Propuesta de Arquitectura Orientada a Calidad para PostgreSQLf

Livia Borjas, Carlos López, Adolfredo Malavé
livacaro7@gmail.com, caugusto2896@gmail.com, adolfredoml@gmail.com

Escuela de Ingeniería Informática, Universidad Católica Andrés Bello Ext. Guayana, Puerto Ordaz, Venezuela

Resumen: La presente investigación tiene como objetivo proponer una arquitectura para el sistema gestor de base de datos objeto relacional difuso (SGBDORD) PostgreSQLf con capacidades de análisis de datos difuso. Partiendo de una revisión documental se estudió el comportamiento y funcionamiento de los componentes que permiten la minería de datos difusa en dicho SGBDORD. También se utilizó como referencia metodológica para la propuesta un marco de trabajo y modelo híbrido-heurístico para diseño arquitectónico de SGBDORD con requisitos novedosos, proceso que hace énfasis en la consideración explícita de los atributos de calidad según las normas ISO 25010 durante el proceso de diseño. La validación de la propuesta se hizo mediante pruebas de escenarios y análisis de compensación de arquitectura (ATAM). Se obtuvo una arquitectura para PostgreSQLf orientada a la calidad, cuyo principal beneficio es favorecer el rendimiento y desempeño del manejador en la ejecución de requisitos novedosos de análisis difuso.

Palabras Clave: Arquitectura; Calidad; ISO 25010; PostgreSQLf.

1. Introducción

El proyecto de transferencia tecnológica PostgreSQLf [1] propone la innovación del sistema gestor de bases de datos objeto relacional (SGBDOR) PostgreSQL [2], incorporándole capacidades de consultas con datos imperfectos y de análisis de datos, mediante lógica difusa, abordando la problemática de rigidez en la representación, gestión y explotación de los datos presentes en los sistemas gestores de bases de datos relacionales (SGBDR). El presente artículo tiene como objetivo exponer una arquitectura orientada a calidad para el sistema gestor de bases de datos objeto relacional difuso (SGBDORD) PostgreSQLf cuyo diseño favorezca atributos de calidad como el desempeño del manejador en términos de tiempo de respuesta durante las consultas de análisis difuso. El artículo está conformado por cinco secciones: la Sección 2 expone los antecedentes, la Sección 3 expone la metodología aplicada, en la Sección 4 se presentan los resultados y en la Sección 5 las conclusiones y recomendaciones.

2. Antecedentes

Con el propósito de comprender la arquitectura actual del SGBD PostgreSQLf, durante esta etapa se realizó una revisión documental en los trabajos que antecedieron, entre los cuales se destacan las publicaciones realizadas por Borjas et al. [1][2], Timarán [3][4], Bazán [5] y Sabatino [6].

2.1 SQLf

El lenguaje de consultas difusas SQLf [1], es una extensión de SQL con lógica difusa [7] que permite realizar consultas flexibles sobre Bases de Datos Relacionales. Los términos difusos son incorporados en el sublenguaje de definición de datos (DDL) y el sublenguaje de manipulación de datos (DML). Estos términos incluyen expresiones lingüísticas tales como: predicados, modificadores, comparadores, conectores y cuantificadores. De esta forma, las condiciones de las consultas pueden involucrar términos difusos definidos por el usuario, operadores y/o subconsultas difusas.

2.2 Extensión de PostgreSQL con Minería de Datos

Timarán [3] presenta una extensión de SQL con primitivas de minería de datos en una arquitectura fuertemente acoplada sobre PostgreSQL, demostrando que la deficiencia de rendimiento y escalabilidad mejora al usar mecanismos de integración en arquitecturas fuertemente acopladas. Para ello, los algoritmos de extracción de conocimiento forman parte del núcleo del SGBD como operaciones primitivas, por lo que son ejecutados en el mismo espacio de direccionamiento que los datos, resultando en un mejor desempeño en costo. Este antecedente demuestra la factibilidad de implementar estos operadores [4].

2.3 PostgreSQLf

Se trata de una extensión de SQLf implementada fuertemente acoplada al SGBDOR PostgreSQL. Esta extensión se conoce como PostgreSQLf [2][5][6], que a través de una arquitectura fuertemente acoplada y un patrón arquitectónico cliente-servidor estructurado por capas de software, ofrece capacidades de análisis de datos difuso, la cual determina que la lógica de desarrollo sea implementada en el lenguaje nativo del SGBD y se coloque dentro del kernel. Entre las capacidades que se han incorporado están: predicados difusos definidos por trapecio, expresión y extensión; operadores difusos AND y OR, modificadores difusos, consultas particionadas difusas en las cláusulas GROUP BY y HAVING; vistas y operación de unión difusas; creación de cuantificadores difusos, consultas cuantificadas difusas de tipo bloque simple; comparadores y conjunciones difusas; subconsultas difusas en la cláusula FROM; bloques anidados Exists-Like y Any-Like, bloque particionado, entre otras características difusas.

3. Metodología

Partiendo del refinamiento de metodologías usadas en el proyecto [1][2][8][9] se propone un marco de trabajo y modelo híbrido-heurístico para diseño arquitectónico de sistema gestor de base de datos objeto relacional difuso (SGBDORD) con requisitos novedosos.

Esta propuesta metodológica incorpora las fases de:

3.1 Fase de Planificación

Esta fase engloba el cálculo de los riesgos que pueden surgir en las distintas fases del proyecto y la estimación de los costos necesarios para llevarlo a cabo, además, en base a las actividades mencionadas se establecen las etapas para el desarrollo del proyecto [10].

3.2 Fase de Especificación de Requerimientos

Es la etapa referente a la definición de las funcionalidades que serán agregadas en la extensión del SGBDOR, centrándose a su vez, en que las contemple rigiéndose bajo los estándares de calidad.

3.3 Fase de Diseño de la Arquitectura

Es la etapa que enmarca la estructura del software a desarrollar, los datos utilizados en ella, los componentes que la integrarán, su representación (mediante diagramas UML 2.0) y características a partir de los requerimientos previamente obtenidos. Se utilizan herramientas como los patrones y estilos arquitectónicos.

3.4 Fase de Validación del Diseño Arquitectónico

Es la fase que tiene como finalidad validar el diseño arquitectónico a través de pruebas y evaluaciones, pudiendo resultar en la aceptación de la arquitectura o denotar la necesidad de cambios en el diseño.

4. Resultados

4.1 Arquitectura de PostgreSQLf

Una vez revisada la planificación y especificados los requerimientos a través de entrevistas y juicio de expertos resultó:

a) Requerimientos Novedosos: El usuario aplicará el proceso de extracción de conocimiento desde el SGBDORD realizando consultas con capacidad de minería de datos difusa soportando tareas descriptivas y predictivas de minería de datos difusa, facilitando almacenar datos bajo el modelo de conjuntos difusos. También se espera que el sistema proveerá un ambiente gráfico inteligente que recomiende la toma de decisiones durante el proceso de extracción de conocimiento. Entre los requisitos no funcionales destacan: Corrección Funcional, Completitud Funcional, Pertinencia Funcional, Utilización de recursos, Estética de la interfaz de usuario, Capacidad de aprendizaje, Capacidad para ser usado, Reutilización, Capacidad para ser probado, Modularidad, Capacidad para ser modificado

b) Arquitectura de Acoplamiento: Se asume como enfoque de integración una arquitectura fuertemente acoplada, la cual, a pesar de que aumenta la complejidad y sacrifica la portabilidad, cumple el requisito no funcional de rendimiento y permite añadir las capacidades de requisitos novedosos al SGBDORD.

c) Modelos de Casos de Uso para Requisitos Novedosos: Para la definición de los casos de uso de la arquitectura se utilizaron como referencia los pasos necesarios para la extracción de conocimiento, tomando en cuenta los requisitos funcionales novedosos levantados previamente, modelando las acciones que puede realizar un usuario en el sistema (véase Figura 1).

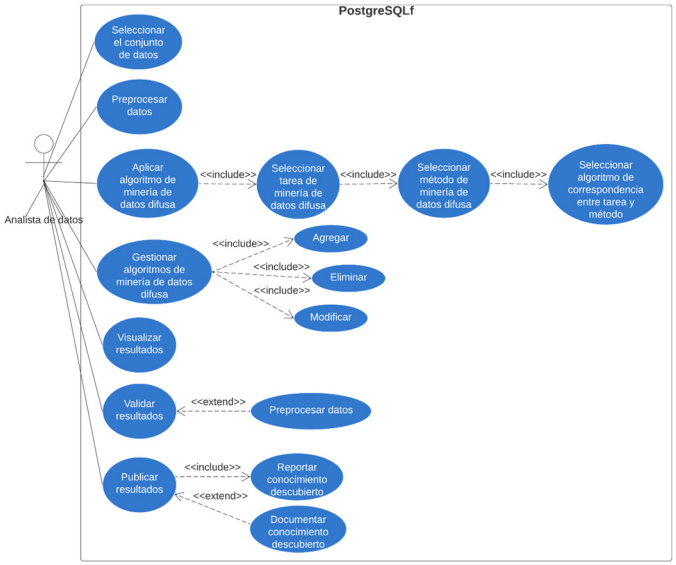

Figura 1: Diagrama de Caso de Uso General de PostgreSQLf

d) Modelo de Calidad para Requisitos Novedosos: Se procedió a generar un árbol de utilidad de atributos de calidad de ATAM [8] (véase Figura 3), ilustrando las propiedades esperadas de un sistema construido con la calidad como principal objetivo. El diagrama inicia con un nodo raíz que simboliza la cualidad global en estudio, siendo en este caso particular PostgreSQLf. A partir del nodo raíz, surgen aquellos nodos que reflejan las características de calidad de la norma ISO 25010 [11], planteadas como punto inicial de la arquitectura, de estos surgen nuevos nodos que reflejan las subcaracteristicas deseadas e idóneas seleccionadas como requerimientos de arquitectura (no funcionales), basadas en el criterio de los investigadores. Dichos nodos, pueden o no, resultar en una o más hojas que representan propiedades de calidad que se esperan del sistema, aplicadas al caso de estudio. Por último, dichas propiedades de calidad resultantes son priorizadas como alta (A), media (M) o baja (B)

4.2 Operadores Algebraicos para Requisitos Novedosos

Para Reglas de Asociación [12] y Clasificación [13], se tomaron como base los operadores algebraicos de asociación diseñados por Timarán [3][4] modificándolos para agregar condiciones difusas:

- Associator (α): tomando los valores de los atributos en una tupla, genera todas sus posibles combinaciones.

- Assorow (αρ): por cada tupla de una relación se generan todos los posibles subconjuntos (itemsets) de los valores de sus atributos.
- Assocol (ακ): utilizada para relaciones de modelo simple para el descubrimiento de reglas de asociación.
- Equikeep (λ): los valores de los atributos de una tupla de una relación son restringidos a los que satisfagan una condición específica.
- Mate (μ): por cada una de las tuplas de una relación se generan todas las posibles combinaciones formadas por los valores no nulos de sus atributos.
- Describe Classifier (βμ): produce una nueva relación donde se almacenan los valores de los atributos que formarán los diferentes nodos del árbol de decisión.

En el caso de agrupamiento [14], se están diseñando dos operadores algebraicos de agrupamiento difusos:

- Centers (θt): tiene como resultado una nueva relación con el centro de los grupos, los cuales sirven como soporte.
- Membership (μθ): genera una relación donde por cada elemento a ubicar existe un valor de pertenencia a cada centro de los clusters.

4.3 Primitivas de SQLf para Operaciones Novedosas

Primitivas de SQLf para reglas de asociación difusa [12]. Para este fin se requirió: un operador capaz de generar reglas de asociación difusa mediante: Una tabla de la base de datos (`NombreTabla`), tamaño específico de la tabla (`TamañoRegla`), soporte mínimo y confianza mínima. Cuya sintaxis:

```
DESCRIBE_FUZZY_ASSOCIATION_RULES(NombreTabla,
TamañoRegla, Soporte, Confianza);
```

Además, se adaptó la sintaxis del bloque simple para permitir consultas con condiciones que contemplen la minería difusa.

```
SELECT label(A) [agregación]
FROM NombreTabla
WHERE ψ GROUP BY label(A)
USING PARTITION p(A) = {L1…Ln};
```

Donde p(A) es una partición difusa definida en el dominio de A, label(A) corresponde a las posibles etiquetas lingüísticas Li de p(A) y ψ es una condición difusa.

Primitivas de SQLf para la clasificación difusa [13]. Se adaptó la sintaxis del bloque simple, la cual dio como resultado la sintaxis siguiente:

```
DESCRIBE_FUZZY_CLASSIFICATION_RULES
[INTO <TablaReglasClasificacionDifusa] FROM
<NombreTablaArbolDatos>
USING <NombreTablaMetricaDifusa>
[DO <SubconsultaCalculoMetrica>]
<SubconsultaCalculoMetrica>::=<SFWMG>
<SFWMG>:: = <SELECT FROM WHERE MATE BY GROUP BY>
```

Dónde: `<NombreTablaArbol>` tabla de nodos del árbol de datos, `<NombreTablaMetrica>` etiquetas lingüísticas, valores de los atributos y métricas utilizadas y `[DO <SubconsultaCalculoMetrica>]` cláusula opcional para calcular Entropía y Ganancia de Información.

Primitivas de SQL para el agrupamiento difuso [14] con sintaxis del bloque simple, resultando:

```
SELECT<ListaAtributosTablaDatos> [INTO<NombreTablaClustering>]
FROM <NombreTablaDatos> WHERE<Condicion>
CLUSTERING <Valor1>, <Valor2>, <Valor3>
GROUP BY <Label (A)>
<Valor1>::=1,2,3,4…
<Valor2>::=1,2,3,4…
<Valor3>::=1,2,3,4…
```

Donde `CLUSTERING` refleja los parámetros de agrupamiento; el `Valor1` indica el número de grupos que se va a buscar, `Valor2` el difusor de los grupos, el `Valor3` el criterio de parada y `Label` (A) corresponde a las posibles etiquetas lingüísticas Li de p(A).

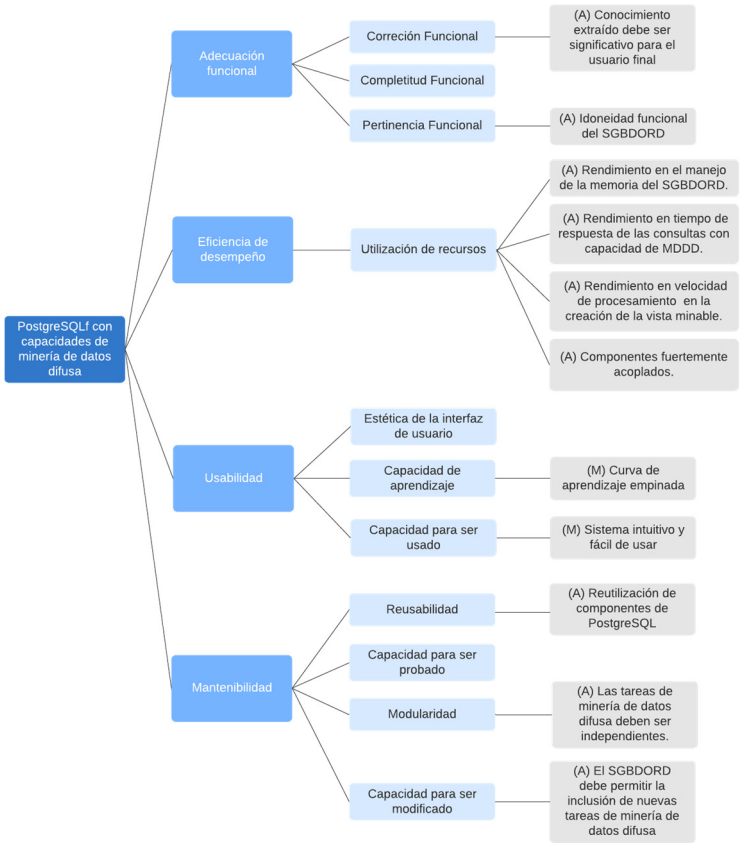

Figura 2: Modelo de Calidad para PostgreSQLf

4.4 Derivación Arquitectónica y Aplicación de la Taxonomía CBSP

Clasificando las propiedades de calidad plasmadas en las hojas del árbol de atributos de calidad (véase Figura 2), resultó la arquitectura base de PostgreSQLf con capacidad de análisis difuso de datos, como se aprecia en la Figura 3, utilizando UML 2.0.

4.5 Validación de la Arquitectura

Se utilizó el método ATAM [15], contando con la opinión de destacados expertos investigadores del tema participantes del Proyecto PostgreSQLf que apoyaron con su juicio de experto, destacando, las Fortalezas: la inclusión de los predicados

difusos en el manejador del catálogo da la ventaja que sean utilizados globalmente en cualquier tarea de minería de datos difusa. Buen rendimiento en las consultas de tareas de minería difusa. El componente de la interfaz gráfica inteligente es vital, facilitando su uso y aprendizaje lo cual minimiza la posibilidad de errores ocasionados por el usuario. Entre los aspectos sensibles destaca que realizar cambios significativos sobre los componentes de tareas de minería de datos es complejo y costoso. Con la finalidad de mitigar riesgos se recomienda desarrollar los operadores algebraicos en nuevas versiones del SGBDOR. Una vez analizados los resultados obtenidos en la evaluación de la arquitectura, se puede observar que las fortalezas plasman los estándares de calidad más relevantes para la arquitectura inicial de PostgreSQLf: rendimiento en términos de tiempo de respuesta.

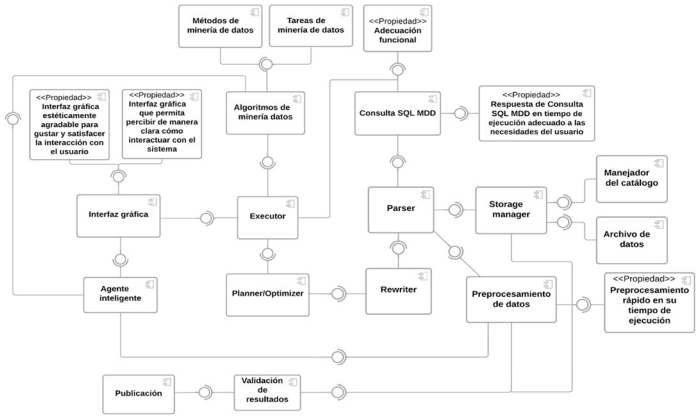

Figura 3: Diagrama UML 2.0 de la Arquitectura de PostgreSQLf

5. Conclusiones

Se diseñó una arquitectura para PostgreSQLf con capacidades de análisis de datos difuso que representa una innovación valiosa del SGBDORD. La arquitectura orientada a la calidad con acopamiento fuerte, siendo los componentes críticos que permiten la minería de datos difusa en PostgreSQLf, los operadores algebraicos que extienden el álgebra relacional junto a sus primitivas. La validación de la arquitectura diseñada denota que su implementación es factible. Adicionalmente a la arquitectura propuesta, se validó la metodología usada.

Agradecimientos

"A ti toda la Gloria", Señor del Cielo, autor de todo conocimiento e inspirador de toda ciencia.

Referencias

[1] A. Aguilera., L. Borjas, C. Rodríguez, *Un Modelo de Proceso Heurístico para Implementación de SGDB Difusos*. Tercera Conferencia Nacional de Computación, Informática y Sistemas / CoNCISa 2015 / ISBN: 978-980-7683- 01-2, 2015.

[2] A. Aguilera, L. Borjas, C. Rodríguez y F. Losavio, Arquitectura *para PostgreSQLf con Facilidades de Minería de Datos Descriptiva Difusa.* Revista Venezolana de Computación, vol. 3, no. 2, 2016.

[3] R. Timarán, *Nuevas Primitivas SQL para el Descubrimiento de Conocimiento en Arquitecturas Fuertemente Acopladas con un Sistema de Gestión de Bases de Datos* (Tesis Doctoral). Universidad del Valle, Santiago de Cali, Colombia, 2005.

[4] R. Timarán, *Arquitectura del SGBD PostgreSQL*. Universidad de Nariño, San Juan de Pasto, Colombia.

[5] G. Bazán, *PostgreSQLf: Implementación de Cuantificadores Difusos Sobre el SGBD PostgreSQL* (Tesis de pregrado). Universidad de Carabobo, Venezuela, 2009.

[6] R. Sabatino, *PostgreSQLf: Extensiones de Cuantificadores Difusos y Agrupamiento Difuso sobre el SGBD PostgreSQL* (Tesis de Pregrado). Universidad de Carabobo, Venezuela, 2013.

[7] L. A. Zadeh, *The Concept of a Linguistic Variable and its Application to Approximate Reasoning*—I. Information Sciences, vol. 8, no. 3, pp. 199-249. 1975.

[8] K. Álvarez, B. Romero, J. Cadenas, D. Coronado y C. Rodríguez, *Arquitectura para la Gestión de Datos Imperfectos en la Era de Big Data*. Revista Venezolana de Computación, vol. 3, no. 2, 2016.

[9] F. Losavio y F. Guillen, *Marco Conceptual para un Diseño Arquitectónico Basado en Aspectos de Calidad*. Revista Universitaria de Investigación, vol. 7, no. 2, pp. 119-138. 2006.

[10] D. Suárez, Modelos *de Estimación para Proyectos de Software. Serie Científica de la Universidad De Las Ciencias Informáticas*, vol. 1, no. 6, 2019.

[11] Organización Internacional de Normalización. (s.f.). *System and Software Quality Requirements and Evaluation (SQuaRE) Series of Standards (ISO 25000)*. https://iso25000.com.

[12] A. García y R. Guevara, *Extensión de SQLf con Primitivas para Soportar Análisis de Datos con Reglas de Asociación Difusas Fuertemente Acopladas al SGBDORD PostgreSQLf.* Trabajo Especial de Grado (Ingeniero en Informática)-Universidad Católica Andrés Bello, Facultad de Ingeniería, Especialidad (Ingeniería Informática). Ciudad Guayana, 2021.

[13] R. Carrasquero y F. González, *Extensión de SQLf para Soportar Análisis de Clasificación Difusa Fuertemente Acoplado a PostgreSQLf.* Trabajo Especial de Grado (Ingeniero en Informática) - Universidad Católica Andrés Bello, Facultad de Ingeniería, Especialidad (Ingeniería Informática). Ciudad Guayana, 2021.

[14] C. Zabala, *Primitivas de Agrupamiento Difuso Fuertemente Acopladas al SGBDORD POSTGRESQLF*. Trabajo Especial de Grado (Ingeniero en Informática) - Universidad Católica Andrés Bello, Facultad de Ingeniería, Especialidad (Ingeniería Informática). Ciudad Guayana, 2021.

[15] A. Delgado, A. Castro y G. Martín, *Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): Un Caso de Estudio*. Reportes Técnicos 07, 2007.