métricas. Por lo tanto, el bloqueo actúa *sólo* sobre dicho par de métricas y *únicamente* dentro del grupo de seguimiento indicado. Estos bloqueos son cargados y actualizados periódicamente por el tomador de decisiones, para definir si da curso o no a este tipo de eventos.

Los parámetros asociados al análisis descriptivo, indican si se permitirá el tránsito y despacho de alarmas vinculadas al exceso de máximo, mínimos o la presencia de outliers. Ocurre también en estos casos, que posiblemente interese una métrica desde el punto de vista de su relación con otras métricas, y no desde su análisis descriptivo en sí mismo. En esta última situación, se podría desear bloquear las alarmas descriptivas sobre una métrica particular. Esto es posible mediante la especificación de las mismas dentro de la tabla *blockedalarmdescriptiveresume* dentro de la base de datos C-INCAMI de los proyectos de M&E. Allí, se debe indicar el identificador del grupo de seguimiento y el identificador de la métrica. De este modo, el bloqueo actúa *sólo* sobre dicha métrica y *únicamente* dentro del grupo de seguimiento indicado. Estos bloqueos, son cargados y actualizados periódicamente por el tomador de decisiones para definir si da curso o no a este tipo de eventos.

En forma análoga al análisis estadístico, es posible que no interese recibir alarmas de identificación de normalidad de todas las variables. Para evitar ello, es posible indicar todas las métricas de las cuales no interese recibir este tipo de alarmas, incorporándolas en la tabla *blockedalarmtgnormality* de la base de datos C-INCAMI de los proyectos de M&E. En ella se deberá especificar, el identificador del grupo de seguimiento y el de la métrica. De igual modo que en los casos anteriores, el bloqueo actúa *sólo* sobre dicha métrica y *únicamente* dentro del grupo de seguimiento indicado.

7.12 Analizador Estadístico

La funcionalidad del analizador estadístico, es implementada mediante la clase *StatisticalAnalyzer* (ver Figura 62). La responsabilidad de la clase *StatisticalAnalyzer,* comienza con la obtención de la instantánea del conjunto de mediciones a partir del buffer de la función de reunión (ver sub-sección 6.5), y culmina con la realización de los análisis estadísticos mencionados anteriormente, con la aplicación de suavizados de datos previos si correspondiese (en base a la parametría) y con la emisión de las alarmas pertinentes cuando corresponda hacerlo.

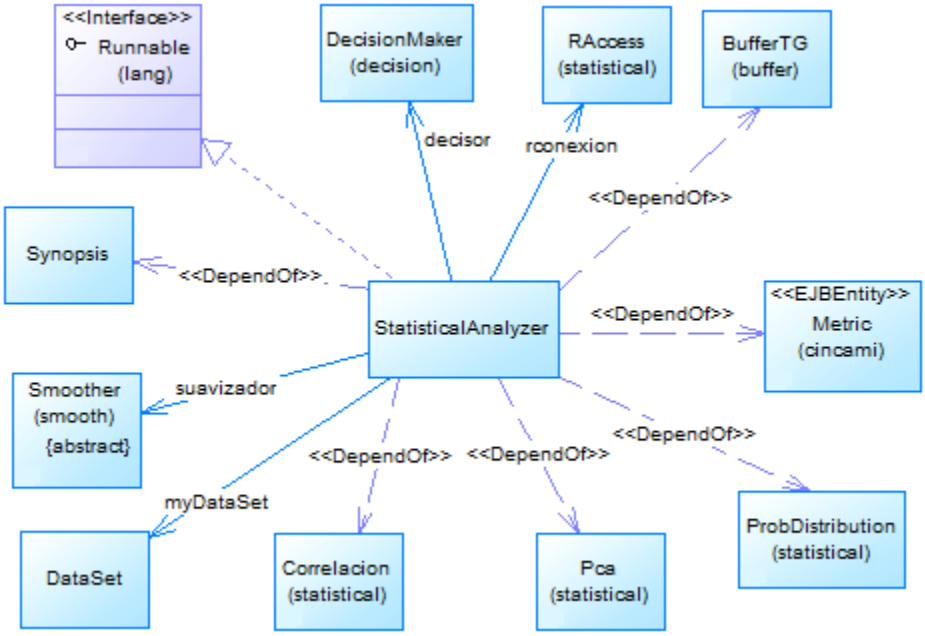


Figura 62. Principales dependencias y relaciones de StatisticalAnalyzer

Como puede observarse en la Figura 62, el analizador estadístico depende de la clase *BufferTG,* ya que debe tener acceso al buffer de mediciones para generar una copia instantánea de las mismas para su análisis posterior. Debe disponer de conexión al motor de cálculo estadístico, y proveer éste, a cada una de las clases que lo requieran coordinando la ejecución del análisis estadístico global. Posee dependencia de las clases *Pca*, *Correlacion*, *ProbDistribution* y *Synopsis,* lo cual es lógico dado que son las que implementan cada uno de los análisis estadísticos en particular. Se asocia con la clase *DataSet,* lo cual permite generar la mencionada instantánea a partir del buffer de la función de reunión y de este modo, organizarlo por grupos de seguimiento.

Se relaciona con la clase *Smoother*, dado que posee la definición del tipo de política de suavizado a aplicar sobre los datos, previo a informarlos a los modelos de clasificación dentro del proceso de toma de decisiones (ver sub-sección 3.4). Es posible, que la propiedad *suavizador* de la clase *StatisticalAnalyzer* sea nulo, lo que indica que no se aplica política de suavización alguna. De este modo y, analizado las funcionalidades estadísticas que se implementan, no es una novedad que el analizador estadístico dependa de la clase *Metric* (según fue definida dentro del proyecto de M&E), ya que es el único mecanismo lógico, explícito y formal contra el cual contrastar las mediciones vinculadas. La clase *StatisticalAnalyzer*, implementa la interface *Runnnable*, ya que mediante la misma define la periodicidad y el modo en que irá actualizando la instantánea de las mediciones, efectuando el análisis y generando las alarmas pertinentes.

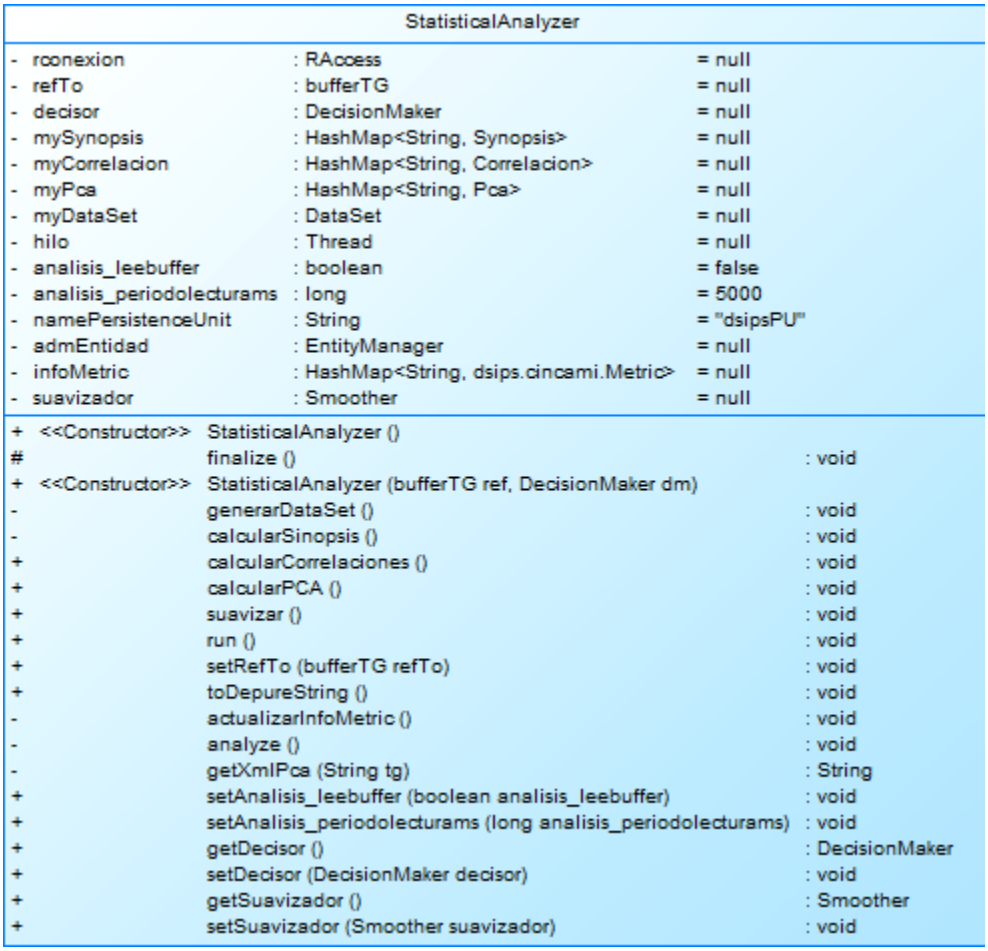


Figura 63. Clase StatisticalAnalyzer

Como propiedad de la clase *StatisticalAnalyzer* (ver Figura 63) pueden encontrarse la sinopsis conteniendo el análisis descriptivo por métrica y grupo de seguimiento, el análisis de correlación (organizado por grupo de seguimiento), el análisis de componentes principales (organizado también por grupo de seguimiento); una pregunta que podría hacerse el lector es: ¿porqué almacenarlos si debo re-calcularlos o re-analizarlos cuando se actualicen los datos? La respuesta desde el punto de vista del flujo de datos es que simplemente, *debe* mantenerse

registros que permitan explicar el comportamiento del grupo (por ej.: del paciente “Juan Perez”) hasta tanto puedan ser reemplazados por otros más recientes, en la medida en que existan. Esto último es fundamental, porque si ante una situación determinada, un grupo de seguimiento deja de recibir mediciones en el buffer, se debe disponer de una síntesis que permita responder a consultas sobre el mismo lo más actualizado posible, y así se estaría dando *una respuesta aproximada del mismo pero respuesta al fin*. Está claro, que contar con la última información conocida y sintetizada, siempre será mejor desde el punto de vista del proceso de toma de decisiones que no disponer de información alguna, ya que la comparación es concreta “*alguna información conocida versus incertidumbre*”. No obstante, la información conocida que permite responder ante situaciones de ausencia de datos, debe tener un límite finito por cuanto los recursos de procesamiento no son ilimitados. Por esto último, el analizador estadístico solo retiene las sinopsis, los resultados del análisis de correlación y los resultados del análisis de componentes principales. Estos últimos, permiten describir la serie de datos para una métrica dada, indicar el tipo de vínculo entre métricas (o variables) y pronunciarse sobre la variabilidad global del sistema.

8 Proceso de Toma de Decisión

En el capítulo 7 se ha discutido el motor de cálculos estadísticos y su rol en la función de suavización. Se analizó el concepto de operación estadística para luego, discutir el análisis de correlación, el análisis de componentes principales, el análisis descriptivo y las pruebas paramétricas/no paramétricas sobre las distribuciones de probabilidad que pudiesen tener los flujos de datos. A seguir, se ha analizado el concepto de sinopsis, y se ha discutido el rol de los suavizadores frente al flujo de datos y su relación con las instantáneas. Se analizó además el rol del tomador de decisiones y los notificadores dentro del análisis estadístico. Finalmente, se discutió el analizador estadístico en forma conjunta con la parametría que rige su comportamiento.

En este capítulo, se discute el contexto de procesamiento y cómo éste afecta a la clasificación de flujos de datos. Además, se discute el análisis masivo en línea (Massive On line Analysis –MOA-) y su relación con el procesador de toma de decisiones.

Luego, se analizan los algoritmos presentados en la sub-sección 2.5 y se fundamenta la elección del algoritmo de clasificación sobre flujos de datos que utiliza el EIPFDcMM. Además, se analiza la estrategia mediante la cual, el EIPFDcMM, aplica los clasificadores en línea, a los grupos de seguimiento como herramienta de soporte de decisión. A continuación, se discute la fase de entrenamiento de los clasificadores, dado que debe considerarse, como se introdujo en la sub-sección 2.5, que los árboles de decisión son métodos supervisados, por lo que requieren de un entrenamiento previo a su empleo efectivo.

Por otra parte, se discute el efecto de la parametría del proyecto de M&E en la base de datos C-INCAMI, en el entrenamiento como al momento de tomar una decisión.

Finalmente, se analiza el tomador de decisiones junto con los notificadores, que a diferencia del capítulo 7, aquí se extiende con el comportamiento propio del análisis de las decisiones dadas por los clasificadores.

8.1 Contexto de Procesamiento y Clasificación en Flujos de Datos

Básicamente, el proceso de toma de decisión obtiene la ventana de datos suavizada junto con su información estadística de resumen, análisis de correlación y el análisis de componentes principales desde el proceso de análisis estadístico, y a partir de allí, nutre a los clasificadores para que produzcan una decisión. Tales decisiones, son informadas al tomador de decisiones, para que éste defina el curso de acción a tomar. La información que obtienen los clasificadores, se caracteriza por contar no solo con datos propios de las métricas asociadas a la entidad bajo análisis (por ej.: el valor de la frecuencia cardíaca del paciente "Juan Perez"), sino que también cuenta con la información de métricas asociadas al contexto de la entidad (por ej.: el valor de la temperatura ambiental asociada al entorno del paciente "Juan Perez"). De este modo, el proceso de toma de decisión puede analizar la entidad bajo análisis, como objeto de decisión circunscripto e incidido por su contexto inmediato.

El contexto de procesamiento y clasificación de flujos de datos, tal como se discutiera en las sub-secciones 2.4 y 2.5, parte de una serie de requerimientos fundamentales que inciden directamente en la estrategia de procesamiento. Estos requerimientos, plantean una serie de condicionantes al procesamiento y clasificación de flujos de datos según (Kirkby, 2007), entre las que se enuncian:

a. **Los datos poseen un número finito de columnas o atributos.** Tal hipótesis supondría que un flujo de mediciones posee una cantidad finita de métricas (o variables) definidas. En este sentido, la cantidad de métricas definidas en un proyecto de M&E podrá ser tan grande como se quiera, pero finita al fin. Por ejemplo, para el paciente trasplantado se definieron el valor de la temperatura ambiental, el valor de la temperatura axilar, el valor de la humedad ambiente, entre otros presentados en el caso de aplicación de la sub-sección 3.5.

b. **El número de filas asociados es muy extenso.** De este modo, los algoritmos deben tener la capacidad de procesar una cantidad infinita de datos, sin que se exceda la memoria disponible y/o los recursos de procesamiento. Este aspecto, será profundizado en el capítulo 9, a través del análisis de la simulación EIPFDcMM sobre el caso de aplicación presentado en la sub-sección 3.5.

c. **El dato tiene un número limitado de posibles etiquetas de clases, en forma típica menos de diez**. La etiqueta de clase, con respecto al EIPFDcMM, representa la tipificación de las situaciones posibles que deberá clasificar el tomador de decisiones, ante el arribo de nuevas instancias y sobre las cuales es entrenado previamente. En este sentido, quienes toman un rol preponderante, son los criterios de decisión asociados a cada indicador sobre el que se basará el curso de acción a seguir por el tomador de decisiones (por ejemplo, ver el asociado con el nivel de temperatura corporal para el paciente trasplantado definido en la Tabla 1).

d. **La cantidad de memoria disponible para el algoritmo de aprendizaje depende de la aplicación.** El tamaño del conjunto de datos para entrenamiento, debiera ser considerablemente mayor a la memoria disponible. Esto es así, dado que los datos de entrenamiento son guiados y seleccionado por expertos, representando la información necesaria y tipificada, de las situaciones que se desean detectar mediante el clasificador. De este modo, el conjunto de instancias[2] con sus correspondientes etiquetas de clase[3], pueden exceder la memoria disponible por el motivo enunciado y adicionalmente, requerir un adicional, dado que la etapa de entrenamiento es la que genera el modelo de clasificación a utilizar. No obstante, una vez que el modelo se encuentra en producción, el mismo solo contará con los recursos de procesamiento y memoria disponibles, por lo que deberá circunscribirse a ellos para evitar incursionar en un desborde (*overflow)* de recursos.

e. **El tiempo de clasificación o re-aprendizaje es limitado**. Existe una cota máxima en cuanto al tiempo que un algoritmo podrá utilizar, para clasificar una nueva instancia o bien, para re-entrenarse (o ajustarse) a partir de una instancia conocida. La idea de este punto, es garantizar la escalabilidad del procesamiento a medida que se incrementa el volumen del flujo de datos.

A partir de las condicionantes planteadas, en (Kirkby, 2007) se plantea como corolario, una serie de nuevos requerimientos, que inciden directamente en los algoritmos de clasificación:

1. **Se debe procesar un ejemplo a la vez e inspeccionarlo como máximo una vez.** Este requerimiento restringe la lectura de los casos a los efectos de garantizar que el peor orden de un algoritmo dado será el lineal. De dicho modo, se define una cota superior a la tasa de crecimiento del orden de procesamiento, en pos de garantizar la escalabilidad, considerando que el volumen de datos que arriba para su procesamiento es enorme y posiblemente ilimitado.

2. **Utilizar una cantidad limitada de memoria.** La idea de limitar la cantidad de memoria a utilizar para la clasificación está relacionada con el requerimiento ‘1’. Es decir, el hecho de definir una cantidad finita de memoria factible de utilización por el algoritmo, tiene por objetivo garantizar, además de la escalabilidad, la estabilidad ante el incremento del volumen de datos.

3. **Trabajar en una cantidad limitada de tiempo.** El requerimiento ‘1’ indicaba que un ejemplo debía leerse a lo más una vez, y ello, sumado al actual requerimiento de limitar la

[2] Se entiende por instancia en EIPFDcMM, a la unidad lógica de medición (ver sub-sección 4.3.3) que tiene a priori, asociado un nivel determinado dentro del criterio de decisión de un indicador dado.
[3] Se entiende por etiqueta de clase en EIPFDcMM, a un nivel determinado de un criterio de decisión de un indicador dado.

cantidad de tiempo para procesar un ejemplo, tiene por objetivo establecer un patrón de crecimiento predecible y estable.

4. **Estar listo para predecir en cualquier momento.** Dado que en un flujo de datos, el arribo de un nuevo dato puede ocurrir en cualquier momento, el clasificador debe ser capaz de actuar cuando sea necesario.

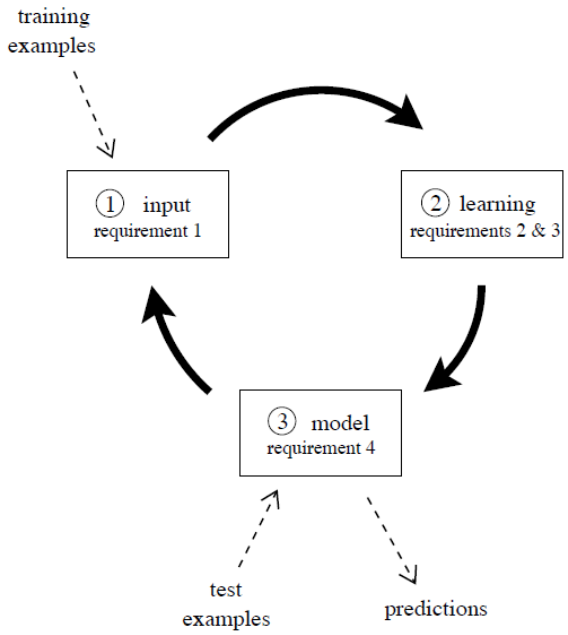


Figura 64. Ciclo de Clasificación de un Flujo de Datos (Kirkby, 2007)

Cada uno de los cuatro requerimientos planteados en (Kirkby, 2007), se asocian con una etapa en particular del ciclo de clasificación de flujos de datos (ver Figura 64). El hecho de limitar la lectura e inspección de cada nueva unidad lógica de medición a una vez, tiene por finalidad evitar la congestión y los posteriores cuellos de botella, y lograr un mejor aprovechamiento de los recursos disponibles (es por eso que se vincula al ingreso de los flujos de datos). La idea de limitar los recursos a emplear y trabajar en una fracción de tiempo dada, es lógico que se asocie a la etapa de aprendizaje, por cuanto el tiempo a aplicar en estudiar un nuevo caso debe ser tal, que le permita tomar ventaja al modelo actualizado, para con la siguiente unidad lógica de medición que arribe. Finalmente, el cuarto requerimiento, se asocia a la etapa de modelado, por cuanto se está exigiendo que el modelo pueda clasificar en cualquier momento, y esto se vincula esencialmente con la propia naturaleza de arribo de los flujos de datos.

8.2 Análisis Masivo en Línea

El Análisis Masivo en Línea (Massive Online Analysis -MOA-) (Bifet, Holmes, et al., 2010), es un entorno de software escrito completamente en JAVA, que implementa algoritmos y experimentos para aprendizaje online sobre flujos de datos. El proyecto MOA, desarrollado por los autores en el Departamento de Ciencias de la Computación de la Universidad de Waikato, Nueva

Zelanda, se encuentra relacionado con el proyecto WEKA (Frank & Witten, 2005), permitiendo la interacción bidireccional entre ambos.

El proyecto MOA, incluye una colección de métodos online y offline, así como herramientas de evaluación. En particular, implementa boosting, bagging y Hoeffding Trees, todos ellos con o sin el enfoque Naive Bayes en las hojas.

El EIPFDcMM utiliza las implementaciones de MOA para llevar adelante las tareas de entrenamiento y ejecución de los clasificadores. De este modo, durante la inicialización del EIPFDcMM, se toman los conjuntos de datos indicados para entrenamiento desde la base de datos C-INCAMI, se entrenan los clasificadores según las directrices de MOA, y los mismos quedan inmediatamente habilitados para clasificar nuevas instancias en cualquier momento, informando de tal situación al tomador de decisiones para que defina los cursos de acción a tomar.

8.3 Selección del Algoritmo de Clasificación sobre Flujos de Datos

En esta tesis, a los efectos de fundamentar la selección del algoritmo de clasificación a utilizar en el proceso de toma de decisión, se parte del análisis comparativo realizado por (Bifet, Holmes, et al., 2009). El citado análisis, ha desarrollado una destacada evaluación de algoritmos, a partir de conjuntos de datos reales, con un volumen de hasta 10 millones de ejemplos, demostrando que los nuevos métodos de conjunto, se ejecutan muy bien comparados con los tradicionales.

Un aspecto destacado de dicho análisis, es que contiene la totalidad de los algoritmos discutidos en la sub-sección 2.5, sin limitarse a ellos. A modo de ejemplo, el estudio permite cuantificar el empleo de memoria, tasas de procesamiento, entre otros aspectos aplicados a algoritmos tales como: ADWIN Bagging and Adaptative-Size Hoeffding Tree (ASHT), Very Fast Decision Tree Learner (VFDT), Very Fast Decision Tree – C Learner (VFDTc), Very Fast Decision Tree Learner (VFDT) y Numerical Interval Pruning (NIP), Streaming Ensemble Algorithm (SEA), Instante-Based Ensemble Pruning, Concept-Adapting Very Fast Decision Tree (CVFDT), entre otros. El ambiente de software utilizado para realizar las pruebas, en el citado estudio, es MOA. A partir de los resultados de las pruebas y las conclusiones en (Bifet, Holmes, et al., 2009), se destaca que:

- ASHT con Bagging, logran en general una exactitud superior a 80%, dependiendo del data set y la variación de conceptos, con un consumo de memoria que no alcanzó los 4 mega bytes (Mb) en ningún momento, y con tiempos de procesamiento que no superaron los 1200 segundos para procesar 10 millones de instancias.

- Naive Bayes, posee un uso muy eficiente de memoria y del tiempo de procesamiento (incluso mejor que ASHT con Bagging), pero a costa de sacrificar exactitud, dado que la misma oscila entre 55% y 73%. Notar que al momento de decidir sobre dos situaciones posibles, si se contara con una probabilidad similar a 55%, casi que sería recomendable no clasificar y simular el disparo de una moneda, ya que se ahorraría notablemente recursos de memoria y procesamiento, sin sacrificar demasiada exactitud.

- El método de Oza Boosting, posee un rango de exactitud que oscila entre 58% y 94%, pero tiene un consumo excesivo de memoria, y para el mismo conjunto de datos, llega a consumir 206 Mb frente a los 4Mb consumidos por ASHT con Bagging.

En el estudio de (Bifet, Holmes, et al., 2009), se analizan numerosos métodos en términos de consumo de memoria, exactitud y tasa de procesamiento, los tres pilares fundamentales que rigen la elección de algoritmos en clasificación de flujos de datos, tal y como se discutió en la sub-sección 8.1. Los algoritmos mencionados, se corresponden con los casos extremos y el caso intermedio del estudio citado, a los efectos de ejemplificar el mismo y no tornarse repetitivo. Finalmente, el mejor equilibrio global entre capacidad de procesamiento, consumo de memoria y estabilidad es obtenido por ASHT con Bagging, por lo que éste ha sido el algoritmo seleccionado en esta tesis para implementar la función de clasificación mediante árboles incrementales.

8.4 Clasificador y Acciones

Como se ha mencionado, EIPFDcMM emplea MOA para implementar los clasificadores que actuarán sobre los flujos de mediciones. De este modo, será necesaria una serie de adaptaciones para poder lograr tal acoplamiento.

La clase *Classifier* en MOA, representa la clase base para los clasificadores a través de la cual, independientemente del algoritmo seleccionado para llevar adelante la funcionalidad, se puede referenciar a cualquier implementación derivada mediante polimorfismo. De este modo, el EIPFDcMM genera un clasificador por grupo de seguimiento, pudiendo cada grupo de seguimiento disponer de implementaciones diferentes (por ejemplo, el clasificador asociado al paciente “Juan Perez” se encuentra *adaptado* a su comportamiento específico, a partir de sus mediciones específicas y por ende, no será el mismo clasificador que se empleará para “Luis García”, quien dispondrá de otro clasificador ajustado a su comportamiento). Así se logra abstraer al EIPFDcMM de aspectos de implementación, disponiendo de los métodos fundamentales indicados por la clase base Classifier. Tales métodos, permitirán llevar adelante la clasificación independientemente de la implementación específica elegida.

La clase *TraceGroupClassifier* del EIPFDcMM (ver Figura 65), es la responsable de asociar a cada grupo de seguimiento un clasificador (por ejemplo, asociará a “Juan Perez” un clasificador que se ajustará a su comportamiento específico, en función de sus mediciones), independientemente de qué implementación (algoritmo) específica posea este. Por lo tanto, el EIPFDcMM propone que cada grupo posea su propio clasificador, con el objetivo de lograr un mejor ajuste a la situación contextual del grupo de seguimiento como de sus particularidades. Adicionalmente, la clase *TraceGroupClassifier* a través del método *setClassifier*, permite alternar entre diferentes implementaciones de clasificadores dinámicamente. Al momento de reemplazar el clasificador para un grupo de seguimiento, se realiza el entrenamiento del nuevo clasificador, y se emplea la antigua versión hasta tanto el nuevo clasificador culmine su entrenamiento.

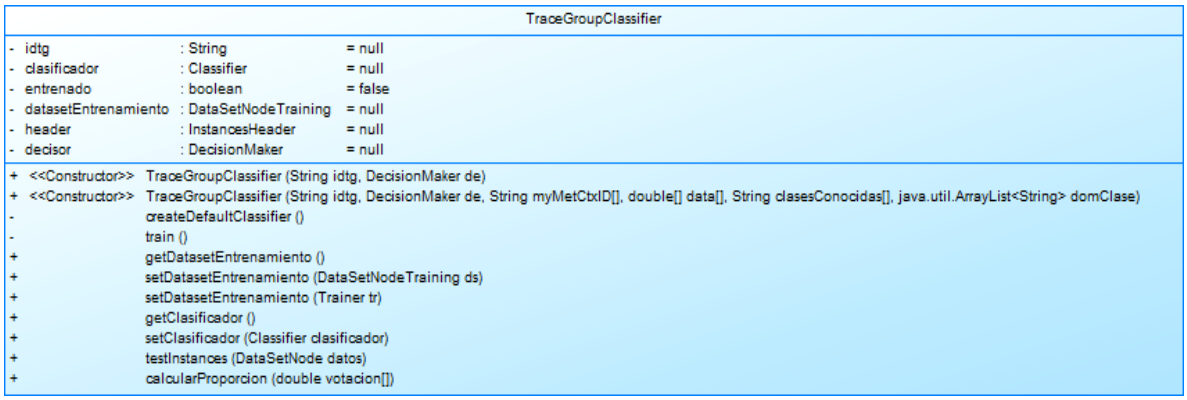


Figura 65. Clase TraceGroupClassifier

En caso de que un grupo de seguimiento no especifique que implementación de clasificador utilizará, la instancia de *TraceGroupClassifier* crea, a través del método *createDefaultClassifier*, un clasificador con la implementación de Adaptative-Size Hoeffding Tree (ASHT) con Bagging, quedando a la espera del grupo de entrenamiento para poder ser entrenado e inicializado, para su posterior puesta en producción.

El entrenamiento del clasificador se realiza informando a la clase *TraceGroupClassifier* la instancia de *DataSetNodeTraining*, que contiene el conjunto de datos de entrenamiento, mediante el método *setDataSetEntrenamiento*. Adicionalmente, la clase *TraceGroupClassifier,* contempla la ejecución del entrenamiento, a través de una instancia que derive de la clase abstracta *Trainer.* Dicha clase, será abordada en la sub-sección 8.5 y sintéticamente, contempla los métodos básicos que deben satisfacerse para poder entrenar a un clasificador, independientemente de donde provengan los datos de entrenamiento.

La clase *TraceGroupClassifier,* exige en ambos constructores, que se indique la instancia de *DecisionMaker*. Este aspecto es fundamental, por lo que el clasificador comunica en forma instantánea, cualquier decisión a la que arribe a la instancia de *DecisionMaker*, para que de acuerdo a su parametría, resuelva el curso de acción a tomar.

La responsabilidad central de la clase *TraceGroupClassifier,* radica en el método *testInstance*. El método recibe una ventana de datos mediante una instancia de DataSetNode, la cual agrupa las mediciones para un grupo de seguimiento (por ejemplo, un conjunto de mediciones del paciente "Juan Perez" que contempla tanto información de sus atributos como de su contexto), dentro del formato en que utiliza la clase *StatiscalAnalyzer* (ver sub-sección 7.8). A partir de ella, clasifica la totalidad de los casos (por ejemplo, a partir de las mediciones de "Juan Perez" y en base a la definición de los indicadores elementales, infiere si el estado actual del paciente, representaría en el futuro cercano un potencial riesgo para el mismo), e informa a la instancia de *DecisionMaker* cada decisión, a los efectos que resuelva el curso de acción a seguir.

Un aspecto importante a destacar, es que en caso de que la instancia de *TraceGroupClassifier* no esté entrenada, no tenga clasificador definido o bien, al momento de pedir que clasifique se encuentre en entrenamiento, la misma no bloqueará el procesamiento

global del EIPFDcMM, sino que omite su participación y permite que el flujo continúe su curso. Esto se sustenta, en el hecho de que si bien existe la imposibilidad de clasificar un caso, dado que el entrenamiento del clasificador no ha sido terminado, existe un monitoreo de la medición con respecto a su definición formal, y un análisis estadístico sobre su evolución, que dispararían alarmas preventivas en caso de detectar situaciones de arrastre, correlación o anomalía formal.

8.5 Entrenamiento

Los árboles de clasificación son métodos supervisados. Estos son así denominados, por cuanto requieren de un entrenamiento previo a su empleo efectivo en producción. En el entrenamiento, se parte de una proporción del conjunto de datos de los que se conoce su etiqueta de clase, para entrenar el modelo. La proporción remanente de datos, se emplea para validar la capacidad predictiva del modelo obtenido, dado que éste último nunca ha tenido contacto con tales datos (dado que no han sido incluidos en el entrenamiento). Si bien éste es el mecanismo de entrenamiento que se empleará para la inicialización de los clasificadores en EIPFDcMM, existen otras alternativas tales como cross-validation, discutidas en (Zhong, Fan, et al., 2010).

La clase *TraceGroupClassifier* (Figura 65), contempla el entrenamiento a partir de una instancia *DataSetNodeTraining* o bien de una instancia derivada de la clase abstracta *Trainer*. La clase *DataSetNodeTraining,* para poder entrenar a un clasificador del framework MOA, debe implementar la interface *MOAObject* e *InstanceStream (ambas* pertenecientes al framework MOA). La interface *MOAObject*, define los métodos de copia de objetos y la estimación de su tamaño en memoria; mientras que la interface *InstanceStream*, define los métodos para conceptualizar un conjunto de datos como una ventana de procesamiento (Window –ver sub-sección 2.1.2-).

A los efectos de simplificar la interoperabilidad con EIPFDcMM y, considerando que el conjunto de entrenamiento es una especialización del conjunto de datos de un grupo de seguimiento, la clase *DataSetNodeTraining* (ver Figura 66) hereda su comportamiento de *DataSetNode,* e incorpora dos elementos fundamentales para el entrenamiento: el conocimiento de la clase a la que corresponde cada caso y el dominio posible de valores de la clase objeto de decisión. Con estos elementos definidos, *DataSetNodeTraining* simplemente tiene que ser pasado como parámetro a través del método *setDataSetEntrenamiento* de *TraceGroupClassifier* para que el clasificador sea entrenado.

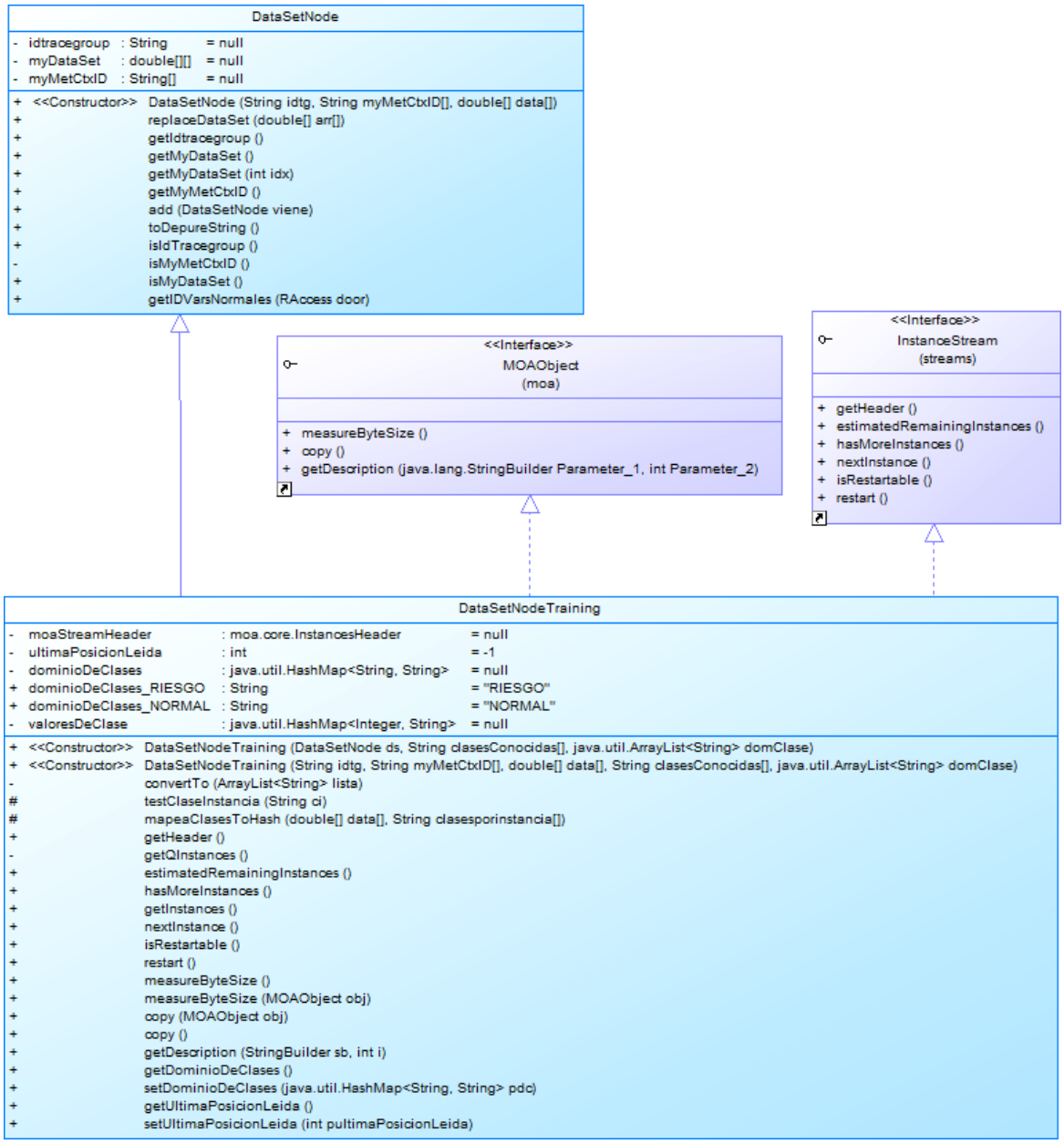


Figura 66. Principales Relaciones de la Clase DataSetNodeTraining.

La clase *DataSetNodeTraining* implementa la interface MOAObject a los efectos de poder serializar sus instancias, cuando así lo requieran las implementaciones del framework MOA. Además, el hecho de implementar la interface *InstanceStream* de MOA, se corresponde con del contexto de clasificación. Esto es, la clasificación se lleva adelante sobre flujos de datos y online, por lo que los datos que se le aprovisionen al clasificador de MOA para su entrenamiento, indistintamente de la implementación (algoritmo) seleccionada, deberá ser en forma de flujos de datos. Así, la clase *DataSetNodeTraining* permite traducir conjuntos de entrenamientos finitos (datasets) al contexto de flujos de datos, en forma transparente y abriendo la posibilidad, de que el origen de los datos de entrenamiento, no sólo sean flujos de datos en sí mismos, sino que

también sea posible brindar otro conjunto de datos, indistintamente de cual fuere su medio de almacenamiento. Por ejemplo, si se contase con información almacenada del paciente "Juan Perez", con respecto a las métricas que ahora se desean monitorizar (valor de la frecuencia cardíaca, valor de la temperatura axilar, etc.), es posible brindar dicha información al clasificador en la etapa de entrenamiento, para que al momento de comenzar la supervisión del paciente, el clasificador ya cuente con el conocimiento de la entidad específica bajo análisis (paciente trasplantado) particularizado al grupo de seguimiento del mismo ("Juan Perez").

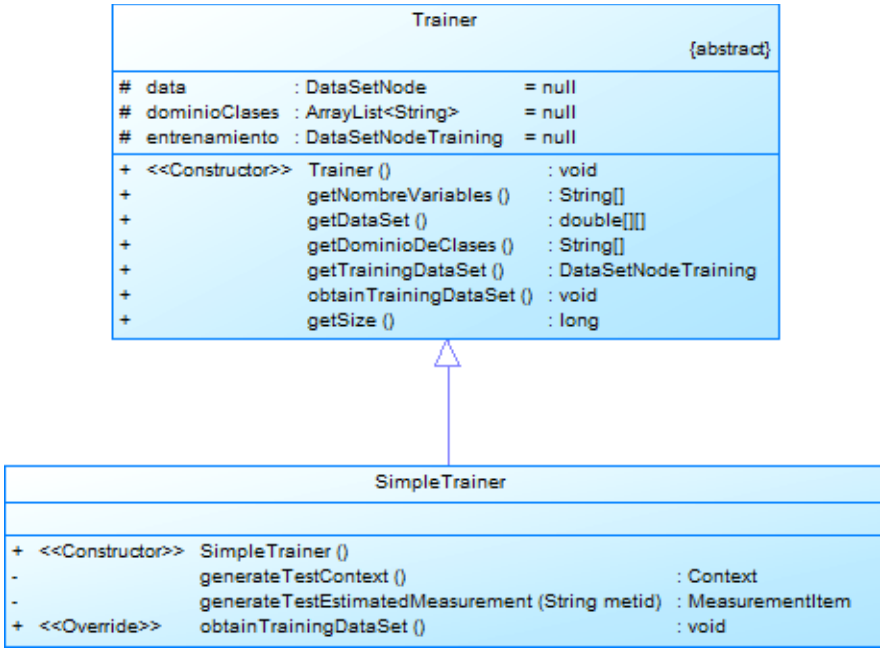


Figura 67. Clase Abstracta Trainer

Otro modo alternativo de entrenamiento, es mediante la derivación de la clase abstracta *Trainer* (ver Figura 67). La clase abstracta, define una serie de métodos fundamentales como la definición de los nombres de las variables, la obtención del conjunto de datos, la obtención del dominio de clases sobre las que se desea decidir, entre otros aspectos, abstrayéndose de donde residen los datos. El cómo obtiene los datos y la satisfacción de los métodos abstractos indicados en la clase base *Trainer* es responsabilidad de las clases derivadas. Esto permite, independizar al EIPFDcMM del lugar de residencia de los repositorios de entrenamiento, cómo se estructuran y almacenan los mismos. A modo de ejemplo, la clase *SimpleTrainer* deriva la clase *Trainer* y genera un conjunto de datos de entrenamiento aleatorio, que sirve como demostración para extensiones posteriores asociadas a la implementación del proceso de obtención de datos de entrenamiento.

8.6 Parametría, Dominio de Clases y Acciones

Dentro de la parametría asociada a grupos de seguimiento en la base de datos C-INCAMI (tabla *tracegroup),* se incorpora un atributo denominado *classToObtainingDataSet.* Tal atributo, permite indicar en forma cualificada, cuál es la clase que deriva de *Trainer,* que será responsable de obtener el conjunto de entrenamiento para el clasificador asociado a un grupo de seguimiento determinado. De este modo, puede deducirse que al igual que ocurriese con el clasificador, el

grupo de entrenamiento (por ej.: el paciente trasplantado) puede ser particularizado a cada grupo de seguimiento y es definido, por los expertos responsables de la definición del proyecto de M&E. Esto es muy interesante, por cuanto permite por ejemplo, a partir de la historia clínica del paciente "Juan Perez", definir qué datos, servirán de base para entrenar a su clasificador, ajustándolo a su historia clínica y dejándolo listo para retroalimentarse, a partir del comportamiento ambulatorio del mismo.

De este modo, el EIPDFcMM además de dar libertad sobre cómo estructurar y obtener el conjunto de entrenamiento, permite que la referenciación a la clase responsable de obtenerlo, sea dinámica. Así, simplemente cambiando el texto con el nombre de la clase cualificada en el atributo *classToObtainingDataSet* en la tabla tracegroup de la base de datos C-INCAMI, se actualizaría la lógica de entrenamiento de su clasificador asociado, sin necesidad de tener que recodificar clase alguna, haciéndolo extensible.

Adicionalmente, la parametría de la base de datos C-INCAMI, permite definir para un grupo de seguimiento, el dominio de clases sobre el cual deberá decidir, su umbral mínimo de probabilidad y la acción a tomar para cada una de ellas. De este modo, la tabla *tracegroup_classaction* dentro de la BD C-INCAMI, permite especificar la información mencionada resaltando dos aspectos fundamentales:

- **Probabilidad mínima**: Si bien un clasificador puede arrojar una decisión asociada a una clase por ej.: RIESGO, la misma siempre tiene una probabilidad asociada. Con este parámetro, se informa a *DecisionMaker* que sólo ejecute la acción asociada a la clase, en la medida que la probabilidad asociada con la decisión informada por el clasificador sea igual o superior al umbral mínimo establecido.

- **Acción**: Las acciones factibles de configuración para cada clase se acotan a una de las siguientes en EIPFDcMM:
    - **NOACTION:** No hacer nada. Esto es coherente cuando la etiqueta de clase es por ejemplo NORMAL, situación en la cual no existiría motivo para notificar en principio ninguna anomalía.
    - **ONLYREGISTER:** Sólo registrar en forma persistente dentro del registro de eventos del EIPFDcMM de la base de datos C-INCAMI. En caso de que el registro de eventos se llene, se reinicia la escritura del mismo pisando los registros más antiguos.
    - **ONLYNOTIFY:** Sólo notificar a los responsables definidos durante la configuración del proyecto de M&E.
    - **REGISTER_AND_NOTIFY:** Registrará y notificará la ocurrencia de la decisión.

Las acciones son llevadas adelante efectivamente por una instancia de *DecisionMaker*. Al momento de inicializarse la instancia, se cargará para cada grupo de seguimiento, su dominio de clases, umbrales y acciones predefinidos los que serán oportunamente comunicados al clasificador del grupo de seguimiento mediante su constructor.

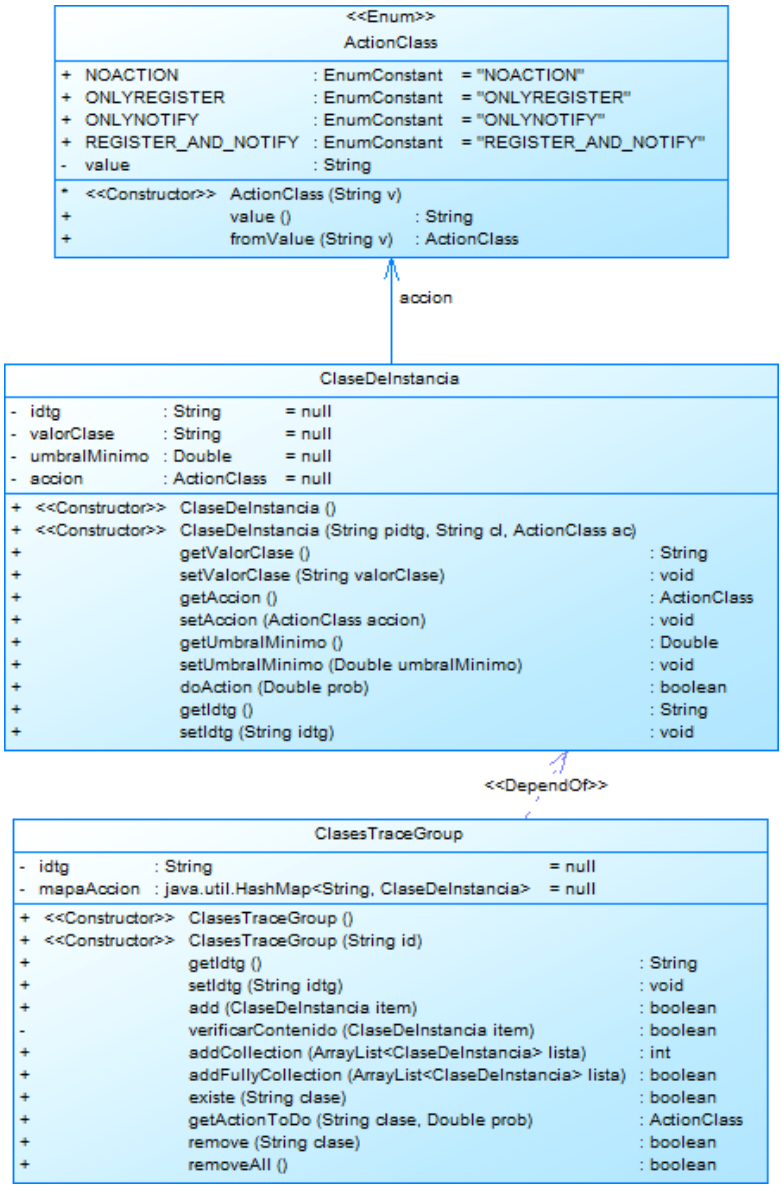


Figura 68. Clase ClasesTraceGroup y sus Principales Relaciones

La forma en que la instancia de *DecisionMaker* gestiona las acciones y umbrales de probabilidad por cada valor de su dominio de clase es sintetizada en la Figura 68. Cada grupo de seguimiento (por ej.: "Juan Perez"), contiene la información de las clases de decisión mediante una instancia de *ClasesTraceGroup,* la cual contiene un conjunto finito de objetos *ClaseDeInstancia* (por ejemplo, los criterios de decisión definidos para un indicador global, similar al realizado en la Tabla 1 para el indicador elemental *Nivel de la Temperatura Corporal*). Esta última clase, es la responsable de identificar cada etiqueta de clase, asociar su umbral mínimo de probabilidad y su acción pertinente. La acción es modelada como una enumeración denominada *ActionClass*, la cual responde a los tipos de acciones posibles que puede asumir el atributo *accion* de la tabla *tracegroup_classaction* en la base de datos C-INCAMI (lugar donde se define y configura el proyecto de M&E).

8.7 Tomador de Decisiones y Clasificadores

La clase *DecisionMaker*, ha incorporado dentro de sus responsabilidades, la obtención del dominio de clases, umbrales mínimos de probabilidad, acciones y su asociación con un grupo de seguimiento dado. Esto último, lo realiza mediante un mapeo por dispersión del ID del grupo de seguimiento, teniendo asociado objetos del tipo *ClasesTraceGroup*. Así, además de conocer el tipo de acción a llevar adelante ante una decisión dada (en la medida que satisfaga la probabilidad mínima indicada), es capaz de informar dentro del proceso de entrenamiento de un clasificador, el dominio de las clases definidas. De este modo, por ejemplo, es posible definir un indicador global específico al paciente "Juan Perez", con sus propios criterios de decisión y probabilidades asociadas.

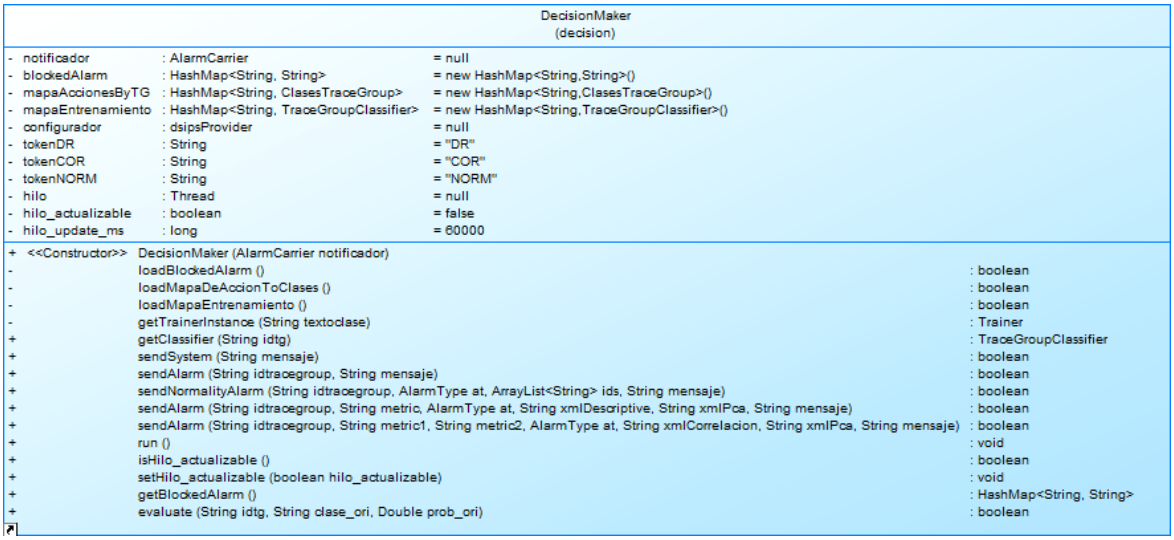


Figura 69. Clase DecisionMaker Adecuada para la Gestión de Dominios de Clases por Grupo de Seguimiento

La clase *DecisionMaker* (ver Figura 69), ha incorporado la responsabilidad de obtener el conjunto de entrenamiento para cada clasificador de un grupo de seguimiento, mediante el método *loadMapaEntrenamiento().* Adicionalmente, la clase *DecisionMaker*, incorpora tres métodos centrales:

- *getTrainerInstance:* A partir del texto con el nombre de la clase totalmente cualificado (derivada de *Trainer*), retorna un objeto que representa el conjunto de entrenamiento. El mismo, será utilizado mediante polimorfismo por el clasificador, para poder llevar adelante el entrenamiento supervisado y así culminar su inicialización.
- *getClassifier:* En caso de estar definido, a partir de un grupo de seguimiento dado, se retornará su clasificador entrenado para su aplicación.
- *evaluate:* A partir del grupo de seguimiento del que se trate, la clase arrojada por el clasificador y su probabilidad asociada, éste método determina (en base a la información del mapa de entrenamiento) la acción a seguir; por ejemplo: Notificar, Registrar, etc.

De este modo, luego de que la función de suavización culmina el análisis estadístico coordinado a través de la clase *StatisticalAnalizer*, informa el objeto DataSetNode (instantánea del buffer multinivel –ver sub-sección 7.8-), al clasificador que la instancia de *DecisionMaker* retorne. Dicho clasificador, irá informando en línea a la instancia de *DecisionMaker*, sobre cada decisión a la que arribe ante cada unidad lógica de medición evaluada, para que mediante el método *evaluate,* decida el curso de acción a tomar.

9 Caso de Aplicación: Implementación del EIPFDcMM

En el capítulo 8, se ha discutido el contexto de procesamiento y cómo éste afecta a la clasificación de flujos de datos. A seguir, se discutió el análisis masivo en línea (MOA) y su relación con el procesador de toma de decisiones. Luego, se fundamentó la elección del algoritmo de clasificación sobre flujos de datos que utiliza el EIPFDcMM. A continuación, se analizó la estrategia mediante la cual, el EIPFDcMM, aplica los clasificadores en línea, a los grupos de seguimiento como herramienta de soporte de decisión. Luego, se discutió la fase de entrenamiento de los clasificadores y entonces, se analizó el efecto de los parámetros del proyecto de M&E en la base de datos C-INCAMI, en el entrenamiento como al momento de tomar una decisión. Finalmente, se analizó cómo el tomador de decisiones adecuaba su comportamiento, a partir del análisis de las decisiones provenientes desde los clasificadores.

En este capítulo, se discuten los aspectos de implementación del EIPFDcMM, y cómo estos influyen al momento de la simulación del escenario asociado al caso de aplicación presentado en la sub-sección 3.5.

Luego, se discute desde el punto de vista experimental y a los efectos de validar inicialmente nuestra estrategia de procesamiento, cómo se planificó, diseñó y ejecutó la simulación sobre un escenario asociado al caso de aplicación, y cuáles fueron los objetivos que se perseguían a través de la misma.

Finalmente, se analizan los resultados surgidos de la simulación, para poder concluir sobre los objetivos planteados en la misma.

9.1 Aspectos de Implementación

Se desarrolló un prototipo en JAVA, empleando el IDE NetBeans 6.9, que permitió implementar las funcionalidades del EIPFDcMM. Dicho prototipo, permite a aquellas fuentes de datos que deseen integrarse con el EIPFDcMM, implementar las interfaces y hacer uso de sus funcionalidades a partir del paquete denominado *dsips-ma.jar.* Dicho paquete, facilita la incorporación de la lógica de recolección y adaptación asociada a la función de reunión en otros software, independientemente de si estos fuesen

o no móviles. El paquete *dsips-ma.jar*, contiene la funcionalidad que inicia con la interface *DataSource* (ver sub-sección 5.1) para con la fuente de datos, hasta aquella asociada con el adaptador de mediciones (clase *MeasurementAdapter,* ver sub-sección 6.4), la cual permitirá configurar y transmitir las mediciones desde diferentes fuentes de datos a la recolección central de mediciones. La recolección central de mediciones, si bien parte de la función de reunión, no está incorporado dentro del paquete *dsips-ma.jar*, sino que es parte del paquete *dsips.jar*.

El paquete *dsips.jar,* tiene implementada la lógica funcional desde la recolección central de mediciones, hasta los clasificadores, tomadores de decisión y notificadores del proceso de toma de decisión. Dicho paquete, permite embeber el procesamiento central, suavización y toma de decisiones en cualquier software que lo requiera. Para el caso de nuestra simulación, se ha implementado la recolección mediante servicios web, por lo cual se desarrolló una aplicación web denominada *dsipsWEB.war*, la cual hace uso del paquete *dsips.jar* para incorporar la lógica funcional y hacer públicos los servicios de configuración (ver sub-sección 5.2), los servicios de transmisión (ver sub-sección 6.2), como así también efectivizar la implementación del buffer multinivel (ver sub-sección 6.5), el proceso de análisis estadístico o suavización (ver capítulo 7) y el proceso de toma de decisión (ver capítulo 8).

La simulación se desarrolló sobre un sistema de memoria compartida con procesador Athlon Dual Core de 64 bits, con plataforma Windows Home Premium. Para ejecutar la aplicación web (*dsipsWEB.war*), se utilizó el servidor de aplicaciones Apache Tomcat 6.0.20 con Java Runtime Environment (JRE) 1.6.0_20, comunicado mediante multihilo, con el software estadístico R mediante el paquete Rserve. Los clientes se implementaron como aplicaciones desktop, las cuales a través del paquete *dsips.jar*, se configuraban y transmitían a los servicios web desplegados en Tomcat (Diván, 2011; Diván, Olsina, et al., 2011a).

9.1.1 Paralelismo en EIPFDcMM

Como se ha visto anteriormente y en términos generales, la lógica funcional del EIPFDcMM se puede separar en dos áreas: la primera está asociada a la entidad bajo análisis, desde la cual debemos recuperar las medidas y transmitirlas; y la segunda, se asocia a la recepción y procesamiento de las mediciones correspondientes a la entidad bajo análisis.

La primera área funcional, asociada con la entidad bajo análisis, comúnmente es desplegada sobre dispositivos móviles los cuales se conectan directamente con sensores (termómetros, marcapasos, etc.) para recolectar las medidas. Sobre dichos dispositivos, se ejecuta la pieza de software que implementa la funcionalidad del adaptador de mediciones (-MA- ver Figura 12) y por ende, recepta/organiza localmente, previo a la transmisión (mediante servicios web) a la función de reunión, las mediciones desde los sensores. El MA, previo a cualquier transmisión y solo durante su inicialización, debe registrar ante los servicios web de configuración del EIPFDcMM, qué métricas implementará cada sensor al que se asocia y a qué entidad bajo análisis medirá. Esto último, puede verse en la Figura 70 asociada al caso de aplicación, donde una cantidad de sensores de marcas/modelos determinados se asocian al paciente 1, y éste al

dispositivo Palm Treo 750. Una vez configurado, el MA se encuentra listo para informar las mediciones a la función de reunión.

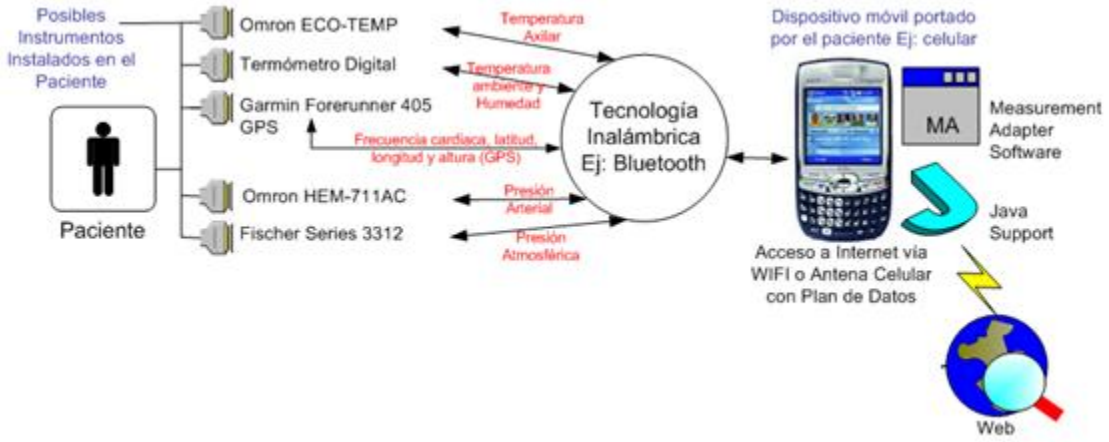


Figura 70. Dispositivos Asociados a Pacientes Ambulatorios Trasplantados en el Esquema de Aplicación del Enfoque Integrado de Procesamiento de Flujos de Datos Centrado en Metadatos de Mediciones

La segunda área funcional, se asocia a la función de reunión, la cual debe receptar los flujos de datos, reunirlos y remitirlos a un buffer central. Dado que los MA se encuentran distribuidos geográficamente recolectando mediciones desde sus entidades bajo análisis asociadas, la forma de receptar los flujos ha sido implementada mediante servicios web desplegados sobre un servidor de aplicaciones Apache Tomcat 6.0.20 con balance de carga. Esto permite, receptar los flujos de mediciones en paralelo desde los adaptadores de mediciones, y dar instrucción de cómo procesar los mismos ante el buffer en función de sus metadatos. La clase responsable de dar instrucción de procesamiento, en base a los metadatos, se denomina *DsipsProviderCommunication*. Dicha clase, presentada en la sub-sección 6.7, corresponde a una clase activa invocada por los servicios web, que permite el acceso al buffer central a través de su gestor (cuya implementación corresponde a las clases *BufferTG* y *BufferMetric* analizadas en la sub-sección 6.5), y ante cada invocación de los servicios web por parte de un MA.

No obstante, el hecho de permitir el paralelismo en la recepción de mediciones mediante servicios web, introduce el siguiente interrogante ¿Se podrá regular de algún modo que ante incrementos abruptos en las tasas de arribo, la tasa de procesamiento no se vea desbordada? La respuesta en EIPFDcMM fue abordada mediante dos mecanismos complementarios:

a) **El buffer local de cada métrica incorpora un comportamiento de cola sincronizada en cada grupo de seguimiento** (ver sub-sección 6.5). Dado que tanto para las actividades de minería de datos como para las técnicas predictivas, el mayor peso relativo se asocia con la historia reciente, cada métrica en el buffer (la que luego se convertirá en variable bajo estudio en los análisis estadísticos posteriores y/o técnicas de minería de datos), incorpora una cola sincronizada con comportamiento dinámico y límite máximo de crecimiento. Esto implica que la cola tiene un tamaño finito pre-configurado (el cual puede ser dinámicamente modificado si se deseara), el cual al llenarse, comienza a descartar en

forma sincronizada tantos datos antiguos como nuevos deseen incorporarse. De hecho, suponiendo que por algún motivo se disminuya dinámicamente el tamaño de cola para la métrica, la cantidad de datos que excedan al nuevo tamaño, serán descartados a partir de los datos más antiguos.

b) **Load Shedding** (ver sub-sección 6.6). La idea de la cola sincronizada supone que el gestor de buffer a nivel de grupo de seguimiento no ha sido desbordado, por lo que al incorporar la solución del inciso 'a' se incorpora también un cuello de botella en el gestor de buffer a nivel de grupo de seguimiento (ver sub-sección 6.5). Para abordar esta problemática, se incorpora la técnica de load shedding con descarte selectivo a nivel del buffer de grupo de seguimiento. Dado que toda medición, previo a derivarse al buffer a nivel de métrica, debe primero atravesar el buffer a nivel de grupo de seguimiento, la técnica de load sheeding basada en metadatos C-INCAMI, permite priorizar las métricas a preservar en caso de desborde y a través de los metadatos en el flujo (ver sub-sección 6.1), determinando si la medición es o no prioritaria, a los efectos de su descarte o permitir su avance al siguiente nivel del buffer. De este modo, se logra regular la tasa de servicio ante incrementos abruptos en la tasa de arribo a nivel de buffer de grupo de seguimiento.

La siguiente cuestión es, si la clase *DsipsProviderCommunication* es quien informa al buffer a nivel grupo de seguimiento las mediciones y éste último mediante load shedding regula la tasa de arribo versus la tasa de servicio, ¿cómo evitar el desborde a nivel servicios web? Es una duda razonable, por cuanto lo mismo ocurrió en el buffer a nivel de métrica, cuando se incorporaba la regulación mediante cola sincronizada. Esto último, provocó que se trasladase la problemática a nivel del buffer de grupo de seguimiento, y allí se aplicó load shedding para resolverlo, por ende, si creciera la cantidad de MA que desean transmitir simultáneamente ¿cómo se podría resolver? Ello fue lo que motivó a implementar los servicios web con balanceo de carga. De este modo, ante una situación de potencial desborde de todos los nodos Apache Tomcat, bastaría con agregar los nodos dinámicamente para incrementar la capacidad de balance de carga, sin que sufra riesgo la aplicación del EIPFDcMM.

9.1.2 Concurrencia en EIPFDcMM

Una vez recibidos los flujos mediante los servicios web, la responsabilidad de organizar las mediciones es dada al gestor de buffer a nivel de grupo de seguimientos, instancia de la clase *BufferTG* (ver sub-sección 6.5). Dicha instancia, intentará acceder al mismo en forma concurrente ante cada petición de los servicios web y la cuestión es ¿cómo lograr la unificación de mediciones, evitando corrupción de datos y sin caer en la serialización? Es por ello que el buffer fue organizado en varios niveles. Dado que cada MA, invocador por excelencia de los servicios web, transmite flujos de mediciones en nombre de un grupo de seguimiento, cada instanciación de servicio web arrojará mediciones para un determinado grupo. Si en paralelo, otro MA invoca el servicio web para transmitir mediciones, seguramente corresponderán a otro grupo de seguimiento. De este modo, si se considera la organización del buffer expuesto en la sub-sección 6.5, no existirá conflicto en las mediciones, por cuanto la clase *BufferTG* implementa una tabla de dispersión para

localizar en tiempo constante el gestor de métricas para el grupo de seguimiento. Localizado éste, informa dicho gestor al servicio web para que prosiga con el aprovisionamiento de mediciones. De este modo, la única serialización posible a nivel buffer de grupo de seguimiento, es el asociado a la localización de la instancia de *BufferMetric* dentro de la tabla de dispersión, lo cual es un tiempo constante y reduce notablemente el riesgo de serialización.

Como se ha mencionado anteriormente, dado que la cola a nivel buffer de métrica se encuentra sincronizada, la única forma posible de serialización radicaría en que el mismo MA invoque por segunda vez los servicios web (a los efectos de transmisión de mediciones), cuando aún el buffer a nivel de métrica de su grupo de seguimiento no haya culminado de procesar la primera invocación. De ser esta la situación, se ponen en funcionamiento los mecanismos de load shedding bajo la política configurada. Pero supongamos además, que la métrica es prioritaria y por ende atraviesa el mecanismo de load shedding inundando el buffer a nivel de métrica ¿se caería en una serialización? No, porque como se mencionó anteriormente, lo que prevalece en EIPFDcMM es la historia reciente, por lo que ante una situación de este tipo, la sincronización de la cola buscaría evitar corrupción del dato (por ej.: que la misma métrica para igual grupo de seguimiento posea dos medidas deterministas diferentes en igual instante de tiempo), y aquí en cambio, se descartaría automáticamente de la cola (sin mediar análisis) un volumen de datos antiguos igual al que desea ingresar, efectuando directamente su reemplazo en igual espacio de memoria.

9.2 Simulación de un Escenario Asociado al Caso de Aplicación

El objetivo de la simulación de un escenario asociado al caso de aplicación, es analizar el comportamiento del EIPFDcMM, en cuanto al tiempo requerido para procesar, analizar estadísticamente la ventana de datos, suavizarla y disparar las alarmas pertinentes si fuera el caso, a medida que varía la cantidad de variables y la cantidad de mediciones (asociadas a cada una de las variables) en las ventanas de procesamiento (Diván, Olsina, et al., 2012). En adelante, se empleará el término variable o métrica en forma indistinta, dado que a los efectos del análisis estadístico, la métrica se constituye en variable bajo estudio, independientemente se asocie ésta a un atributo de entidad o propiedad de contexto.

A los efectos de la regulación de la simulación, se definen las siguientes variables:

- ***Cantidad de variables (qVar)*** intervinientes en una ventana de transmisión/procesamiento
- ***Cantidad de mediciones (meds)*** informadas dentro de la ventana de transmisión/procesamiento asociada a cada variable

Las variables en *qvar*, son representadas por las métricas directas que definen qué medir. En ésta última, se tienen en cuenta tanto las métricas vinculadas a atributos de la entidad como aquellas vinculadas a propiedades del contexto de la entidad. De este modo, la cantidad de variables incluye tanto a métricas asociadas a los atributos de la entidad, como así también a las propiedades contextuales.

La simulación variará desde un mínimo de 3 variables hasta 99, como así también variará desde 100 mediciones hasta 1000, asociado a cada una de ellas. Se parte de 3 variables como mínimo, para que tenga sentido realizar el análisis de correlación y el de componentes principales dado que el objeto final es medir tiempos. Si en la práctica, la cantidad de variables fuese inferior a 3, desde el punto de vista del procesamiento sería beneficioso ya que el volumen de datos se ve reducido. Se fija un máximo de 99 variables, ya que si bien en la vida real pueden existir proyectos con mayor cantidad, esta magnitud dado el rango que define, nos permitiría aproximar una tendencia en cuanto al consumo marginal de tiempo. En cuanto a las mediciones, se parte aleatoriamente de 100 mediciones para simular un tamaño mínimo de ventana, si fuese menor el tamaño, se reitera, sería beneficioso al procesamiento, ya que el objeto en este sentido es medir tiempo insumido por el análisis estadístico. Como cota máxima, se toman 1000 mediciones para cada variable involucrada. El número es escogido racionalmente desde el punto de vista de los data streams, ya que el flujo de datos se caracteriza por transmitir flujos tan pronto como sea posible y si fuese el caso ideal en forma continua sin interrupciones, el tamaño de la ventana tendería a disminuir hasta 1 medición por variable. No obstante, se planteó la variación hasta 1000 mediciones previendo la posible incorporación de transmisiones asíncronas a ráfagas.

Téngase en cuenta que el caso máximo analizado considera 99 variables con 1000 mediciones cada una, lo que representa el análisis estadístico de 99000 mediciones en forma conjunta, para una ventana dada (1 único envío desde la instancia de *MeasurementAdapter*). La idea de establecer el máximo de mediciones por variable en 1000, es a los efectos de analizar la variabilidad del consumo de tiempo marginal a medida que se incrementa el requerimiento de procesamiento, por lo que nos permitirá analizar una tendencia en este sentido, ante el supuesto caso de que se desee a posteriori analizar 1500, 2000 o más mediciones por variables.

Téngase en cuenta que cada medición será analizada como un tipo *Double* de JAVA, por lo que el requerimiento mínimo de memoria, en caso de desear reproducir esta prueba en otra arquitectura, será:

$$\text{Memoria}_{\text{mínima}} = (3*(\textit{tamaño del Double en la Arquitectura})*100) + \textit{Memoria}_{\textit{programa}} + \textit{Memoria}_{R} + \textit{Memoria}_{\textit{Rserve}}$$

En forma análoga, el tamaño máximo de memoria requerido será:

$$\text{Memoria}_{\text{máxima}} = (99*(\textit{tamaño del Double en la Arquitectura})*1000) + \textit{Memoria}_{\textit{programa}} + \textit{Memoria}_{R} + \textit{Memoria}_{\textit{Rserve}}$$

De momento, la simulación ha planteado dos variables encargadas de regular la misma: *qVar* y *meds*. Dado que el objeto de la simulación es medir tiempos, se incorporan las siguientes variables a la simulación:

- **startup**: Tiempo en milisegundos insumido en la apertura de la conexión a Rserve

- **gendato**: Tiempo en milisegundos insumido en la preparación del conjunto de datos aleatorio
- **andesc**: Tiempo en milisegundos insumido en el Análisis Descriptivo de los datos
- **cor**: Tiempo en milisegundos insumido en el Análisis de Correspondencias
- **pca**: Tiempo en milisegundos insumido en el Análisis de Componentes principales
- **total**: Tiempo en milisegundos insumido en el proceso total

Así, la simulación tomará los tiempos en milisegundos asociado a cada una de las variables recientemente indicadas, para cada variación de las variables *qVar* y *meds*. A partir de los datos obtenidos, es que se realiza el análisis descriptivo, de correlación y de componentes principales en la siguiente sub-sección.

9.3 Análisis de Datos

A partir de los datos obtenidos desde la simulación, la Tabla 13 muestra el resumen descriptivo de las variables asociadas con los análisis estadísticos, inicialización y tiempo total.

	Startup	gendato	andesc	cor	pca	total
Mínimo	0.00000	0.00000	15.00000	0.00000	0.00000	15.00000
Primer Cuartil	0.00000	0.00000	94.00000	15.00000	16.00000	125.00000
Mediana	0.00000	0.00000	203.00000	16.00000	31.00000	250.00000
Media	0.03561	0.23640	268.60000	24.57000	50.40000	343.00000
Tercer Cuartil	0.00000	0.00000	453.00000	31.00000	63.00000	546.00000
Máximo	47.00000	16.00000	936.00000	110.00000	234.00000	1092.00000
Desv.Standard	1.29363	1.97301	204.16967	23.56094	46.33493	265.65655

Tabla 13. Resumen Descriptivo. Valores expresados en milisegundos (ms)

Las variables *startup* y *gendato,* prácticamente no inciden en el tiempo total. Esto último, puede deducirse a partir del tiempo máximo insumido por cada una de ellas y si adicionalmente, se compara sus desviaciones estándar contra cualquiera de las variables involucradas en el análisis estadístico propiamente dicho, la relación es de al menos 1 a 11,94 (que resulta de 23,56094/1,97301).

De la Tabla 13**,** surge que la variable *andesc* tiene una alta incidencia en el tiempo total y le seguiría en orden descendente las variables *pca* y *cor*. De igual modo, la variable *andesc* es la que mayor desvío estándar presenta (204,16967), seguido por *pca* (46,33493) y *cor* (23,56094), lo que considerando las medianas, medias, máximos y mínimos estaría indicando un desplazamiento de la media hacia valores superiores.

El boxplot de la Figura 71, permite identificar visualmente cómo los máximos de cada variable empujan la media hacia valores superiores, principalmente en las variables *andesc* y *total*. Adicionalmente, puede observarse que las variables *pca* y *cor* presentan outliers superiores (representados con círculos en el boxplot), lo cual indicaría posibles demoras adicionales a la normal, que podrían asociarse con el crecimiento de la ventana de procesamiento.

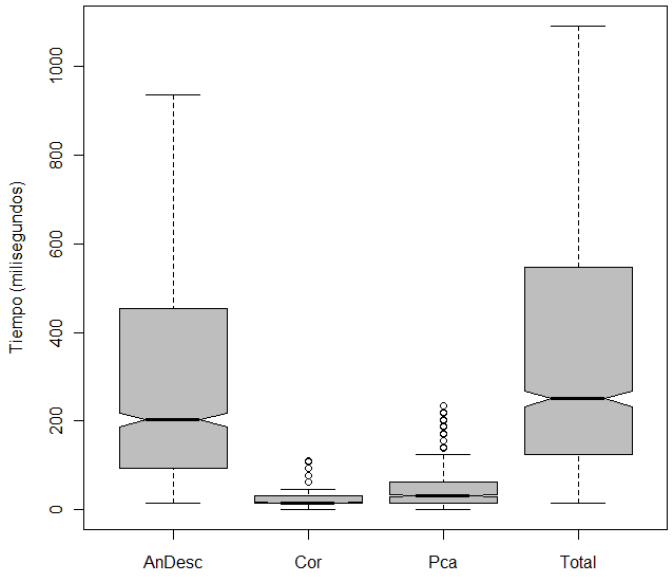


Figura 71. Boxplot de las variables andesc, cor, pca y total

La Figura 72 permite visualizar el comportamiento conjunto, de la evolución en la cantidad de mediciones, la cantidad de variables y el tiempo total del proceso en milisegundos. Desde la vista frontal tiempo-mediciones, puede observarse que a medida que se incrementa la cantidad de mediciones, la evolución del tiempo total del proceso no es significativa. Por el contrario, desde la perspectiva variables mediciones (ver Figura 73), el incremento en la cantidad de variables produce incrementos temporales visualmente superiores a los primeros.

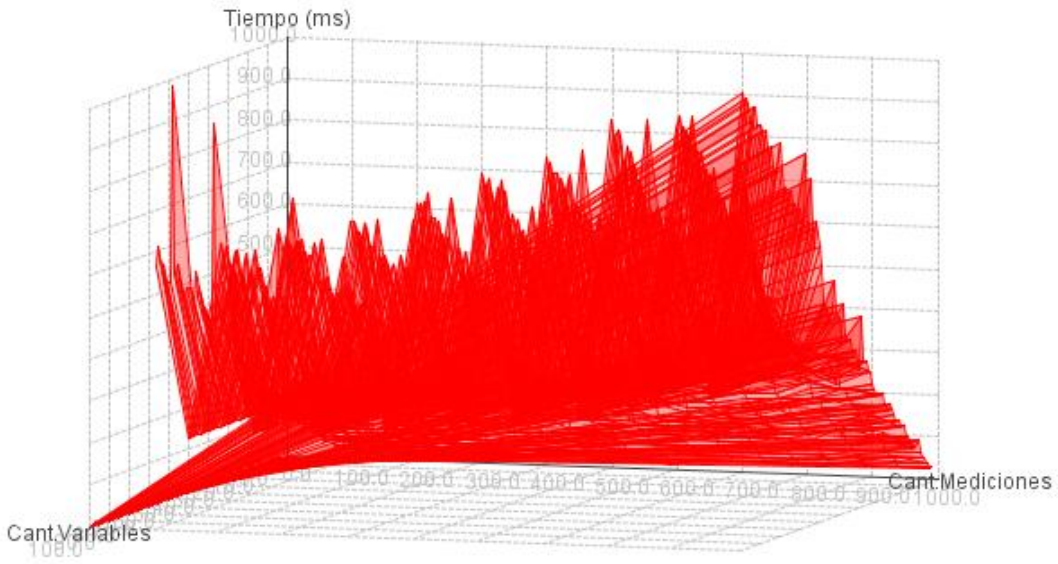


Figura 72. Evolución del tiempo, cantidad de variables y cantidad de mediciones. Vista Frontal Tiempo-Mediciones.

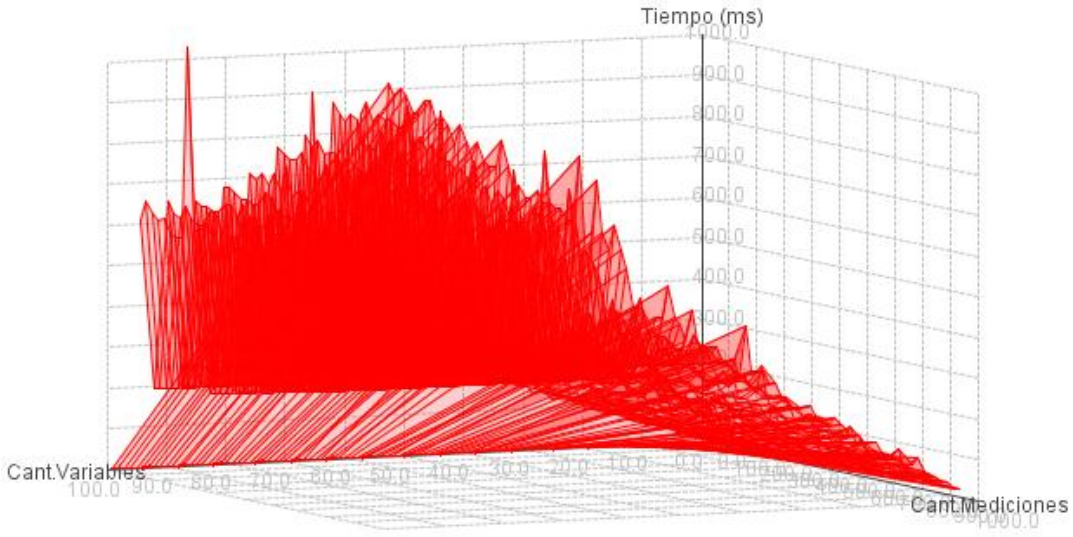


Figura 73. Evolución del tiempo, cantidad de variables y cantidad de mediciones. Vista Frontal Variables-Mediciones.

Los incrementos temporales que produce el incremento en la cantidad de variables es visualmente superior al que produce el incremento en la cantidad de mediciones (ver Figura 72 y Figura 73). Adicionalmente, se puede observar desde una perspectiva superior de la gráfica (vista superior de los ejes variables – mediciones, en la Figura 74), analizándola desde la perspectiva temporal, que a medida que la cantidad de variables se incrementan comienzan a aparecer picos de procesamiento.

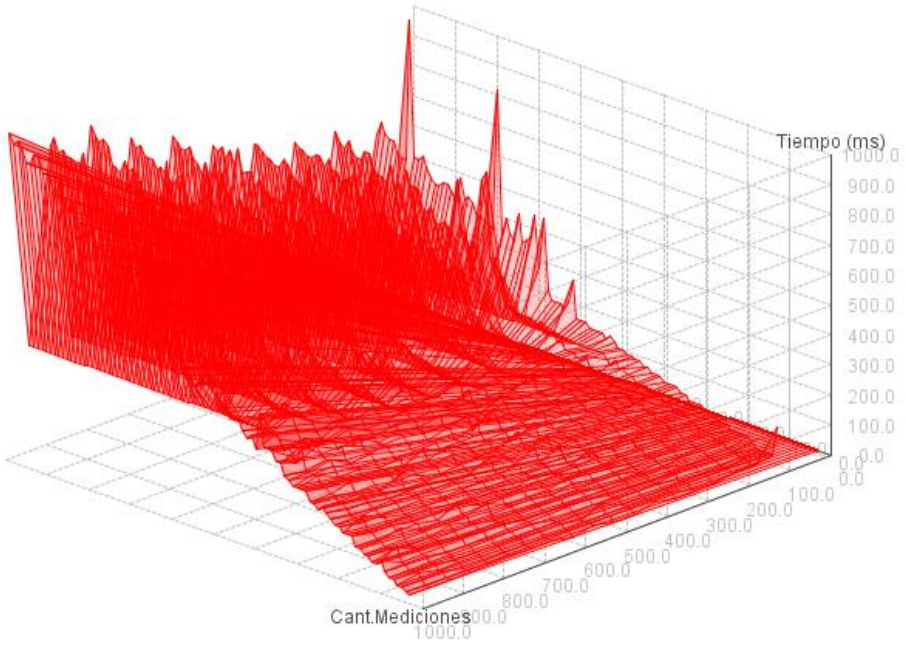


Figura 74. Evolución del tiempo, cantidad de variables y cantidad de mediciones. Vista Superior Mediciones-Tiempo.

En este último sentido, puede diferenciarse claramente dos zonas visuales, una que podría denominarse “serena“, donde el incremento en el número de variables y observaciones no afectan

el normal crecimiento al tiempo global, y otra que podríamos denominar “picada” (por su forma), donde el incremento de las variable incorporan cierto grado de volatilidad, dado por la aparición de outliers en el tiempo global de procesamiento.

9.3.1 Variabilidad del Sistema

Esta sección, tiene por finalidad, analizar los factores que incorporan mayor variabilidad en el sistema, en función del análisis descriptivo de datos discutido anteriormente. A los efectos de analizar las variables que más aportan a la variabilidad global del sistema, se aplica el análisis de componentes principales (PCA), mediante la utilización de la matriz de correlación, exponiéndose en la Figura 75, la variabilidad explicada para cada componente principal. Como se puede apreciar en la Figura 71, existen variables que presentan outliers. Esto último, implica que si se deseara efectuar el análisis de componentes principales, empleando matriz de covarianza en lugar de matriz de correlación, se estaría incorporando un problema por cuanto el outlier afecta, al igual que a la media aritmética, al cálculo de la matriz de covarianza, arrastrando sus valores hacia la zona de incidencia del outlier (sea superior o inferior).

Componente	sd	var	% Explicada	Acumulada
1	2,1546760750	4,6426289882	58,03%	58,03%
2	1,0553707930	1,1138075107	13,92%	71,96%
3	0,9974267030	0,9948600279	12,44%	84,39%
4	0,9893999600	0,9789122808	12,24%	96,63%
5	0,3768626410	0,1420254502	1,78%	98,40%
6	0,3235947290	0,1047135486	1,31%	99,71%
7	0,1518239450	0,0230505103	0,29%	100,00%
8	0,0012977970	0,0000016843	0,00%	100,00%
		8,0000000010		

Figura 75. Variabilidad Explicada de los Componentes Principales

Notar que el primer componente principal (PC), explica por sí solo el 58% de la variabilidad del sistema. De este modo, es sumamente interesante analizar la composición de los aportes del primer componente principal, el cual se muestra en la Tabla 14 como salida del software R.

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
qVar	**-0,428**	0,302	0,374	-0,202	0,731	0,000	0,000	0,000
meds	-0,144	**-0,868**	-0,165	0,113	0,384	-0,188	0,000	0,000
stratup	0,213	**-0,970**	0,110	0,000	0,000	0,000	0,000	0,000
gendato	0,104	0,132	0,983	0,000	0,000	0,000	0,000	0,000
andesc	**-0,449**	0,189	0,314	-0,537	-0,602	0,000	0,000	0,000
cor	**-0,434**	-0,141	-0,738	-0,487	0,000	0,000	0,000	0,000
pca	**-0,435**	-0,182	-0,223	0,822	0,167	-0,137	0,000	0,000
total	**-0,460**	0,103	0,137	-0,379	0,783	0,000	0,000	0,000

Tabla 14. Salida del software R para el Análisis de Componentes Principales

El análisis de componentes principales, confirma que el tiempo asociado a las variables *startup* y *gendato*, prácticamente no inciden en el componente principal (PC) 1. No obstante, ello

no significa que se debe descartar en términos de variabilidad, dado que representa la variable de mayor aporte en PC 3, donde explica el 12,44% de la variabilidad del sistema.

La PC 1 confirma lo esquematizado visualmente en el análisis descriptivo (ver Figura 72 y Figura 73). Esto es, que el incremento de las mediciones no aporta inicialmente demasiado al tiempo total de procesamiento, mientras que sí lo hace de un modo relativamente homogéneo *qvar, andesc, total, pca y cor.* En forma análoga a lo dicho en el párrafo anterior, no debe descartarse a *meds* solo por que aporta poco a la variabilidad en PC1, ya que es la variable que más aporta en PC2, donde explica el 13,92% de la variabilidad del sistema.

De este modo, PCA nos permite determinar tres ejes claros de variabilidad: 1) PC 1, el cual se nutre principalmente de las variables *qvar, andesc, total, pca y cor* y explica el 58,03% 2) PC 2, el cual se nutre principalmente de *meds* y explica el 13,92% y finalmente, 3) PC 3, el cual se nutre principalmente de *startup* y explica el 12,44%. De este modo, los tres primeros componentes principales explican el 84,39% de la variabilidad del sistema.

9.3.2 Análisis de Correlación

Dado que se ha detectado la presencia de outliers en alguna de las variables bajo estudio, es conveniente emplear la matriz de correlación para eventualmente detectar posibles relaciones lineales entre las variables.

```
> cor(matriz)
                qVar          meds       Startup       GenDato       AnDesc
qVar     1.000000e+00  8.942024e-05 -0.039881269  0.124507229  0.97169273
meds     8.942024e-05  1.000000e+00 -0.044269866  0.034513644  0.12442329
Startup -3.988127e-02 -4.426987e-02  1.000000000 -0.003413529 -0.02989642
GenDato  1.245072e-01  3.451364e-02 -0.003413529  1.000000000  0.13021685
AnDesc   9.716927e-01  1.244233e-01 -0.029896421  0.130216851  1.00000000
Cor      7.853603e-01  3.998799e-01 -0.028730492  0.100511248  0.84501302
Pca      7.735025e-01  4.491646e-01 -0.011535517  0.106564187  0.84588770
Total    9.520086e-01  2.094318e-01 -0.022693096  0.134735993  0.99177132
               Cor         Pca       Total
qVar     0.78536035  0.77350250  0.95200862
meds     0.39987990  0.44916456  0.20943178
Startup -0.02873049 -0.01153552 -0.02269310
GenDato  0.10051125  0.10656419  0.13473599
AnDesc   0.84501302  0.84588770  0.99177132
Cor      1.00000000  0.88982053  0.89385578
Pca      0.88982053  1.00000000  0.90409854
Total    0.89385578  0.90409854  1.00000000
```

Figura 76. Salida del Análisis de Correlación del software R

En principio y por lo expuesto en la Figura 76, existiría una fuerte correlación entre *qvar* (cantidad de variables) para con las variables *andesc, cor, pca y total,* pero no ocurre lo mismo con la cantidad de mediciones (*meds*). Esto último, ratifica analíticamente lo que se exponía gráficamente con respecto a los incrementos en el tiempo total en las figuras Figura 72 y Figura 73.

Se puede observar en la Figura 76, correlaciones entre *total* con respecto a *pca, cor y andesc.* Ello es así, dado que total se compone de la sumatoria de los anteriores, y como pudo observarse en el análisis descriptivo, son quienes definen el valor final del tiempo total de procesamiento. De este modo se podría indicar que el PC 1 es en realidad explicado por *qvar* y *total*. De este modo, si se aplicase PCA solo sobre *meds*, *qvar* y *total,* es decir que se extraerían las variables correlacionadas, debiera arrojar un resultado similar al indicado en el PCA original (ver Figura 77 y Figura 75).

Componente	sd	var	% Explicada	Acumulada
1	1,4052728000	1,9747916424	65,83%	65,83%
2	0,9999812000	0,9999624004	33,33%	99,16%
3	0,1588897000	0,0252459368	0,84%	100,00%

Figura 77. Componentes Principales sin variables correlacionadas

Según lo expuesto en la Figura 77, entre la primera y segunda componente se explica prácticamente la variabilidad completa del sistema. En función de esto, es de suma importancia analizar la composición de cada una de las componentes principales del PCA sin variables correlacionadas (ver Tabla 15).

	Comp.1	Comp.2	Comp.3
qVar	**0,691**	0,215	0,691
meds	0,152	**-0,977**	0,152
total	**0,707**	-0,707	0,000

Tabla 15. Salida del software R para PCA. Composición de componentes basados en variables no correlacionadas

Es importante destacar, que quienes vuelven a explicar la variabilidad del sistema son *qvar* y *total* a través de PC 1, siendo la variable *meds* la de mayor aporte en PC 2. De este modo se deduce, que la principal variabilidad del sistema dentro del contexto de las pruebas realizadas va de la mano con la cantidad de variables, y en menor medida con la cantidad de mediciones.

9.3.3 Análisis del Incremento del Tiempo de Procesamiento

De acuerdo al análisis realizado en las sub-secciones 9.3.1 y 9.3.2, se sabe que el incremento en el tiempo de procesamiento, está influido principalmente por la evolución del número de variables más que por el incremento en la cantidad de mediciones en cada ventana de procesamiento.

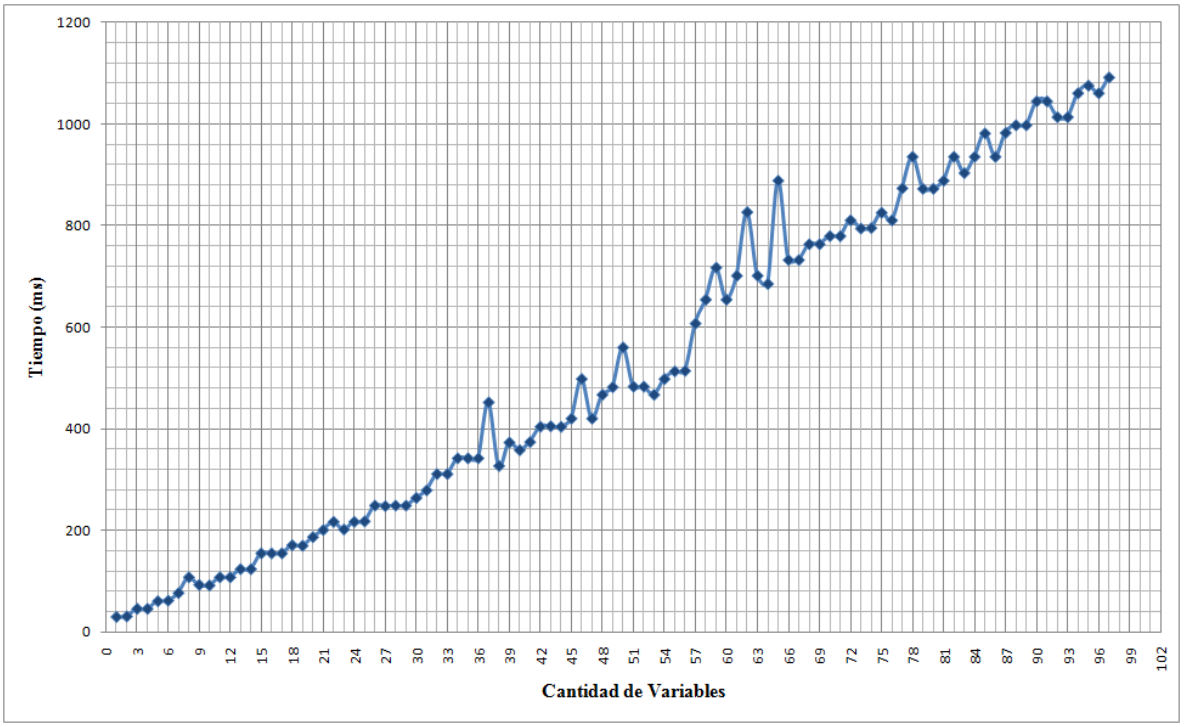


Figura 78. Evolución del tiempo de procesamiento en función de la cantidad de variables

Con el objeto de ratificar estadísticamente el coeficiente de correlación (basado en Pearson) entre la variable *total* y *qvar*, a continuación se expone el intervalo de confianza para las variables mencionadas y el coeficiente dado con un nivel de confianza de 0.95.

Pearson's product-moment correlation
data: parte[, 1] and parte[, 2]
t = 112.9214, df = 1318, **p-value < 2.2e-16**
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.9466804 0.9568162
sample estimates:
cor
0.9520086

Figura 79. Intervalo de Confianza para el Coeficiente de Correlación entre la variable *total* y *qvar* (Nivel de Confianza de 95%)

La probabilidad arrojada por el test es claramente inferior a 0,05. De este modo, existiría correlación y su intervalo de confianza [0,946 ; 0,956], indica claramente que no se corre riesgo de incorporar al 0 dentro del mismo. Esto último, representa un riesgo que podría darse, incluso aunque se dispusiera de un coeficiente de correlación alto, motivo por el cual se justifica el análisis del intervalo de confianza. Así, queda demostrado analítica y gráficamente, que el coeficiente de correlación entre la variable *qvar* y *total* es confiable y expondría una fuerte relación lineal entre ellas.

La Figura 78 muestra la evolución del tiempo de procesamiento en función de la cantidad de variables. El comportamiento del tiempo total en función de la cantidad de variables es aproximadamente lineal, en términos visuales, lo cual es corroborado analíticamente mediante la

matriz de correlación de la Figura 76, la que expone un coeficiente de correlación de 0.95 en ambas variables.

La Figura 78 se ha construido, fijando en 1000 las mediciones (variable *meds*) y obteniendo los datos del tiempo y cantidad de variables sobre sus respectivos ejes. Ello representa en realidad, un corte en el plano sobre gráfico de superficie expuesto en la Figura 72, Figura 73 y Figura 74 respectivamente.

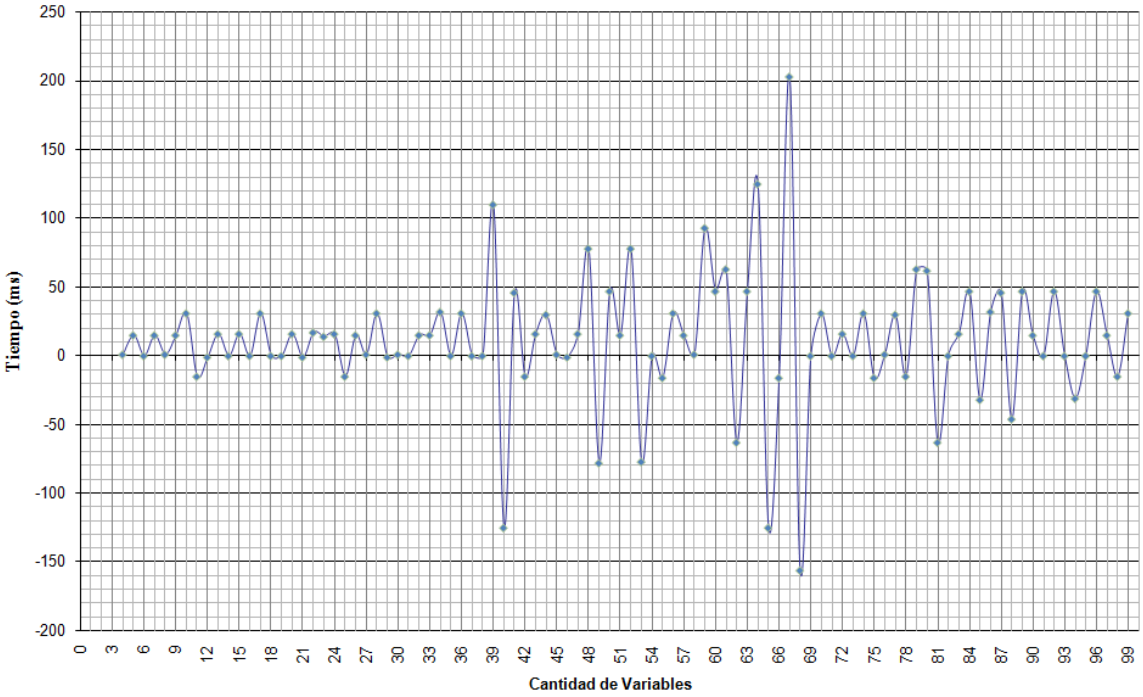


Figura 80. Evolución del tiempo unitario de procesamiento en función de la cantidad de variables

La Figura 80 muestra el incremento unitario de tiempo de procesamiento en milisegundos, a medida que evoluciona la cantidad de variables. De este modo, se puede apreciar que el incremento unitario del tiempo de procesamiento por cada variable que se incorpore (incremento marginal del tiempo), es de ± 50 milisegundos.

Es importante destacar observando a la Figura 80, que cuando el tiempo total de procesamiento está en función de 35 a 70 variables, se presenta un comportamiento atípico. Este mismo comportamiento, ya había sido señalado en el gráfico de superficie en la Figura 72 y en la Figura 73. Dicho comportamiento, posiblemente se manifieste por la transición en el volumen de datos, teniendo en cuenta la cantidad de variables y su localización en los caches primarios, secundarios y memoria principal de la computadora.

Finalmente, debe destacarse que las conclusiones a las que aquí se han arribado, fruto del análisis de los datos surgidos de la simulación, son válidas en la medida que se acoten a la actual simulación y para el escenario planteado. Bajo ningún punto de vista, las conclusiones pueden ser generalizadas a otros escenarios. Para esto último, se requiere de un análisis estadístico de mayor

alcance, en donde se someta nuestra estrategia de procesamiento a otros escenarios, se comparen los resultados de los mismos (sobre la misma base en que aquí se definió para la simulación), y se realicen diferentes contrastes de hipótesis (dócima), para aceptar o no los resultados aquí presentados.

10 Conclusiones

En este capítulo, se concluye con las contribuciones realizadas, resaltando los principales aspectos de las mismas, y finalmente, se analiza el trabajo a futuro, delineando potenciales nuevos enfoques y aspectos adicionales de validación.

10.1 Contribuciones

En esta tesis se presentó el Enfoque Integrado de Procesamiento de Flujos de Datos centrado en Metadatos de Mediciones (EIPFDcMM). El mismo fue diseñado e implementado bajo la premisa de permitir la incorporación de fuentes de datos heterogéneas, cuyos flujos de mediciones estructurados y enriquecidos con metadatos embebidos basados C-INCAMI, permitieron realizar análisis estadísticos de un modo consistente a los efectos de implementar un comportamiento detectivo y a su vez, permitiendo incorporar información contextual a las mediciones, para enriquecer la función de clasificación con el objeto de implementar el comportamiento predictivo. Tanto la implementación del comportamiento detectivo como del predictivo tienen asociados mecanismos de alarma, lo que permite proceder a la notificación ante la eventual identificación de una zona de riesgo.

Para el diseño e implementación del EIPFDcMM, se debieron integrar conocimientos y tecnologías de distintas áreas de investigación, como son el área de Ingeniería del Software, Minería de Datos, Bases de Datos, Estadística, entre otras. Como consecuencia, los aportes realizados son interdisciplinarios.

A continuación se analizan los resultados obtenidos, teniendo en cuanta las principales contribuciones realizadas en las distintas áreas.

10.1.1 Desde el punto de vista de las fuentes de datos y su procesamiento

Los dispositivos de medición, normalmente poseen su propia estructura interna de gestión de datos, por lo que en la gran mayoría de los casos, la recolección de tales datos, se asocia a productos y/o servicios adicionales del fabricante del dispositivo. Por lo que es posible que intervengan diferentes tipos de dispositivos en un proceso de medición, y que pertenezcan a diferentes fabricantes. Esto último, conlleva a un incremento en la complejidad con respecto a la recolección de los datos, con la

situación adicional, de que los datos a recuperar son eso, solo datos, adolecen de información complementaria que permitan inferir su significado o bien contextualizarlos.

El EIPFDcMM plantea por un lado, la incorporación de C-INCAMI junto con su base conceptual ontológica, como elemento fundamental para incorporar metadatos en forma conjunta a los datos, con el objetivo de permitir inferir el significado de un dato como así también, el de contextualizar el entorno donde la medida es obtenida. Por otro lado, permite la integración de cualquier fuente de datos que implemente la interface *DataSource*.

De este modo, el EIPFDcMM contribuye a la repetitividad, consistencia y automatización del proceso de medición y evaluación a partir de data streams, incorporando una estructura a los datos del flujo (sub-sección 6.1), que permite incorporar significado del dato a procesar, sustentado en la base conceptual del marco formal de medición y evaluación C-INCAMI.

Además, el EIPFDcMM contribuye al proceso de medición, permitiendo la incorporación de *cualquier* fuente de datos, independientemente del fabricante, en la medida que implemente la interface *DataSource*. Dicha interface, posee una serie de métodos, cuyos objetivos se encuentran claramente establecidos (sub-sección 5.1), los que permiten establecer la comunicación en un lenguaje homogéneo, entre fuentes de datos y adaptador de mediciones (sub-sección 6.4).

El hecho de incorporar metadatos junto a las mediciones dentro de los flujos de datos, le permite al EIPFDcMM hacer trasparente la asociación de una medida determinada, con respecto a una métrica asociada la que a su vez, puede cuantificar tanto a un atributo de una entidad bajo análisis como a una propiedad de contexto de un entorno o contexto definido. Tales asociaciones, se basan en la definición de un proyecto de M&E, sobre el cual se configuran luego las fuentes de datos (capítulo 5). Esto le permite al EIPFDcMM, entre otras cuestiones, conocer la definición formal de la métrica a la que se asocia la medida (escala, tipo de escala, unidad, etc.) y el nivel de prioridad de una métrica con respecto a otras métricas. De este modo, nuestra estrategia de procesamiento incorpora técnicas de descarte selectivo (sub-sección 6.6), basadas en los metadatos definidos y adicionalmente, sustentados sobre la base conceptual de C-INCAMI, lo que permite evitar desbordes en la cola de servicio bajo conocimiento de la prioridad o importancia relativa del dato que se descarta. De este modo, nuestra estrategia contribuye al proceso de medición, incorporando técnicas que evitan el desborde en la cola de servicios para la recolección de mediciones, analizando conceptualmente la importancia relativa de cada medida, en función de la base conceptual sobre la que se sustenta.

10.1.2 Desde el punto de vista detectivo

Como se mencionó en la sub-sección anterior, normalmente los dispositivos de medición arrojan solo datos, careciendo de información que permitan inferir su significado o bien, contextualizar el entorno donde la medida fue obtenida. En este sentido, independientemente de la obtención del dato, el hecho es ¿cómo saber si el dato obtenido es esperado, racional por ejemplo si el dispositivo sufrió algún desperfecto o bien requiere de una calibración? Si no existiera contra qué comparar el dato, es decir se adoleciera del patrón referencial, no se podría concluir nada al respecto. Es aquí en donde nuestra estrategia realiza una importante

contribución, dado que al incorporar los metadatos al flujo de mediciones, es posible identificar a qué métrica corresponde una medida y por ende, obtener su definición *formal*.

El hecho de contar con la definición formal de una métrica (escala, tipo de escala, unidad, etc.), permite que se conozca instantáneamente, si la medida se encuentra dentro de su dominio de definición y de este modo, identificar inmediatamente si una fuente de datos sufre algún inconveniente o bien, se encuentra descalibrada.

Es más, nuestra estrategia de procesamiento, plantea como contribución adicional, la incorporación de sinopsis (sub-sección 7.7) a diferentes niveles de granularidad y basada en metadatos. Esto permite, por un lado responder a consultas ante cortes no previstos de la fuente de datos, y por otro lado, analizar las sinopsis en diferentes niveles de detalle de la métrica. En este último sentido y dado la estructuración del acceso a las sinopsis, se cuenta con resúmenes estadísticos de una métrica a nivel general o bien, a nivel de grupos de seguimiento. Esto facilita, por un lado, calcular una distancia entre la métrica circunscripta a un grupo de seguimiento con respecto al nivel general, y por otro lado, calcular la distancia entre grupos de seguimiento para dicha métrica a los efectos de determinar un grado de similitud.

Además, el hecho de mantener una sinopsis por cada métrica en cada grupo de seguimiento, es decir, el hecho de mantener el comportamiento estadístico descriptivo que posee cada métrica en cada grupo de seguimiento en particular, permitiría a futuro, incorporar técnicas de agrupamiento (clustering), que identifiquen grupos de seguimiento con comportamientos similares, añadiendo un nuevo tipo de análisis a la estrategia de procesamiento.

Un beneficio derivado de la gestión de sinopsis y sus diferentes niveles de granularidad, radica en la capacidad de identificar instantáneamente outliers para una métrica dada. En efecto, una vez que se obtienen las medidas descriptivas, la determinación de un valor que se comporta en forma diferente a la serie general de datos, corresponde a la aplicación de una fórmula mediante el empleo de rangos inter-cuartiles. Adicionalmente, mediante el análisis de la tasa de cambio del coeficiente de variación, es posible detectar inicialmente, si se está incorporando ruido en la serie de mediciones de una métrica dada. Dado que tales análisis se aplican a nivel de métrica, y en donde cada sinopsis corresponde a una métrica en particular para un nivel de detalle dado (sea a nivel global o por grupo de seguimiento), es posible aplicar la detección de outlier o ruido en cualquiera de dichos niveles.

10.1.3 Desde el punto de vista predictivo

Nuestra estrategia de procesamiento, ha incorporado la implementación algorítimica de clasificación en línea del framework MOA (Bifet, Holmes, et al., 2010). Dicho framework, tiene como fortaleza el hecho de poseer una algorítmica adaptada al procesamiento específico de flujos de datos, lo que permite claramente diferenciarlo de entornos de minería de datos tradicionales. No obstante, no está dentro del alcance de dicho framework, el análisis del significado del dato o su contexto, sino que su objetivo se acota a la algorítmica de clasificación propiamente dicha, entre otras funciones de minería de datos. De este modo, MOA toma el dato como tal, es decir, diferencia los tipos de datos (entero, flotante, etc.) pero desconoce el dominio, unidades, escalas,

tipo de escalas, etc. sobre el que se define cada uno de ellos. Por ejemplo: si se considera una métrica para valor de la temperatura axilar, MOA podría recibir un valor asociado de 150 y lo procesaría, pero la cuestión de fondo es ¿dicho valor es racional? Claramente no, pero ello lo sabemos en base a la definición formal de la métrica. Ahora bien, si la métrica estuviera referida al valor de la temperatura de un horno pirolítico, el valor 150 seguramente sería racional. De este modo, ¿cómo MOA puede diferenciar una u otra situación? Claramente no puede, sin embargo no es su principal objetivo. Allí es donde se incorpora la contribución de nuestra estrategia, el clasificador solo recibe mediciones consistentes, es decir, cuyos valores corresponden a su dominio.

Por lo que el clasificador no sólo se ve beneficiado por la consistencia del dato que le ingresa, lo cual le permite una mayor consistencia en el análisis, sino que también el enfoque se encarga de traducir a datos (para MOA), mediciones asociadas a métricas de atributos de entidad como de propiedades de contexto, incorporando información que permite circunscribir el contexto de medición y siendo capaz, de interpretar el modelo de clasificación, comprendiendo si los discriminadores del modelo de clasificación, corresponden a atributos o propiedades del contexto.

Adicionalmente y gracias a la adecuación de la algorítmica de MOA al procesamiento de flujos de datos, el procesamiento de nuevas mediciones arribadas en los flujos como unidades lógicas de medición (sub-sección 4.3.3), permite actualizar al clasificador no sólo con mediciones de un atributo de la entidad, sino también con la situación del entorno, a través de métricas asociadas con sus propiedades de contexto. De este modo, el clasificador se ajusta tanto a la entidad bajo análisis como a su contexto inmediato del cual parten sus medidas.

Por otra parte, otra contribución interesante de nuestra estrategia de procesamiento, es la asociación de cada clasificador a un grupo de seguimiento. Por ende, existirán tantos clasificadores como grupo de seguimiento, permitiéndose un seguimiento y clasificación personalizado al mismo. La idea de grupo de seguimiento en este sentido, es de suma utilidad al momento de monitorear personas u objetos en movimiento. Esta última situación, fue la presentada en el caso de aplicación de la sub-sección 3.5, donde la entidad bajo análisis corresponde al paciente trasplantado ambulatorio, y gracias al concepto de grupo de seguimiento, fue posible diferenciar a un paciente de otro, incluso adecuando el comportamiento predictivo a las mediciones particulares que arriben para cada paciente. De este modo, se deduce que los mecanismos de alarma terminan estando en función del propio grupo de seguimiento y de su contexto, aunque es posible que rijan normativas generales que apliquen a todos los grupos, tal y como se analizó con los parámetros asociados al analizador estadístico y al tomador de decisiones (sub-secciones 7.11 y 8.6 respectivamente).

Por último, un aspecto considerado en EIPFDcMM es que, independientemente de si se detecta una situación potencialmente anómala en el análisis estadístico o bien durante la clasificación, el tomador de decisiones, en base a los parámetros del proyecto de M&E, tiene la potestad de dar curso u omitir una alarma. En nuestra estrategia de procesamiento, el mecanismo

de alarmas está planteado de tal modo, que es muy simple extender los tipos de notificación para futuras implementaciones (sub-sección 7.10), por lo que sólo es necesario implementar la interface *AlarmCarrier*. De este modo, detectada la situación *en línea*, la misma es notificada al tomador de decisiones junto con la información estadística, si estuviese disponible (capítulo 7), para emplear la instancia de la clase que implemente la interface *AlarmCarrier (*asociada al tomador de decisiones), a los efectos de dar curso a la notificación. Notar, que la peor situación que se informa al tomador de decisiones es detectiva, y si la misma, proviene del clasificador, se estaría actuando preventivamente en base a la experiencia adquirida.

10.1.4 Desde el punto de vista de implementación y simulación del enfoque integrado

El EIPFDcMM cuenta con un prototipo implementado en lenguaje Java. El sistema se ha subdividido en dos paquetes principales: a) *dsips-ma.jar* b) *dsips.jar*. El primer paquete, corresponde al subconjunto de funcionalidades que se implementan para la recolección de mediciones y por ende, es donde reside, entre otros, la interface *DataSource* a implementar por las fuentes de datos. El segundo paquete, contiene la totalidad de la lógica funcional que va desde la recepción de las mediciones a partir de uno o más adaptadores de mediciones, hasta la notificación de alarmas. Adicionalmente, se desarrolló una aplicación web, denominada *dsipsWEB*, la que permitió desplegar los servicios de configuración y transmisión (ver capítulos 5 y 6), haciendo uso del paquete dsips.jar, a través de Apache Tomcat 6.0.20 con balance de carga. De este modo, nuestra estrategia de procesamiento, cuenta con una herramienta prototípica, que aplica todos los conceptos discutidos en esta tesis.

El desarrollo del prototipo ha servido, en primer lugar, como prueba de conceptos del enfoque integrado, considerando sólo aquéllos fundamentos, métodos y procedimientos empleados en esta tesis. De este modo, queda abierta la posibilidad no sólo a incorporar mejoras conceptuales a nuestra estrategia de procesamiento, sino también a discutir aspectos puntuales de implementación de nuevos procedimientos, que deriven en una mejora del prototipo. Por ejemplo, en un futuro, se podría plantear el reemplazo de MOA, por algoritmos de clasificación en línea basados en implementaciones en lenguaje C y enlazadas a clases Java mediante JNDI. Adicionalmente, podrían plantearse pruebas de rendimiento del prototipo actual versus las nuevas implementaciones, para establecer una medida comparativa de mejora. En segundo lugar, el prototipo fue de suma utilidad para el diseño y ejecución de la simulación del caso de estudio presentado en la sub-sección 3.5.

De este modo, el prototipo es en definitiva, una importante contribución asociada a esta tesis, por cuanto no sólo permite validar aspectos conceptuales de la misma, sino que incorpora la posibilidad de plantear mejoras en términos de implementación, que adecúen al prototipo en otros contextos de procesamiento, tales como entornos móviles, computación en grilla, etc.

A partir del prototipo del EIPFDcMM, se planificó, diseño y ejecutó en forma experimental, una simulación (capítulo 9) sobre el caso de aplicación de la sub-sección 3.5. Tal simulación, permitió validar inicialmente la estrategia de procesamiento, con respecto a la integración de

fuentes de datos, como así también, en cuanto a la consistencia de los datos y su posterior análisis. De dicha simulación, surgieron resultados a destacar tales como:

1. *Se obtuvo un umbral máximo de procesamiento de 1 segundo, para procesar 99 variables (o métricas) con 1000 mediciones cada una*

2. *El tiempo total de procesamiento es afectado primariamente por el incremento en la cantidad de variables (o métricas) y en mucho menor medida por el incremento de las mediciones*

3. *El incremento de la variabilidad del sistema está mayormente influido por el incremento en la cantidad de variables (o métricas)*

4. *El análisis descriptivo es quien insume la mayor proporción del tiempo total de procesamiento*

Cabe indicar que la simulación corresponde a una necesidad experimental de validar inicialmente la estrategia de procesamiento sobre un caso de aplicación, pero ello no supone que sus resultados sean estadísticamente válidos (en el sentido de validez externa o de generalización de conclusiones). Los resultados enunciados solo son válidos, en la medida que se circunscriban al caso de aplicación planteado. Para lograr que tales resultados sean estadísticamente válidos y generalizables, la estrategia de procesamiento debiera ser sometida a diferentes escenarios, basarse en la misma definición de variables de la simulación, recolectar las mediciones y plantear diferentes contrastes de hipótesis para aceptar o no, cada una de las conclusiones parciales enumeradas anteriormente. Hecha esta aclaración, si se analiza el tiempo máximo obtenido en la simulación, para detectar anomalías en pacientes trasplantados ambulatorios, la estrategia de procesamiento ajustaría sin inconvenientes para su aplicación en la detección de riesgos en la salud de los mismos, dado que en solo 1 segundo, se estaría analizando el conjunto de indicadores del paciente como de su contexto, para proceder a notificar ante un riesgo en la salud del mismo.

Otro aspecto interesante a destacar de nuestra estrategia de procesamiento, es que mediante el análisis estadístico en línea del flujo de mediciones, permite detectar asociaciones entre métricas, sean éstas dentro del grupo de seguimiento o bien, entre métricas de diferentes grupos de seguimiento. Esto representa una contribución importante, por cuanto permite detectar en tiempo real, situaciones de arrastre de una métrica sobre otras, que pudieran significar que la entidad bajo análisis, termine en una zona de riesgo. Esta última situación ha sido esquematizada en la Figura 16, del caso de aplicación presentado en la sub-sección 3.5.

10.2 Trabajo a Futuro

Una línea de avance está asociada a extender las funciones de minería de datos sobre nuestra estrategia de procesamiento. En este sentido, nuestra estrategia sólo ha abordado la clasificación y en particular los árboles de clasificación incrementales, quedando abierta la posibilidad de incorporar otras estrategias de clasificación, como por ejemplo redes bayesianas o

bien, incorporar nuevas capacidades funcionales, como el agrupamiento (clustering) en grupos de seguimiento, entre otras. De hecho, actualmente se está avanzando en una línea de investigación de modelado de información de contexto y algoritmos para el cálculo de similitud de contexto, llevada a cabo por otros integrantes de nuestro grupo de I+D (Molina, Rossi, et al., 2010), cuyos resultados podrían complementar, la aplicación de técnicas de agrupamiento sobre flujos de mediciones en línea.

Si bien nuestra estrategia, ha planteado las diferentes componentes que intervienen desde la recolección de mediciones hasta la notificación de alarmas, no se ha formalizado el proceso tal como se ha realizado en Becker et al (Becker, Molina, et al., 2010). Por lo que un trabajo a futuro será la formalización del proceso asociado al enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones.

Finalmente, otro trabajo a futuro (como se ha indicado en la sub-sección 10.1.4), se corresponde con la necesidad someter experimentalmente nuestra estrategia de procesamiento a diferentes escenarios, basado en el diseño, planificación y ejecución de la simulación del capítulo 9, a los efectos validar estadísticamente los resultados parciales obtenidos.

11 Referencias

Abajo Martínez, N. (2004). "ANN quality diagnostic models for packaging manufacturing: an industrial data mining case study". ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). *pp.* 799-804. Seattle (USA). ACM Press.

Aleman-Beza, B., Halaschek, C., Arpinar, B. & Sheth, A. (2003). "Context-Aware Semantic Association Ranking". Semantic Web and Databases Workshop. pp. 33-50. Berlin (Germany).

Arasu, A., Babu, S. & Widom, J. (2006). "The CQL Continuos Query Language: Semantics Foundation and Query Execution". Very Large Ddatabase (VLDB) Journal. Vol. 15 (2), pp. 121-142.

Arasu, A., Chaudhuri, S. & Kaushik, R. (2008). "Transformation-based Framework for Record Matching". International Conference on Data Engineering (ICDE). pp. 40-49. Cancun (Mexico). IEEE.

Arasu, A., Ganti, V. & Kaushik, R. (2006). "Efficient Exact Set-Similarity Joins". ACM Very Large Databases (VLDB). pp. 918-929. Seoul (Korea). ACM Press.

Babcock, B. & Chaudhuri, S. (2005). "Towards a Robust Query Optimizer: A Principled and Practical Approach". ACM Special Interest Group on Management of Data (SIGMOD). pp. 119-130. Maryland (USA). ACM Press.

Babcock, B., Babu, S., Datar, M., Motwani, R. & Thomas, D. (2004). "Operator Scheduling in Data Stream Systems". Very Large Database (VLDB) Journal. Vol. 13 (4), pp. 333-353.

Babcock, B., Datar, M. & Motwani, R. (2004). "Load Shedding for Aggregation Queries over Data Streams". International Conference on Data Engineering (ICDE). pp. 350-361. Boston (USA). IEEE.

Babu, S. & Duan, S. (2007). "Processing Forecasting Queries". ACM Special Interest Group on Management of Data (SIGMOD). pp. 711-722. Beijing (China). ACM Press.

Babu, S., Bond, C., Chandramouli, B. & Yang, J. (2007). "Query Suspend and Resume". ACM Special Interest Group on Management of Data (SIGMOD). pp. 557-568. Beijing (China). ACM Press.

Barnett, V. & Lewis, T. (1994). "Outliers in Statistical Data". (3rd ed.). Wiley.

Basili, V., Caldiera, G. & Rombach, D. (1994). "The Goal Question Metric Approach. En Encyclopedia of Software Engineering". Vol. I, pp. 528-532. Wiley.

Becker, P. & Olsina, L. (2010). "Towards Support Processes for Web Projects". International Conference on Web Engineering (ICWE). pp. 102-113. Vienna (Austria). Springer.

Becker, P., Molina, H. & Olsina, L. (2010). "Measurement and evaluation as a quality driver". Journal Ingénierie des Systèmes d'Information (JISI). Vol. 15 (6), pp. 33-62.

Bifet, A., Holmes, G., Kirkby, R. & Pfahringer, B. (2010). “MOA: Massive Online Analysis”. Journal of Machine Learning Research. Vol. XI, pp. 1601-1604.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R. & Gavaldà, R. (2009). “New Ensemble Methods For Evolving Data Streams”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 139-148. Paris (France). ACM Press.

Botan, I., Alonso, G., Fischer, P., Kossmann, D. & Tatbul, N. (2009). “Flexible and Scalable Storage Management for Data-Intensive Stream Processing”. Extending Database Technology (EDBT). pp. 934-945. Saint Petersburg (Russia). ACM Press.

Botan, I., Cho, Y., Derakhshan, R., Lindar, N., Gupta, A., Haas, L., Kim, K., Lee, C., Mundada, G., Shan, M., Tatbul, N., Yan, Y., Yun, B. & Zhang, J. (2010). “A Demonstration of the MaxStream Federated Stream Processing System”. International Conference on Data Engineering (ICDE). pp. 1093-1096. California (USA). IEEE.

Box, G. & Jenkins, M. (1991). “Time Series Analysis”. Prentice Hall.

Chakravarthy, S. & Jiang, Q. (2009). “Stream Data Processing: A Quality of Service Perspective”. Springer.

Chakravarti, I., Laha, R. & Roy, J. (1967). “Handbook of Methods Applied Statistics”. Vol. I. Wiley.

Chambers, J., Cleveland, W., Kleiner, B. & Tukey, P. (1983). “Graphical Methods for Data Analysis”. Duxbury Press.

Chaudhry, N., Shaw, K. & Abdelguerfi, M. (Eds.). (2005). “Stream Data Management”. Springer.

Congreso de la República Argentina. (1969). “Capítulo 3: De los Productos Alimenticios, Condiciones Generales”. En Código Alimentario Argentino. Buenos Aires (Argentina).

Dey, A. (2001). “Understanding and Using Context”. Journal of Personal and Ubiquitous Computing. Vol. 5 (1), pp. 4-7.

Diao, Y., Liu, A., Peng, L. & Sutton, C. (2009). “Capturing Data Uncertainty in High-Volume Stream Processing”. Biennial Conference on Innovative Data Systems Research (CIDR). California (USA).

Diggle, P., Heagerty, P., Liang, K. & Zeger, S. (2002). “Analysis of Longitudinal Data” (2nd ed.). Oxford (England). Oxford University Press.

Diván, M. (2006). “Construcción de un modelo de centro de distribución celular para acopio de miel en la cooperativa de Doblas”. Tesis de Maestría en Administración de Negocios en UTN – Facultad Córdoba (1st ed.). Santa Rosa, La Pampa (Argentina). Divsar.

Diván, M. (2007). “Introducción a la Tecnología de la Información para Profesionales de Ciencias Económicas”. Santa Rosa, La Pampa (Argentina). Facultad de Ciencias Económicas y Jurídicas , Universidad Nacional de La Pampa (UNLPAM).

Diván, M. (2011). “Reunión y Ordenamiento de Flujos de Datos Simultáneos y Concurrentes Basados en C-INCAMI”. La Plata (Argentina). Tesis de Especialista en Cómputo de Altas Prestaciones y Tecnología GRID. Facultad de Informática. Universidad Nacional de La Plata.

Diván, M. & Olsina, L. (2008). “Integrando Enfoques de Medición y Evaluación con Minería de Datos y Procesamiento de Flujos”. X Workshop de Investigadores en Ciencias de la Computación. General Pico, La Pampa (Argentina). Editorial de la Universidad Nacional de La Pampa (EdUNLPam).

Diván, M. & Olsina, L. (2009a). “Enfoque Integrado para el Procesamiento de Flujos de Datos: Un Escenario de Uso”. Conferencia Iberoamericana de "Software Engineering" (CIbSE). pp. 374-387. Medellín (Colombia).

Diván, M. & Olsina, L. (2009b). “Especificando Fuentes de Datos en el Esquema Integrado de Procesamiento de Flujos”. Congreso Argentino de Ciencias de la Computación (CACIC). San Salvador de Jujuy, Jujuy (Argentina). Universidad Nacional de Jujuy.

Diván, M., Molina, H. & Olsina, L. (2008). “Hacia un Modelo Integrado de Procesamiento de Flujos de Datos”. Congreso Argentino de Ciencias de la Computación (CACIC). Workshop WISBD. Chilecito, La Rioja (Argentina). Universidad Nacional de Chilecito.

Diván, M., Olsina, L. & Gordillo, S. (2011a). “Procesamiento de Flujos de Datos Enriquecidos con Metadatos de Mediciones: Un Análisis Estadístico”. Congreso Iberoamericano de "Software Engineering" (CIbSE). Rio de Janeiro (Brasil). Pontifícia Universidade Católica (PUC) do Rio de Janeiro.

Diván, M., Olsina, L. & Gordillo, S. (2011b). “Simulación de un Enfoque Integrado de Procesamiento de Flujos Aplicado a un Escenario de Pacientes”. Jornadas Argentinas de Informática e Investigación Operativa (JAIIO). Córdoba (Argentina). Sociedad Argentina de Informática (SADIO).

Diván, M., Olsina, L. & Gordillo, S. (2012). “Strategy for Data Stream Processing based on Measurement Metadata: An Outpatient Monitoring Scenario”. Special Interest Group on Health Informatics (SIG-HIT) International Health Informatics (IHI) Symposium. En Evaluación. Miami, Florida (USA). ACM SIG-HIT .

Dodge, G. & Gorman, T. (2000). “Essential Oracle8i Data Warehousing”. New York (USA). Wiley.

Dolin, R., Alschuler, L., Beebe, C., Biron, P., Lee Boyer, S., Essin, D., Kimber, E., Lincoln, T. & Mattison, J. (2001). “The Practice of Informatics. Review: The HL7 Clinical Document Architecture”. Journal of the American Medical Informatics Association (JAMIA). Vol. 8, pp. 552-569.

Domingos, P. (2000). “A Unified Bias-Variance Descomposition and its Applications”. International Conference on Machine Learning (ICML). pp. 231-238. California (USA).

Domingos, P. & Hulten, G. (2000). “Mining High-Speed Data Streams”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 71-80. Boston (USA). ACM Press.

Duin, R., Tortorella, F. & Marrocco, C. (2008). “Maximizing the area under the ROC curve by pairwise feature combination”. Journal of Pattern Recognition. Vol. 41 (6), pp. 1961-1974.

Fan, W., Chu, F., Wang, H. & Yu, P. (2002). “Pruning and Dynamic Scheduling of Cost-Sensitive Ensembles”. National Conference on Artificial Intelligence. pp. 146-151. Alberta (Canada). ACM Press.

Frank, E. & Witten, I. (2005). “Data Mining. Practical Machine Learning Tools and Techniques”. Morgan Kaufmann.

Gama, J., Rocha, R. & Medas, P. (2003). “Accurate Decision Tree for Mining High-Speed Data Streams”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 523-528. Washington (USA). ACM Press.

Getta, L. & Vossough, E. (2003). “Optimization of Data Stream Processing”. ACM Special Interest Group on Management of Data (SIGMOD) Record. Vol. 33 (3), pp. 34-39.

Golab, L. & Özsu, M. (2003). “Data Stream Management Issues –A Survey”. Technical Report, School of Computer Science, University of Waterloo.

Han, J. & Lamber, M. (2001). “Data Mining. Concepts and Techniques”. San Francisco (USA). Morgan Kaufmann.

Hornik, K. (2011). “Frequently Asked Questions on R”. Última actualización el 22 de Febrero de 2011. Último acceso el 21 de marzo de 2011. The Comprehensive R Archive Network en http://cran.r-project.org/doc/FAQ/R-FAQ.html.

Hulten, G., Spencer, L. & Domingos, P. (2001). “Mining Time-Changing Data Streams”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 97-106. California (USA). ACM Press.

Hwang, J., Balazinska, M., Rasin, A., Çetintemel, U., Stonebraker, M. & Zdonik, S. (2005). “High Availability Algorithms for Distributed Stream Processing”. IEEE International Conference on Data Engineering (ICDE). pp. 779-790. Tokio (Japón). IEEE.

Ihaka, R. & Gentleman, R. (1996). “R: A language for data analysis and graphics”. Journal of Computational and Graphical Statistics. Vol. 5 (3), pp. 299-314.

Jahnke, J. (2005). “Toward Context-Aware Computing in Clinical Care”. Object-Oriented Programming, Systems, Language and Applications (OOPSLA). California (USA).

Jin, R. & Agrawal, G. (2003). “Efficient Decision Tree Construction on Streaming Data”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 571-576. Washington (USA). ACM Press.

Johnson, D. (1998). “Métodos Multivariados Aplicados al Análisis de Datos”. Thomson Editores.

Kaplan, R. & Norton, D. (1992). “The Balanced Scorecard – Measures That Drive Performance”. Harvard Business Review. Vol. 70 (1), 71-79.

Kaplan, R. & Norton, D. (1993). “Putting the Balanced Scorecard to Work”. Harvard Business Review. Vol. 71 (5), pp. 134-147.

Kaplan, R. & Norton, D. (1996). “Using the Balanced Scorecard as a Strategic Management System”. *Harvard Business Review , 71* (1), pp. 75-85.

Kaplan, R. & Norton, D. (2000). “Cómo Utilizar el Cuadro de Mando Integral”. Buenos Aires (Argentina). Editorial Gestión 2000.

Kirkby, R. (2007). “Improving Hoeffding Tree”. PhD Tesis of Computer Science, University of Waikato, Hamilton (New Zealand).

Koudas, N. & Srivastava, D. (2005). “Data Stream Query Processing: A Tutorial”. International Conference on Data Engineering (ICDE). pp. 1145. Tokyo (Japan). IEEE.

Langsam, Y., Augenstein, M. & Tenenbaum, A. (1997). “Estructuras de Datos con C y C++” (2nd ed.). Prentice Hall.

Li, M., Ganesan, D. & Shenoy, P. (2006). “PRESTO: Feedback-driven Data Management in Sensor Networks”. Symposium on Networked Systems Design & Implementation. California (USA).

Lindar, N., Güc, B., Lau, P., Özal, A., Soner, M. & Tatbul, N. (2009). “DejaVu: Declarative Pattern Matching over Live and Archived Streams of Event”. ACM Special Interest Group on Management of Data (SIGMOD). pp. 1023-1026. Rhode Island (USA). ACM Press.

Medhat Gaber, M., Zaslavsky, A. & Krishnaswamy, S. (2005a). “Resource-aware Mining of Data Streams”. Universal Computer Science. Vol. 11 (8), pp. 1440-1453.

Medhat Gaber, M., Zaslavsky, A. & Krishnaswamy, S. (2005b). “Mining Data Streams: A Review”. ACM Special Interest Group on Management of Data (SIGMOD) Record. Vol. 34 (2), pp. 18-26.

Molina, H. & Olsina, L. (2007). “Towards the Support of Contextual Information to a Measurement and Evaluation Framework”. International Conference on the Quality of Information and Communications Technology (QUATIC). pp. 154–163. Lisboa (Portugal). IEEE.

Molina, H., Rossi, G. & Olsina, L. (2010). “Context-Based Recommendation Approach for Measurement and Evaluation Projects”. Journal of Software Engineering and Applications (JSEA). Vol. 3 (12), pp. 1089-1106.

Namit, J., Gehrke, J. & Balakrishan, H. (2008). “Towards a Streaming SQL Standard”. Proceedings of the VLDB Endowment. Vol. 1 (2), pp. 1379-1390.

Olsina, L. & Martín, M. (2004). “Ontology for Software Metrics and Indicators”. Journal of Web Engineering (JWE). Vol. 3 (4), pp. 262-281.

Olsina, L., Molina, H. & Papa, F. (2005). “Organization-Oriented Measurement and Evaluation Framework for Software and Web Engineering Projects”. International Conference on Web Engineering (ICWE). Vol. 3579/2005, pp. 351-361. Sydney (Australia). Springer Berlin / Heidelberg.

Olsina, L., Papa, F. & Molina, H. (2007). “How to Measure and Evaluate Web Applications in a Consistent Way”. Chapter 13 in Web Engineering: Modelling and Implementing Web Applications. pp. 385–420. Springer Book HCIS.

Ort, E. & Mehta, B. (2003). “Java Architecture for XML Binding”. Oracle Sun Developer Network. Último acceso el 17 de Marzo de 2011, desde Oracle Corporation en http://www.oracle.com/technetwork/articles/javase/index-140168.html.

Oza, N. & Rusell, S. (2001). “Online Bagging and Boosting. International Workshop on Artificial Intelligence and Statistics”. pp. 105-112. Florida (USA). Morgan Kaufmann.

Pérez López, C. (2005). “Métodos Estadísticos Avanzados con SPSS”. Madrid (España). Thomson.

Petersa, R., Becketta, N., Forettea, F., Tuomilehtoa, J., Ritchiea, C., Waltona, I., Waldmana, A., Clarkea, R., Poultera, R., Fletchera, A. & Bulpitta, C. (2009). “Vascular risk factors and cognitive function among 3763 participants in the Hypertension in the Very Elderly Trial (HYVET): a cross-sectional analysis”. International Psychogeriatrics (Cambridge Journals). Vol. 21 (2), pp. 359-368.

Rosen, L. (2005). “Open Source Licensing. Software Freedom and Intellectual Property Law” (1st ed.). Massachuetts (USA). Prentice Hall - Professional Technical Reference (PTR).

Rundensteiner, W., Mani, M. & Wei, M. (2008). “Utility-driven Load Shedding for XML Stream Processing”. International World Wide Web (WWW) Conference. pp. 855-864. Beijing (China). ACM Press.

Ryvkina, E., Maskey, A., Cherniack, M. & Zdonik, S. (2006). “Revision Processing in a Stream Processing Engine: A High-Level Design”. IEEE International Conference on Data Engineering (ICDE). pp. 141. Atlanta (USA). IEEE.

Singh, S., Vajirkar, P. & Lee, Y. (2003a). “Context-Based Data Mining using Ontologies”. Lecture Notes in Computer Science. Vol. 2813, pp. 405-418. Springer Berlin / Heidelberg.

Singh, S., Vajirkar, P. & Lee, Y. (2003b). “Context-aware data mining framework for wireless medical application”. Lectures Notes in Computer Science of Springer. Vol. 2736, pp. 381-391.

Srivastava, U. & Widom, J. (2004). “Flexible Time Management in Data Stream Systems”. ACM Principles of Database Systems (PODS). pp. 263-274. Paris (Francia). ACM Press.

Stamou, G. & Wallace, M. (2002). “Towards a Context Aware Mining of User Interests for Consumption of Multimedia Documents”. IEEE International Conference on Multimedia & Expo (ICME). Vol. 1, pp. 733-736. Lausanne (Switzerland). IEEE.

Stephens, M. (1974). “EDF Statistics for Goodness of Fit and Some Comparisons”. Journal of the American Statistical Association. Vol. 69, pp. 730-737.

Street, W. & Kim, Y. (2001). “A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 377-382. California (USA). ACM Press.

Tatbul, N. (2010). “Streaming Data Integration: Challenges and Opportunities”. International Conference on Data Engineering (ICDE). International Workshop on New Trends in Information Integration (NTII). pp. 155. California (USA). IEEE.

Tatbul, N. & Zdonik, S. (2006). “Window-aware Load Shedding for Aggregation Queries over Data Streams”. ACM Very Large Database (VLDB). pp. 799-810. Seoul (Korea). ACM Press.

Toman, D. (2009). “Data Expiration and Aggregate Queries”. Alberto Mendelzon International Workshop on Foundations of Data Management (AMW). Arequipa (Peru).

Tranh, T., Sutton, C., Cocci, R., Nie, Y., Diao, Y. & Shenoy, P. (2009). “Probabilistic Inference over RFID Streams in Mobile Environments”. International Conference on Data Engineering (ICDE). pp. 1096-1107. Shanghai (China). IEEE.

Urbanek, S. (2003). “Rserve. A Fast Way To Provide R Funcionality to Applications”. Workshop on Distributed Statistical Computing (DSC). pp. 1-11. Vienna (Austria).

Wang, H., Fan, W., Yu, P. & Han, J. (2003). “Mining Concept-Drifting Data Streams using Ensemble Classifiers”. ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). International Conference on Knowledge Discovery and Data Mining. pp. 226-235. Washington (USA). ACM Press.

Zhong, E., Fan, W., Yang, Q., Verscheure, O. & Ren, J. (2010). “Cross Validation Framework to Choose Amongst Models and Datasets for Transfer Learning”. Journal of Machine Learning and Knowledge Discovery in Databases. Vol. 6323, pp. 547-562.

Apéndice A. Base de Datos C-INCAMI

En este apéndice, nos proponemos analizar los aspectos conceptuales destacados de la base de datos C-INCAMI (referida como C-INCAMI DB en la Figura 11) y como éstos inciden en el comportamiento del prototipo asociado al EIPFDcMM.

En primer lugar, el apéndice abordará los aspectos conceptuales asociados a la definición de un proyecto de M&E, el cual sirve como punto de partida para llevar adelante la configuración de las fuentes de datos, oportunamente analizadas en el capítulo 5 y 6.

Seguido, se analizan los aspectos asociados a la definición de una métrica, sea que ésta se asocie a un atributo o bien, a una propiedad de contexto. Estos aspectos son fundamentales al momento de definir los grupos de seguimiento, recolección de mediciones, etc. como así también su impacto en la organización de las mediciones para su posterior análisis estadístico, tal y como se ha expuesto en los capítulos 4 y 6.

Por último, se analizan los conceptos asociados a la definición de los indicadores y los mecanismos de alarmas. Tales definiciones, son de vital importancia a los efectos de guiar el comportamiento de las técnicas estadísticas, los clasificadores on-line, como así también las alarmas que pudiesen dispararse, tal y como se analizó en los capítulos 7 y 8, junto con la simulación del capítulo 9.

A.1 Modelo Conceptual asociado a la definición del proyecto de M&E

El modelo conceptual subyacente a la base de dato C-INCAMI, se basa en una aproximación relacional del marco de M&E C-INCAMI, ya presentado en la sub-sección 2.2.3 y el capítulo 4. Dado que el modelo de base de datos lógico, asociado con la base de datos C-INCAMI, ha sido desplegado sobre un contexto relacional y partiendo de que el marco C-INCAMI se sustenta en el paradigma orientado a objetos, es lógico que nuestra aproximación haya requerido algún ajuste en términos de implementación, que permitan compatibilizar ambos paradigmas, en post de mantener intactas las cuestiones conceptuales del marco.

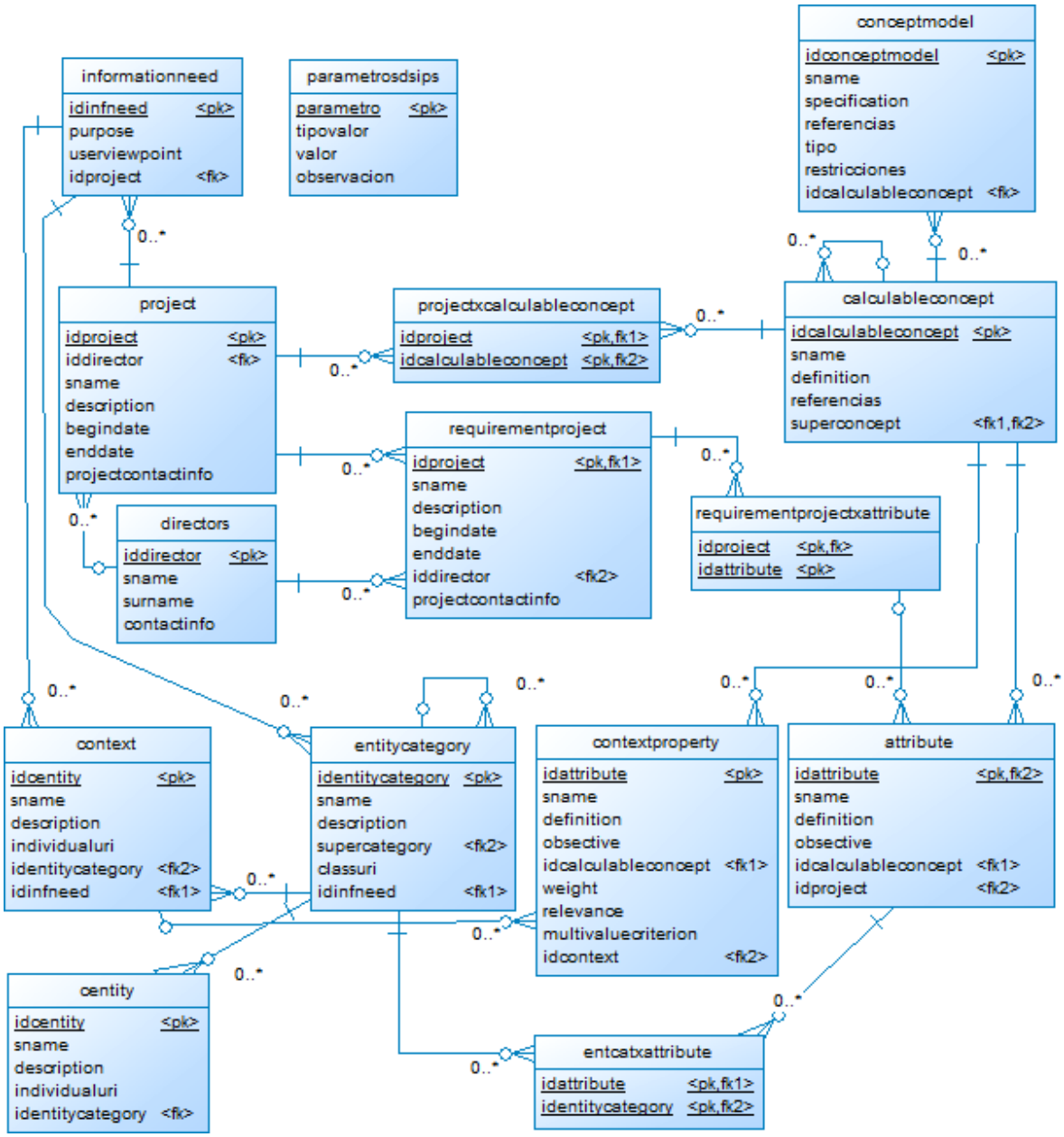


Figura 81. Vista Relacional del sub-esquema de la base de datos C-INCAMI asociado con la definición del proyecto de M&E

La Figura 81 muestra el sub-esquema de la base de datos C-INCAMI asociado con la definición del proyecto de M&E. Como un aspecto particular, note que en la mencionada figura se encuentra una tabla denominada *parametrosdsips.* Ésta última, en realidad contiene una serie de parámetros que rigen el comportamiento del prototipo en términos operativos, pero que no contiene relación directa con los conceptos del marco, por ello ha sido expuesta como *sin relación* con respecto a los restantes conceptos. Un ejemplo de esto último es el parámetro STATISTICAL_ALARM (presentado entre otros, en la sub-sección 7.11) cuya finalidad es indicar al *DecisionMaker* si dará curso o no a las alarmas surgidas desde el análisis estadístico.

El sub-esquema contiene la tabla *Project* la cual almacena la definición de los proyectos de M&E, se vincula con *InformationNeed* a los efectos de establecer qué necesidad de información satisface. La relación entre *project* y *directors* determina el rol del coordinador global del proyecto de M&E, mientras que la relación entre *requirementproject* y *directors* señala la presencia de un sub-director particular para un requerimiento dentro del proyecto.

El proyecto estará vinculado a varios conceptos calculables (*projectxcalculableconcept*). Cada concepto calculable (*calculableconcept*) puede asociarse con un modelo conceptual (*conceptmodel*). Adicionalmente, puede tener vinculado varios atributos (*attribute*) o bien, varias propiedades de contexto (*contextproperty*). Cada requerimiento del proyecto puede asociarse con muchos atributos de un concepto calculable (*requirementprojectxattribute*).

Una necesidad de información (*informationneed*), puede tener asociada varias definiciones de contexto (*context*) como así también, puede asociarse con varias categorías de entidad (*entitycategory*). A una categoría de entidad, puede definírsele un contexto en particular y dicho contexto, es caracterizado por sus propiedades de contexto (*contextProperty*). A su vez, cada categoría de entidad es caracterizada por una serie de atributos (*entcatxattribute*), y dentro de una categoría de entidad, pueden existir varias entidades a monitorear (*centity*)

Cada uno de estos conceptos, deben estar previamente definidos, para poder proceder a la especificación de las métricas e indicadores asociados. Esto último es lógico, por cuanto y por ejemplo, para poder asociar una métrica a un atributo determinado, debiera tenerse noción del concepto calculable y/o el requerimiento del proyecto de M&E específico al cual se vinculará.

A.2 Modelo conceptual asociado a la definición de la métrica

La Figura 82 muestra el sub-esquema de la base de datos C-INCAMI asociado con la definición de la métrica. Todas las métricas son definidas en la entidad *metric*, allí puede apreciarse la relación con el atributo (*attribute*) o propiedad de contexto (*contextproperty*) que implementa. Adicionalmente, puede observarse la definición del error máximo permitido para la métrica, como así también los valores de referencia para la misma (atributos mínimo, máximo y normal). Toda métrica tiene asociada una escala numérica (*numericalscale*), la cual a su vez se vincula con la tabla *scale.*

Cada métrica puede tener asociadas varias mediciones (*measurement*). Estas mediciones pueden o no ser deterministas, por lo que una medición puede tener asociada varias medidas con su correspondiente probabilidad (*measure*). Un conjunto de mediciones puede asociarse con *trainingdataset,* cuyo objetivo es definir un grupo representativo de mediciones para una métrica. Si bien y a priori, ello parecería contradictorio con el planteo de procesamiento de flujos de datos on-line, no lo es en absoluto, por cuanto dicho grupo representativo de mediciones, constituye el *dataset* para entrenar a los clasificadores on-line en el momento 0 (cero). Es por ello, que se mencionaba dentro de la tesis, que el grupo de mediciones para entrenar un clasificador (ver sub-sección 8.5), podría ser provisto por especialistas del área bajo medición. Así, tales observaciones estarían almacenadas en las entidades *measurement*, *measure* y *trainingdataset* para ser

utilizadas automáticamente al momento en que el clasificador sea inicializado para un proyecto de medición dado.

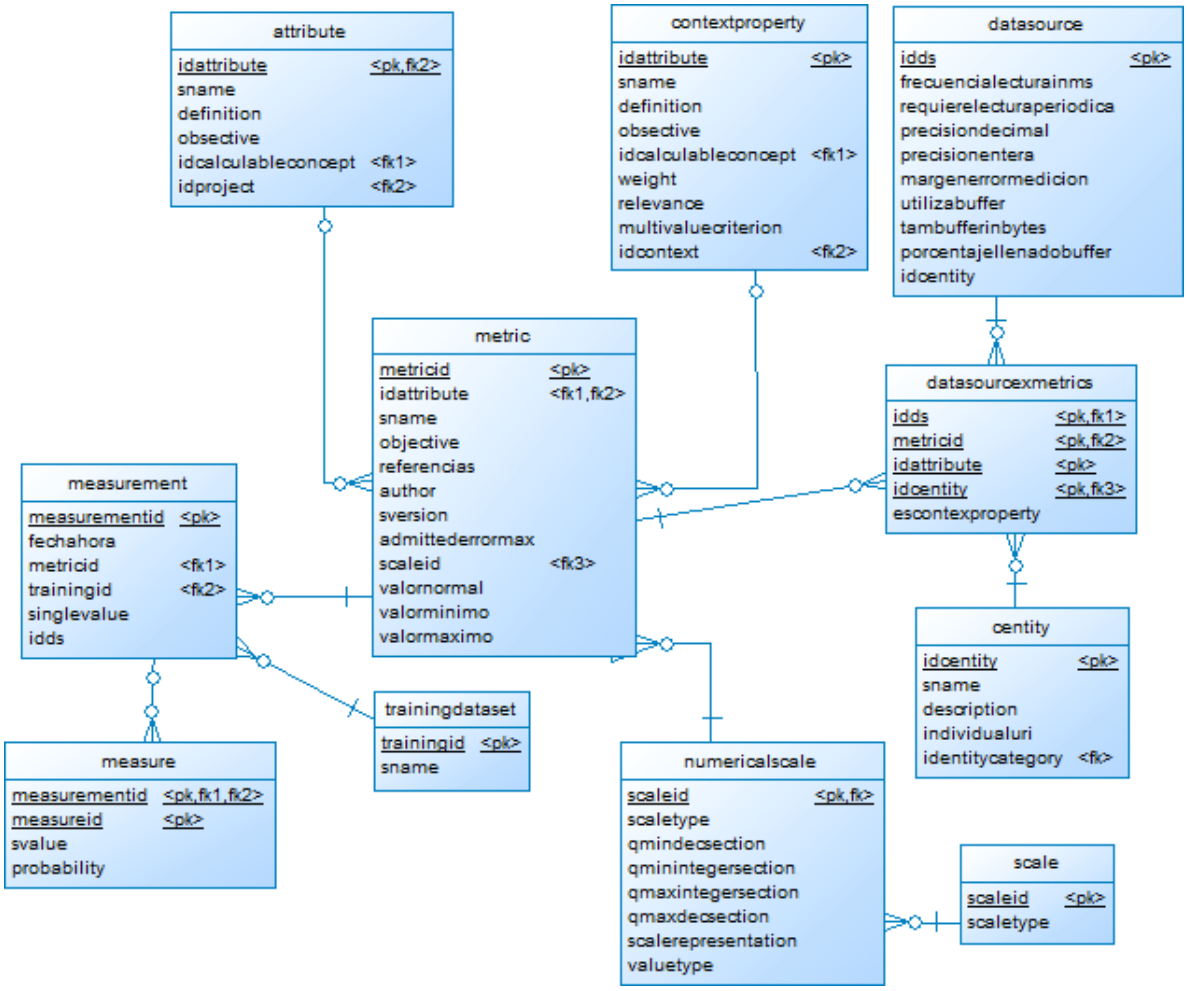


Figura 82. Vista Relacional del sub-esquema de la base de datos C-INCAMI asociado con la definición de la métrica

La tabla *datasource* almacena las fuentes de datos que han implementado la interface *Datasource* (ver sub-sección 5.1), y han llevado adelante el proceso de configuración descrito en el capítulo 5. De este modo, se identifica a cada una de las fuentes de datos que interactúa con el prototipo, permitiendo entre otras cuestiones un resguardo o copia de seguridad de la configuración de la misma con respecto a uno o más proyectos de medición. Es decir, que si la fuente de datos por algún motivo sufriera una desconfiguración, mediante los servicios web presentados en el capítulo 5, podría recuperar la información de configuración para evitar llevar adelante todo el proceso nuevamente.

Recapitulando el proceso de configuración de la fuente de datos explicado en la sub-sección 5.5, la fuente de datos debía indicar con qué métrica se asociaría junto con la entidad bajo estudio a la que mediría. Esto último, es lo que almacena la entidad *datasourcexmetrics,* es decir que su función es mantener la relación para cada una de las métricas con las que se asocia una

fuente de datos, la entidad que mide específicamente dicha métrica (para esa fuente de datos) y si el atributo en cuestión, es o no una propiedad contextual. En éste último sentido, cabe aclarar que tanto *attribute* como *contextproperty* poseen idéntica clave primaria, con lo que desde el punto de vista relacional podrían haber sido fusionados bajo una misma entidad, no obstante ello, se ha primado su separación en post de mantener claramente diferenciado los conceptos vinculados a cada uno de ellos.

A.3 Modelo Conceptual asociado a la definición de indicadores y mecanismos de alarma

La Figura 83 muestra el sub-esquema de la base de datos C-INCAMI asociado con la definición de indicadores y mecanismos de alarma.

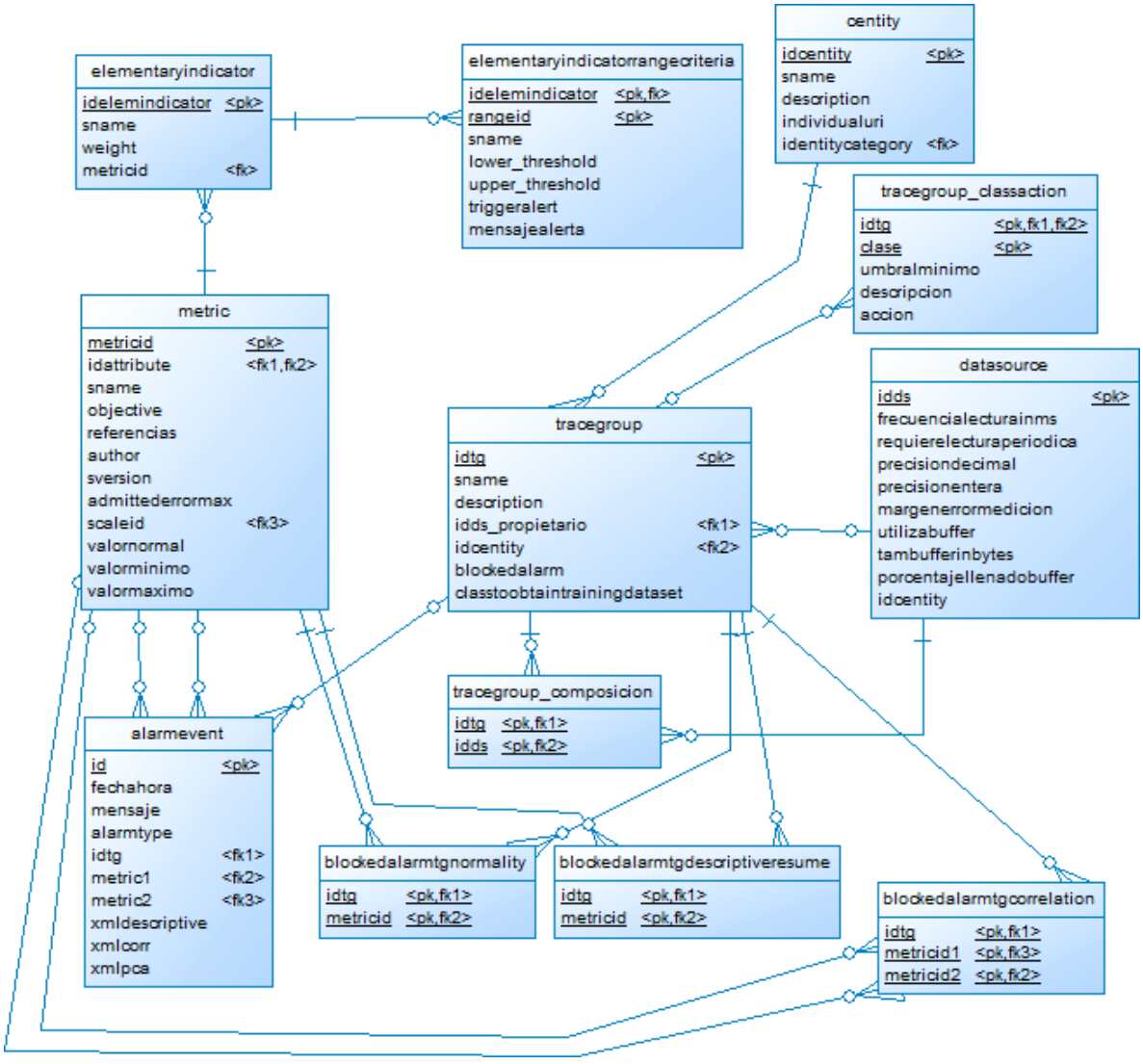


Figura 83. Vista Relacional del sub-esquema de la base de datos C-INCAMI asociado con la definición de indicadores y mecanismos de alarma

Al igual que las tablas analizadas en la sub-sección A.2, permiten definir un patrón de referencia para el análisis on-line de una medida y por ende, son de vital importancia en el análisis estadístico (ver capítulo 7), los conceptos aquí vertidos, representan un marco de referencia obligado para el proceso de toma de decisiones, ya que aquí, entre otras cuestiones, los expertos vierten en términos de indicadores, su conocimiento y experticia sobre el dominio del proyecto de M&E.

Como puede apreciarse en la Figura 83, una definición de métrica puede aprovisionar o nutrir a más de un indicador elemental (*elementaryindicator*). Cada indicador elemental tiene asociado diferentes rangos (*elementaryindicatorrangecriteria*), en donde cada uno establece sus umbrales, si corresponde o no emitir una alarma y el mensaje asociado a la misma. Dicha definición es de particular importancia para el proceso de toma de decisión planteado en el capítulo 8, por cuanto define un patrón de referencia para el DecisionMaker.

La base de datos mantiene relación entre cada fuente de datos y el grupo de seguimiento al que pertenecen (*tracegroup_composicion*) o bien del cual son propietarias (*tracegroup*). Cada grupo de seguimiento, en función de las etiquetas de clases definidas para el clasificador (rangos de interpretación nominales u ordinales de los indicadores), puede particularizar el tipo de acción a tomar (tracegroup_classaction) en la medida que satisfaga una probabilidad de ocurrencia mínima requerida. Esto permite potenciar los aspectos de particularización del clasificador al grupo de seguimiento, tal y como fue analizado en las sub-secciones 8.4 a 8.7.

Un aspecto de fundamental importancia en este sub-esquema de base de datos, es que permite configurar por cada grupo de seguimiento los tipos de alarmas que se prefieren no emitir, porque posiblemente las relaciones o situaciones que la provocan son conocidas a priori, y constituirían una sobrecarga innecesaria de procesamiento. En este sentido, es posible bloquear las alarmas por detección de normalidad en una métrica particular para un grupo de seguimiento (*blockedalarmtgnormality*), bloquear las alarmas surgidas del análisis descriptivo de una determinada métrica en un grupo de seguimiento (*blockedalarmtgdescriptiveresume*) y bloquear alarmas por detección de correlación entre dos métricas de un grupo de seguimiento (*blockedalarmtgcorrelation*).

Como forma básica de prueba de las alarmas emitidas, el modelo de datos contiene la posibilidad de desviar las alarmas hacia a un registro persistente (*alarmevent*). Ello es útil al momento de probar la configuración inicial del proyecto y verificar el régimen de alarmas emitidas, teniendo en cuenta que cuando el mismo esté en producción, las mismas serán emitidas a los responsables definidos en el proyecto. Por ejemplo, para el escenario planteado en el capítulo 9, serían los doctores del centro de salud.