UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

Herramienta de Visualización de Resultados de Pruebas Estadísticas sobre Sistemas Manejadores de Base de Datos Analíticos.

TESIS
QUE PARA OPTAR POR EL GRADO DE:
Maestro en Ciencia e Ingeniería de la Computación

PRESENTA:
Trujillo Torres Octavio

DIRECTOR DE TESIS:
Dr. Javier García García
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

CIUDAD UNIVERSITARIA, CD. DE MÉXICO. ENERO 2017

ÍNDICE GENERAL

ÍNDICE GENERAL III

ÍNDICE DE TABLAS VII

ÍNDICE DE FIGURAS IX

1. Introducción **1**

1.1. Objetivo 2

1.2. Aportaciones 3

1.3. Trabajos Relacionados 3

1.3.1. Primer Antecedente 4

1.3.2. Segundo Antecedente 4

1.3.3. Tercer Antecedente 5

2. Marco Teórico **7**

2.1. OLAP 7

2.1.1. Cubo OLAP 9

2.1.2. Operaciones Sobre Cubos 10

2.1.3. Retícula 11

2.2. Pruebas de Hipótesis 13

2.3. Ergonomía 18

2.4. Usabilidad 19

3. Almacenes de Datos y Sistemas Manejadores de Bases de Datos Analíticos **21**

3.1. Almacenes de Datos 21

3.2. Sistemas Manejadores de Bases de Datos Columnares 23

3.2.1. Compresión de Datos . . . 24
3.2.2. HP Vertica . . . 25
3.3. Sistemas Manejadores de Base de Datos por Arreglos . . . 28
3.3.1. Arreglos . . . 28
3.3.2. Chunks . . . 29
3.3.3. Diferencia entre Arreglos y Tablas . . . 31
3.4. SciDB . . . 31
3.4.1. Arquitectura . . . 32
3.4.2. Modelo de Datos . . . 33
3.4.3. Lenguajes de Manipulación de Datos . . . 35

4. Pruebas de Usuario **37**
4.1. Breve descripción del Desarrollo Centrado en el Usuario (DCU) . . . 37
4.2. Evaluación de Interfaces sin Usuarios . . . 39
4.2.1. Pruebas de Usabilidad . . . 39
4.2.2. Criterios Ergonómicos . . . 39
4.3. Evaluación del Sistema . . . 42
4.4. Evaluación de Interfaces con Usuarios . . . 51
4.4.1. Laboratorio . . . 51
4.4.2. Etapas de la Prueba . . . 53

5. Herramienta OLAP **57**
5.1. Consultas AQL para la Generación de Cubos . . . 58
5.1.1. Arreglo base . . . 59
5.1.2. Media de una sola población . . . 60
5.1.3. Diferencia entre medias de dos poblaciones . . . 61
5.1.4. Proporción de una población . . . 64
5.1.5. Diferencia entre las proporciones de dos poblaciones . . . 66
5.1.6. Varianza de una sola población . . . 69
5.1.7. Razón de las varianzas de dos poblaciones . . . 70
5.1.8. Chi-cuadrada (prueba de independencia) . . . 73
5.2. Interfaz Propuesta . . . 74

6. Experimentación y Pruebas con Usuarios **81**
6.1. Comparación de Tiempos de Ejecución de Pruebas Estadísticas . . . 81

6.1.1. Definición Tabla y Arreglo 82
6.1.2. Tiempos 83
6.2. Pruebas con Usuarios 86
6.2.1. Perfil del Usuario 86
6.2.2. Hipótesis 87
6.2.3. Evaluación Subjetiva 92
6.2.4. Propuesta de mejora de interfaz 94

7. Conclusiones y Trabajo Futuro **97**

BIBLIOGRAFÍA **99**

A. Documentos Plan de Pruebas **103**

ÍNDICE DE TABLAS

2.1. Condiciones en las que es posible cometer un error de tipo I y un error tipo II, tomado de [10] 15

4.1. Tabla de análisis de la interfaz de selección de DBMS. 44
4.2. Tabla de análisis de la interfaz de selección de tabla. 45
4.3. Tabla de análisis de la interfaz de selección de dimensiones y medidas. 46
4.4. Tabla de análisis de la interfaz de selección de parámetros. 48
4.5. Tabla de análisis de la interfaz de visualización del cubo OLAP. . . . 49
4.6. Tabla de análisis de la interfaz de visualización en forma de lattice. . 50

6.1. Tabla comparativa de tiempos de respuesta (segundos). 84
6.2. Análisis de Actividades Usuario 1 88
6.3. Análisis de Actividades Usuario 2 89
6.4. Análisis de Actividades Usuario 3 90
6.5. Análisis de Actividades Usuario 4 91

ÍNDICE DE FIGURAS

1.1. Visualización de todos los cuboides. Imagen de [25]. 5
1.2. Visualización de pares de cuboides significantes. Imagen de [25]. . . . 5
1.3. Detalle del cuboide seleccionado. Imagen de [13]. 6
1.4. Ejemplo de grupos similares que difieren en una dimensión. Imagen de [13]. 6

2.1. Ejemplo de un cubo OLAP, tomada de [14]. 10
2.2. Ejemplo de un lattice, tomada de [14] 11
2.3. Estructura del Triangulo de Pascal, tomada de [7]. 12
2.4. Número de cuboides por nivel 13

3.1. Almacenamiento columnar, tomado de [1] 24
3.2. Modelo de Almacenamiento en *Vertica*, tomado de [26] 27
3.3. Modelo de Almacenamiento en *Vertica*, tomado de [23] 27
3.4. Estructura de un arreglo, tomado de [6]. 30
3.5. Particionamiento del arreglo por atributos, tomado de [6]. 30
3.6. Descomposición cada atributo en porciones de igual tamaño (chunks), tomado de [6]. 30
3.7. Arquitectura *SciDB*, tomado de [28] 33
3.8. Representación de un arreglo de 2 dimensiones, tomado de [22] 34
3.9. Definición de un arreglo de 2 dimensiones con 3 atributos, tomado de [22] 35

4.1. Interfaz de selección de DBMS. 43
4.2. Interfaz de selección de tabla. 44
4.3. Interfaz de selección de dimensiones y medidas. 45
4.4. Interfaz de selección de parámetros para la prueba. 47

4.5. Interfaz de visualización del cubo OLAP. 49
4.6. Interfaz de visualización en forma de retícula. 50
4.7. Ejemplo del laboratorio de pruebas con usuario, tomado de [30]. . . . 52

5.1. Interfaz de inicio. 75
5.2. Interfaz de inicio, agrupación de elementos. 76
5.3. Interfaz de inicio con mensaje de error de conexión al sistema manejador de base de datos. 77
5.4. Interfaz de inicio con mensajes de error por parámetros faltantes o incorrectos. 77
5.5. Interfaz de configuración de parámetros de conexión con el sistema manejador de base de datos. 78
5.6. Interfaz de visualización de resultados, sección cubo OLAP. 79
5.7. Interfaz de visualización de resultados, sección retícula multidimensional. 79

6.1. Atributos relevantes de la tabla tbbm. 83
6.2. Atributos relevantes del arreglo tbbm 83
6.3. Visualización de resultados en forma de cubo OLAP. 85
6.4. Visualización de resultados en forma de retícula multidimensional. . . 85
6.5. Respuestas de los usuarios a la pregunta 1 del cuestionario de evaluación subjetiva. 92
6.6. Respuestas de los usuarios a la pregunta 4 del cuestionario de evaluación subjetiva. 93
6.7. Respuestas de los usuarios a la pregunta 9 del cuestionario de evaluación subjetiva. 93
6.8. Interfaz de inicio. 94
6.9. Interfaz de visualización de resultados. 95

CAPÍTULO 1

Introducción

Hoy en día existe una gran necesidad por contar con herramientas y estrategias que permitan el análisis de grandes volúmenes de datos que son generados tanto por las empresas como en el ámbito de la investigación científica. Estos grandes volúmenes se encuentran principalmente en almacenes de datos los cuales se tiene que analizar y explorar para fines que van desde la presentación de informes para el análisis de estrategias de negocio, hasta la ejecución de algoritmos de minería de datos, en este sentido las herramientas de tipo OLAP nos facilitan este trabajo. Dentro de la gran variedad de herramientas OLAP, existen los sistemas basados en columnas los cuales presentan grandes ventajas en cuanto a tiempo de respuesta y espacio de almacenamiento ya que suelen usar algoritmos de compresión de datos, sin embargo, existe un renovado interés en la comunidad científica por los sistemas basados en arreglos ya que presentan ciertas ventajas dada la naturaleza de los datos y el procesamiento que se requiere para poder obtener algún resultado.

Dentro del análisis de datos existen muchas técnicas y algoritmos que nos ayudan a obtener alguna conclusión sobre un conjunto de datos, siendo las pruebas estadísticas parte fundamental de este proceso y dentro de éstas, las pruebas de hipótesis resultan especialmente útiles cuando se quiere validar una hipótesis sobre un conjunto de datos [24].

Por otro lado, existe un cambio en la forma en la que se desarrollan las aplicaciones. Anteriormente se desarrollaban teniendo en mente lo que las computadoras podían hacer, ahora se deben desarrollar teniendo en mente lo que el usuario puede

hacer, cómo lo menciona Ben Shneiderman [32], siendo exitosas aquellas herramientas que estén en armonía con necesidades de los usuarios que las van a usar.

Teniendo esto en mente, en el presente trabajo se contruyó una herramienta de análisis y visualización de resultados de pruebas estadísticas para ambientes analíticos ya sea basados en tablas o en arreglos. La herramienta permite flexibilidad sobre el modelo de datos que se prefiera utilizar y esto en forma transparente para el usuario durante el proceso de realización de ejecución de las pruebas. La aplicación desarrollada está implementada en JAVA por lo que adquiere todas las ventajas inherentes al lenguaje y utiliza estándares de SQL para su adaptabilidad a los sistemas existentes dentro de una empresa u organización. En su desarrollo se tomó en cuenta las necesidades del usuario y se siguieron criterios ergonómicos que mejoran la utilización de la misma, todo esto como una aportación importante a la herramienta base[1].

1.1. Objetivo

El objetivo de la presente investigación es proveer de una herramienta de visualización de cubos multidimensionales que pueda ser utilizada en ambientes analíticos del tipo OLAP mediante la implementación de pruebas estadísticas sobre sistemas manejadores de bases de datos (DBMS) analíticas y científicas tomando como base el prototipo presentado en [13], además de proporcionar una herramienta que no presente problemas en la percepción por parte del usuario, problemas de aprendizaje o dificulte la realización de las tareas del usuario.

Con los objetivos anteriores se busca tener una herramienta que podrá ser aplicada en distintos campos y contextos. La investigación realizada derivó en la incorporación de elementos en la herramienta cuyo objetivo fue evitar problemas para el usuario en el uso de ésta. El enfoque de su desarrollo se centró en la tarea a realizar y en la interpretación de los resultados obtenidos lo que mejoró en gran medida la usabilidad de la misma.

[1]Herramienta que utiliza estadística y OLAP para obtener cuboides que difieren en una dimensión sobre manejadores de base de datos relacionales tradicionales. Presenta la información en forma de cubos multidimensionales y retículas presentada en [13].

En cuanto a la parte tecnológica de análisis científico de la herramienta se contempló implementarla en sistemas manejadores de base de datos columnares como *HP Vertica*, científicos con estructuras de arreglos como *SciDB*. Finalmente se probó el desempeño de la herramienta y se corroboró su eficiencia en la tarea estadística contemplada.

1.2. Aportaciones

En particular se aporta lo siguiente al campo de las herramientas de análisis de datos tipo OLAP.

- La implementación de pruebas estadísticas definidas en [12] sobre sistemas manejadores de base de datos analíticas y científicas como lo son *HP Vertica* y *SciDB* respectivamente.
- Mejora de la interfaz de usuario siguiendo criterios ergonómicos establecidos en [4] para mejorar el uso y manejo de la herramienta teniendo siempre en mente al usuario.
- Comparación de resultados obtenidos de la ejecución de las diferentes pruebas estadísticas sobre sistemas manejadores de bases de datos científicos basados en arreglos (*SciDB*) y analíticos basados en tablas (*HP Vertica*) contra los obtenidos sobre sistemas manejadores de base de datos relacionales (*PostgreSQL*).

1.3. Trabajos Relacionados

En el campo de visualización de cubos OLAP diversas herramientas proponen alternativas de presentación. Algunas de ellas se enfocan en diferentes técnicas de visualización mediante la representación de gráficas, histogramas o cubos 3D. Se muestran aquellas que son más cercanas al tema de estudio.

1.3.1. Primer Antecedente

En [24] se propone una técnica que combina retículas (del inglés lattice) y pruebas estadísticas para descubrir diferencias significativas en el grado de enfermedad de grupos de pacientes y para entender una causa-efecto, se centra en el grupo de pacientes que difieren en una dimensión. Esta técnica tiene por objetivo la integración de pruebas estadísticas y técnicas OLAP para la formulación y validación de hipótesis médicas en múltiples subconjuntos de un conjunto de datos.

Con el algoritmo propuesto en este artículo se realizan las siguientes tareas: generación dinámica de consultas SQL para encontrar el cubo n-dimensional, generación de la retícula y ejecución de una prueba estadística. Se parte de la hipótesis de que un factor de riesgo específico combinado con otros factores de riesgo que pueden llevar a una alta probabilidad de desarrollar la enfermedad.

1.3.2. Segundo Antecedente

En [25], el artículo propone un algoritmo que automatiza el proceso de exploración y análisis de un conjunto de datos, combinando cubos OLAP con pruebas estadísticas. Dicho algoritmo calcula automáticamente todos los cuboides de un cubo multidimensional mientras se aplican pruebas estadísticas para descubrir diferencias significativas en atributos medida de un cubo.

Se usa la prueba estadística para comparar la media de pares de grupos de poblaciones. Se asume que las medidas del cubo tienen una distribución normal. Objetivos para la aplicación de pruebas estadísticas:

- Descubrir diferencias significativas entre los dos grupos de un cuboide o por lo menos en una medida.
- Cuando exista una diferencia significativa, separar los grupos que difieren en una dimensión.

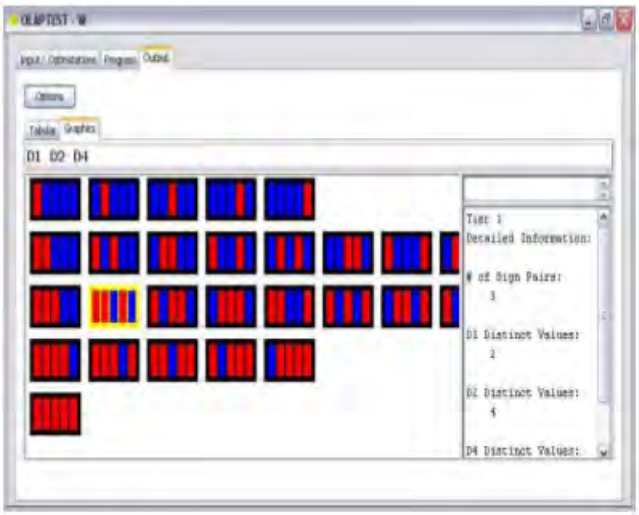

Figura 1.1: Visualización de todos los cuboides. Imagen de [25].

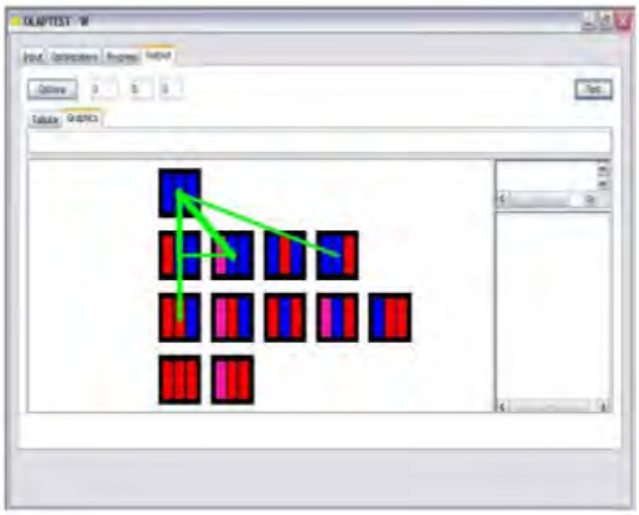

Figura 1.2: Visualización de pares de cuboides significantes. Imagen de [25].

En la Figura 1.1 se puede apreciar la visualización de todos los cuboides involucrados y en la Figura 1.2 se resaltan aquellos pares de cuboides que son significantes.

1.3.3. Tercer Antecedente

La principal aportación de la tesis [13] fue la implementación y representación de cubos n-dimensionales en forma de retícula, como una mejora a la herramienta presentada en [12], con esta mejora es posible identificar los cuboides integran un nodo en la retícula con tan solo seleccionarlo, mostrando el nivel de agregación en

el que se encuentra y cuales son los valores que lo componen, Figura 1.3.

En otro panel se muestra la cantidad de grupos altamente similares que difieren en una dimensión (cuadros en color azul), esto como resultado de la aplicación de la prueba estadística, Figura 1.4.

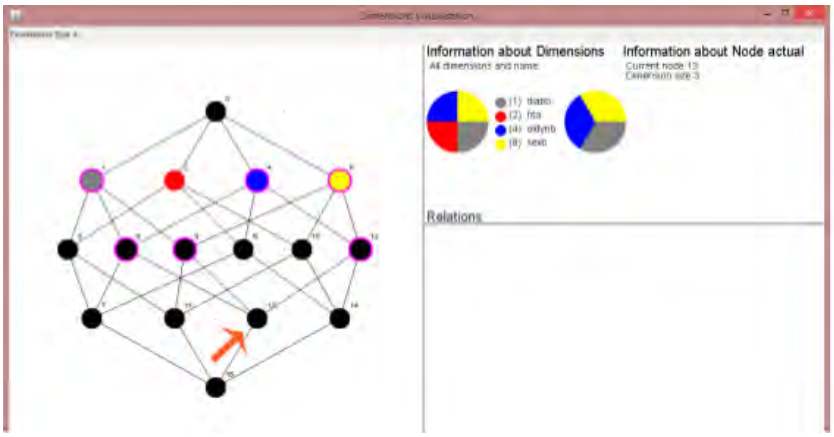

Figura 1.3: Detalle del cuboide seleccionado. Imagen de [13].

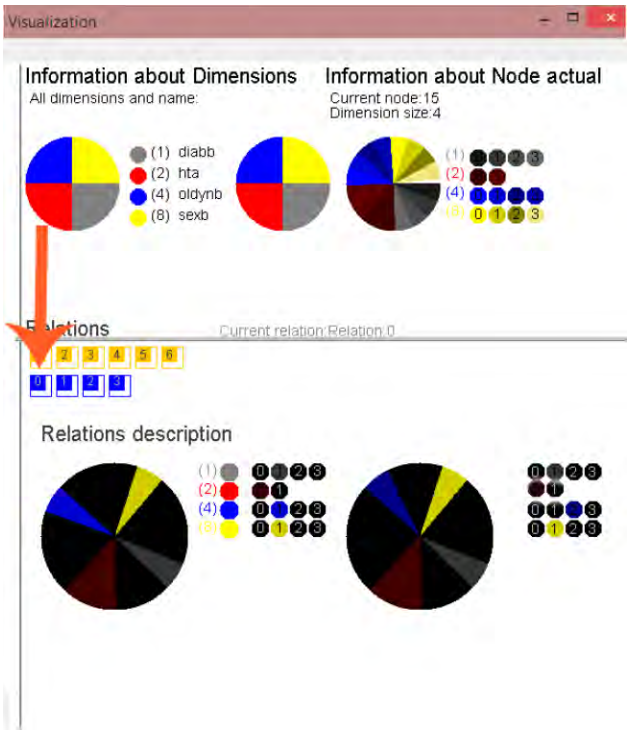

Figura 1.4: Ejemplo de grupos similares que difieren en una dimensión. Imagen de [13].

CAPÍTULO 2

Marco Teórico

En el presente capítulo se describen algunos conceptos referentes a la representación multidimensional de los datos y la descripción de algunos conceptos importantes en el área del desarrollo de aplicaciones centradas en el usuario. Esto con la finalidad de tener un mejor entendimiento del funcionamiento de la herramienta base así como la mejora a la misma.

2.1. OLAP

El término OLAP (por sus siglas en inglés, Online Analytical Processing) hace referencia a sistemas orientados al procesamiento analítico de datos y de toma de decisiones. Este análisis suele implicar, generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil, como podría ser: tendencias de ventas, patrones de comportamiento de los consumidores, elaboración de informes complejos, etc [33]. En estos sistemas el acceso a los datos suele ser de solo lectura ya que la principal operación es la consulta, con muy pocas inserciones, actualizaciones o eliminaciones y en donde los datos se estructuran según las áreas de negocio. Los sistemas gestores de bases de datos tipo OLAP se suelen alimentar de información procedente de los sistemas operacionales existentes, mediante un proceso de extracción, transformación y carga (ETL). Aunque el término OLAP fue descrito por primera vez por E.F. Codd en 1993 con ideas provenientes de los años 80's, aun no existe una definición formal de este concepto. Como lo propuso Nigel Pendse, uno de los analistas más importantes de OLAP y Business Intelligence, en [29], una herramienta para que sea considerada como herramienta OLAP debe pa-

sar la prueba FASMI (Fast Analysis of Shared Multidimensional Information, por sus siglas en inglés). Por lo tanto, estas herramientas deben ser lo suficientemente eficientes para permitir consultas interactivas; deben ayudar a la tarea de análisis proporcionando flexibilidad en el uso de herramientas estadísticas; deben proporcionar mecanismos de seguridad (tanto en el sentido de confidencialidad como en el sentido de integridad) que permitan el intercambio de datos; deben proporcionar una visión multidimensional de manera que los cubos de datos puedan ser usados por los usuarios; por último, deben ser capaces de manejar grandes volúmenes de datos. Sin embargo, no hay medidas ni umbrales para todas las características por lo que se acordó que para determinar si una herramienta es OLAP o no, esta debe ofrecer una visión multidimensional de los datos [2]. Originalmente OLAP implica sistemas manejadores de bases de datos (DBMS) multidimensionales sin embargo existen diferentes arquitecturas para los sistemas OLAP, estos pueden ser:

ROLAP: Significa Proceso Analítico Relacional en Línea y es un intento por desarrollar sistemas dimensionales basados en DBMS relacionales. Es decir, se trata de sistemas y herramientas OLAP construidos sobre sistemas relacionales. No requieren, en principio, el almacenamiento de información, ya que pueden acceder directamente a la fuente de dichos datos, estas herramientas tipo ROLAP acceden a sistemas relacionales y generan consultas SQL para calcular la información a nivel apropiado cuando el usuario final lo requiere.

MOLAP: Significa Procesamiento Analítico Multidimensional en Línea, el cual hace uso de un DBMS multidimensional para proporcionar el análisis y poder visualizar datos de forma multidimensional. La principal diferencia con las herramientas ROLAP es que se requiere un pre-procesamiento y almacenamiento de la información contenida en el cubo OLAP. MOLAP almacena estos datos en una matriz de almacenamiento multidimensional optimizada. Como resultado de esto se obtienen un mejor desempeño, pero se requiere un mayor espacio de almacenamiento.

HOLAP: Significa Procesamiento Analítico Híbrido en Línea y es una combinación de las anteriores, almacenando lo datos en una base de datos relacional mientras que las agregaciones de estos, se almacena en una base de datos multidimensional.

Tanto las herramientas ROLAP como las MOLAP están diseñadas para hacer el análisis de datos a través del uso de modelos de datos multidimensionales, aunque en el caso de ROLAP estos modelos no se implementan sobre un sistema multidimensional, sino sobre un sistema relacional clásico. ROLAP requiere de menor almacenamiento que MOLAP, pero es más lento para hacer análisis de datos.

2.1.1. Cubo OLAP

Los almacenes de datos usan un modelo de datos basado en un modelo multidimensional. Este modelo está típicamente organizado alrededor de un tema central (por ejemplo, ventas o envíos) y dimensiones [14]. Este tema central está representado por la tabla de hechos. Los hechos representan el punto de interés para el proceso de toma de decisiones. Las dimensiones son perspectivas o entidades respecto a las cuales una organización desea ver su información. Cada dimensión puede tener una tabla asociada a ella la cual es llamada "*tabla dimensión*" que describe a la dimensión. Por ejemplo, la tabla dimensión para producto (item) puede contener los atributos: *nombre_producto*, *marca* y *tipo*. Estas dimensiones definen el nivel de granularidad en la cual podemos representar los hechos adicionalmente la tablas de hechos contienen columnas numéricas llamadas medidas. A estas medidas se les puede aplicar funciones de agregación.

Por otra parte, un elemento fundamental de los sistemas OLAP es el concepto de cubo OLAP, estos cubos se componen de agregaciones de las medidas y se clasifican por dimensiones. Estos cubos son estructuras de datos que para representarlas existen limitaciones pues se pueden conformar por más de dos dimensiones. Una herramienta apropiada que los maneje deberá ser capaz de representarlos visualmente en forma adecuada, deberá obtenerlos en forma rápida y deberá permitir efectuar una navegación por los diferentes niveles de agregación de los mismos. Aunque el término de "cubo" sugiere que existen tres dimensiones, un cubo puede tener cualquier número de dimensiones, por esta razón se suele usar el término hipercubo en lugar de cubo. En un cubo de datos OLAP, cada nivel de agregación es un cuboide. Dado un conjunto de datos, podemos generar un cuboide para cada uno de los posibles subconjuntos de las dimensiones dadas [14]. En este sentido, un cuboide puede ser visto como la proyección de la tabla de hechos sobre un conjunto de dimensio-

nes, produciendo un conjunto de celdas con medidas asociadas agregadas. Los cubos podrán estar pre-sumarizados por de cada dimensión o conjunto de dimensiones. En la Figura 2.1, se muestra un ejemplo de cubo de datos tridimensional OLAP. En el frente, la cara visible del cubo, se ven cifras de ventas de cada producto (item) para cada trimestre (quarter) en la ciudad de Vancouver, los productos y los trimestres representan las dos primeras dimensiones (item, time). La tercera dimensión en este ejemplo es la ubicación (location) la cual está representada por las diferentes ciudades (Vancouver, Toronto, New York y Chicago). Las cifras de ventas de cada producto para cada ciudad en cada trimestre aparecen en una celda del cubo tridimensional donde los usuarios pueden ver los datos por cualquier dimensión de interés para ellos.

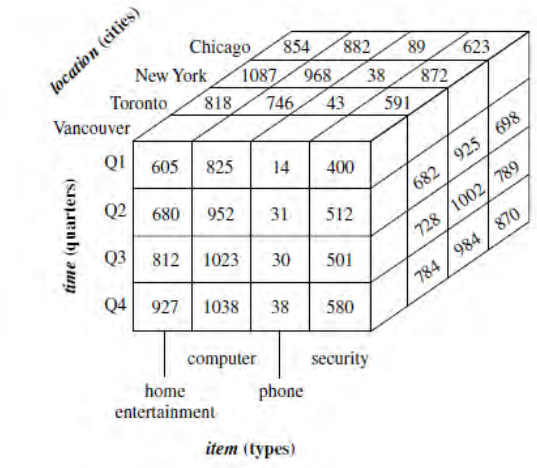

Figura 2.1: Ejemplo de un cubo OLAP, tomada de [14].

2.1.2. Operaciones Sobre Cubos

En el modelo multidimensional, los datos están organizados a través de múltiples dimensiones, y cada dimensión contiene múltiples niveles de abstracción [14]. Esta organización nos permite visualizar la información en diferentes perspectivas las cuales pueden ser el resultado de aplicar diferentes tipos de operaciones sobre un cubo de datos. En [14] se describe las operaciones comunes sobre cubos OLAP, las cuales son:

Roll-Up: La operación de Roll-Up (también conocida como Drill-Up) realiza agregaciones sobre un cubo de datos, ya sea subiendo en la jerarquía de las dimensiones o por la reducción de dimensiones.

Drill-Down: Esta es la operación opuesta de Roll-Up. Se parte de datos menos agregados a datos con un mayor nivel de agregación y detalle. Se puede realizar mediante la introducción de una nueva dimensión o bajando en la jerarquía de dimensiones.

Slice y Dice: La operación Slice realiza una selección en una dimensión de un cubo dado, resultando en un subcubo. La operación Dice define un subcubo mediante la realización de una selección en dos o más dimensiones.

Pivot: La operación Pivot (también es conocida como "Rotate") es una operación de visualización la cual hace girar los ejes de la vista de datos con la finalidad de proporcionar una visualización alternativa de los mismos.

2.1.3. Retícula

Una retícula es una estructura que representa un cubo de datos OLAP de N dimensiones. En una retícula cada nodo es un cuboide, en los cuales se muestran datos a diferentes niveles de agregación [14]. Esta estructura de cuboides es equivalente al cubo de datos explicado anteriormente. Se le denomina *appex* al cuboide 0-dimensional y al cuboide que representa a la agregación de todas las dimensiones de la retícula se le llama *base*. En la Figura 2.2 se muestra una retícula de cuboides que representa al cubo de datos de la Figura 2.1 en el cual se le agregó una dimensión extra (proveedor) dando como resultado un cubo de datos de cuatro dimensiones (tiempo (time), producto (item), ubicación (lacation) y proveedor (supplier)).

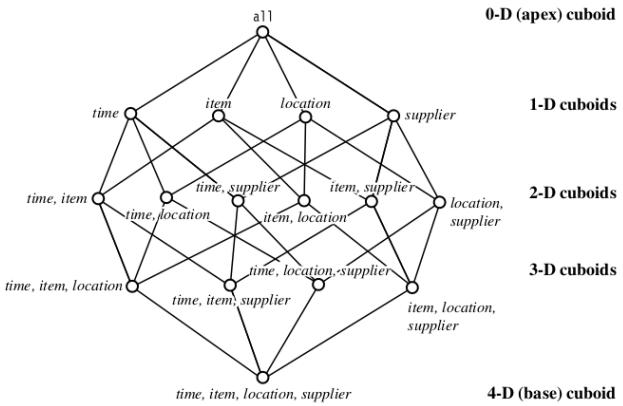

Figura 2.2: Ejemplo de un lattice, tomada de [14]

Número Cuboides por Nivel

Existe una relación entre la cantidad de cuboides por nivel que tiene una retícula, el cual representa un cubo n-dimensional, y los coeficientes del n-ésimo renglón del triángulo de Pascal. El triángulo de Pascal (nombrado así en honor al matemático y filósofo francés Blaise Pascal (1623-1662)) es un triángulo isósceles infinito, en el cual sus lados están formados por unos (Figura 2.3) y los elementos interiores están acomodados de tal forma que cada uno de ellos es la suma de los dos números anteriores [35].

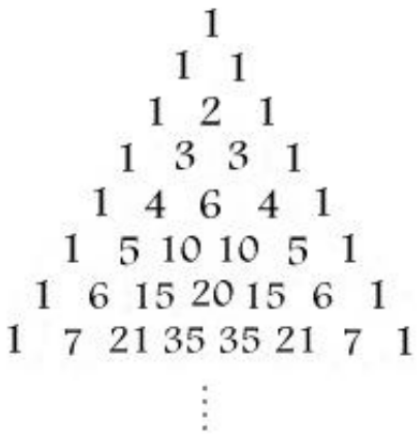

Figura 2.3: Estructura del Triangulo de Pascal, tomada de [7].

De esta estructura podemos observar algunas características relacionadas con la estructura de una retícula como que el número de elementos en el n-ésimo renglón es igual a $n+1$ (dado que la cima del triángulo es considerado el renglón cero), lo cual representa que existen $n+1$ renglones en un retícula de dimensión n. También es importante notar que la suma de los coeficientes del n-ésimo renglón es igual a 2^n, lo cual coincide con el número de cuboides existentes en un retícula n-dimensional. Con esto podemos saber el número de renglones y cuantos cuboides tendrá cualquier retícula.

Para saber el número de cuboides por nivel en el $k-$ésimo nivel de un retícula de dimensión n tenemos lo siguiente: nos fijamos en el renglón n del triángulo, este número aparecerá en la k-ésima posición del renglón n (ya que el primer nivel, el cual inicia con uno, representa un cubo OLAP de dimensión 0). Dado que, los elementos del n-ésimo renglón del triángulo de Pascal coinciden con los coeficientes

binomiales del Teorema del Binomio de Newton (Ecuación 2.1), es posible usar estos coeficientes para calcular el número de cuboides por nivel de la retícula.

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k, \text{ donde } \binom{n}{k} = \frac{n!}{k!(n-k)!}, \tag{2.1}$$

donde n es la dimensión de la retícula y k es el renglón del que se quiere conocer el número de cuboides que contendrá. En la Figura 2.4, se muestra un ejemplo del cálculo de cuboides por nivel usando tanto el triángulo de Pascal y el Teorema del Binomio de Newton para una retícula que representa un cubo de cuatro dimensiones ($n = 4$).

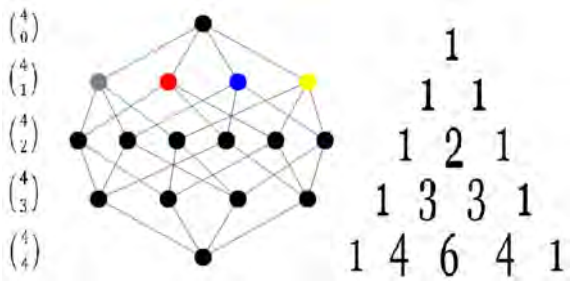

Figura 2.4: Número de cuboides por nivel

2.2. Pruebas de Hipótesis

Una parte importante de la estadística es la inferencia estadística la cual nos permite llegar a generalizaciones respecto de las características de una población, utilizando las observaciones empíricas de una muestra tomada al azar, en este sentido las pruebas de hipótesis son uno de los aspectos más útiles de la inferencia estadística, ya que muchos tipos de problemas de toma de decisiones, ensayos o experimentos en el mundo de la ciencia se pueden formular como prueba de hipótesis.

El propósito de las pruebas de hipótesis es ayudar al médico, investigador o administrador a tomar una decisión en torno a una población, al examinar una muestra de ella [10]. En las pruebas estadísticas se trabaja con dos hipótesis estadísticas. La primera es la hipótesis que debe probarse, mejor conocida como hipótesis nula, y que se representa por el símbolo de H_0. En general la hipótesis nula se establece con el propósito de ser rechazada. Durante el proceso de la prueba, la hipótesis nula puede ser rechazada o no rechazada. Si la hipótesis nula no se rechaza, se dirá que

los datos sobre los cuales se basa la prueba no proporcionan evidencia suficiente que cause el rechazo. Si el procedimiento de la prueba conduce al rechazo, se concluye que los datos disponibles no son compatibles con la hipótesis nula, pero sirve como apoyo para la hipótesis alternativa, la cual está representada por H_a. La verdad o falsedad de una hipótesis estadística nunca se sabe con absoluta certeza, a menos que se examine toda la población, lo cual, por supuesto, sería poco práctico en la mayoría de las situaciones [36]. En vez de eso se toma una muestra aleatoria de la población de interés y se utilizan los datos contenidos en ella para proporcionar evidencia que respalde o no la hipótesis. Los elementos de una prueba de hipótesis [10] son:

- **Hipótesis:** Una hipótesis se define como una proposición acerca de una o más poblaciones. Las hipótesis se establecen de tal forma que puedan ser evaluadas por medio de técnicas estadísticas adecuadas.

- **Estadístico de Prueba:** El estadístico de prueba es alguna estadística que se puede calcular a partir de los datos de la muestra. Este estadístico sirve como un productor de decisiones, ya que la decisión de rechazar o no la hipótesis nula depende de la magnitud del estadístico de prueba.

- **Regiones de Rechazo y Aceptación:** Todos los valores posibles del estadístico de prueba son puntos sobre el eje horizontal de la gráfica de distribución de dicho estadístico y se dividen en dos grupos: región de aceptación y región de rechazo. Los valores en la región de rechazo son aquellos que tienen la menor probabilidad de ocurrir si la hipótesis nula es verdadera, mientras que lo que forman la región de aceptación tienen la mayor probabilidad de ocurrir si la hipótesis nula es verdadera.

- **Regla de decisión:** La regla de decisión señala que se debe rechazar la hipótesis nula si el valor del estadístico de prueba que se calcula a partir de la muestra es uno de los valores de la región de rechazo, y que no se debe rechazar la hipótesis nula si el valor calculado del estadístico de prueba es uno de los valores de la región de aceptación.

- **Nivel de Significación:** El nivel de significación, representado por α, define qué valores del estadístico de prueba pertenecen a la región de rechazo y cuales

pertenecen a la región de aceptación. Se elige un valor pequeño de α para hacer que la probabilidad de rechazo para una hipótesis nula sea pequeña. Los valores de α comúnmente empleados son 0.01, 0.05 y 0.10.

El error que se comete cuando se rechaza un hipótesis nula verdadera se conoce como error de tipo I. El error de tipo II se comete cuando no se rechaza una hipótesis nula falsa, en la Tabla 2.1 se muestra la ocurrencia de los errores de acuerdo a la decisión tomada sobre la hipótesis nula.

		Condición Hipótesis Nula	
		Verdadera	Falsa
Acción	No rechazar H_0	Acción Correcta	Error tipo II
	Rechazar H_0	Error tipo I	Acción Correcta

Tabla 2.1: Condiciones en las que es posible cometer un error de tipo I y un error tipo II, tomado de [10]

A continuación se hace la descripción de las pruebas de hipótesis empleadas en la herramienta y definidas en [10, 36, 11].

Prueba de hipótesis para la media de una población: Es una prueba en la que se hace una suposición (hipótesis) acerca de un parámetro de población (media), con la finalidad de contrastar la hipótesis de que la media (μ) de una población, toma un valor específico (μ_0). El estadístico de prueba y algunas formulaciones de hipótesis son:

$H_0 : \mu = \mu_0$
$H_a : \mu \neq \mu_0$.
Estadístico de Prueba:

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}},$$

donde n es el tamaño de la muestra, σ es la desviación estándar de la población y $\bar{x}$ es la media de la muestra.

Prueba de hipótesis para la diferencia entre las media de dos poblaciones: Es una prueba de hipótesis que comprender la diferencia entre la media de dos poblaciones aleatorias independientes (μ_1 y μ_2), se utiliza con frecuencia para determinar si es razonable o no que las dos poblaciones son distintas entre sí [10]. Las hipótesis se pueden formular de la siguiente manera:

- $H_0 : \mu_1 - \mu_2 = 0 \qquad H_a : \mu_1 - \mu_2 \neq 0$
- $H_0 : \mu_1 - \mu_2 \geq 0 \qquad H_a : \mu_1 - \mu_2 < 0$
- $H_0 : \mu_1 - \mu_2 \leq 0 \qquad H_a : \mu_1 - \mu_2 > 0$

Sin embargo, es posible probar la hipótesis de que la diferencia es igual que, mayor o igual que o menor o igual que algún valor distinto de cero. El estadístico de prueba es

$$z = \frac{(\bar{x_1} - \bar{x_2}) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}},$$

donde $\bar{x_i}$ es la media de la muestra i, σ_i es la desviación estándar de la población i y n_i es el tamaño de la muestra i.

Prueba de hipótesis para la proporción de una población: Es una prueba de hipótesis la cual contempla la proporción p de individuos u objetos en una población que poseen una propiedad especial (por ejemplo, carros con transmisión manual o fumadores que fuman cigarros sin filtro). Si un individuo u objeto con la propiedad es etiquetado como éxito, entonces p es la proporción de población de éxitos. Las pruebas relacionadas con p se basarán en una muestra aleatoria de tamaño n de la población [11]. Las hipótesis de esta prueba y el estadístico de prueba son:

$H_0 : p = p_0$
$H_a : p \neq p_0$.
Estadístico de Prueba:

$$z = \frac{p - p_0}{\sqrt{\frac{p_0 q_0}{n}}},$$

donde n es el tamaño de la muestra, p es la proporción observada, p_0 es la proporción asumida y q_0 es $1 - p_0$.

Prueba de hipótesis para la proporción de dos poblaciones: El objetivo de esta prueba es determinar si dos muestras independientes fueron tomadas de dos poblaciones y las cuales presentan la misma proporción de elementos con una característica especial. Cuando la hipótesis nula que va a probarse es $p_1 - p_2 = 0$, se supone que las proporciones de las dos poblaciones son iguales. Esto se utiliza como justificación para combinar los resultados de las dos muestras

y obtener una estimación ponderada de la proporción común supuesta. Si se adopta este procedimiento, se calcula.

$$\bar{p} = \frac{x_1 + x_2}{n_1 + n_2},$$

donde n_1 y n_2 son, respectivamente, el número de la primera y segunda muestra que poseen la característica en interés. Esta estimación ponderada de $p = p_1 = p_2$ se utiliza para calcular:

$$\hat{\sigma}_{\bar{p}_1 - \bar{p}_2} = \sqrt{\frac{\bar{p}(1-\bar{p})}{n_1} + \frac{\bar{p}(1-\bar{p})}{n_2}},$$

el cual es el error estándar del estimador. Con esto el estimador de la prueba es:

$$z = \frac{p_1 - p_2}{\hat{\sigma}_{\bar{p}_1 - \bar{p}_2}}$$

Prueba de hipótesis para la varianza de una población (χ^2): Esta prueba permite comparar la varianza de una población. Cuando los datos disponibles para el análisis forman una muestra aleatoria simple, extraída de una población que sigue una distribución normal, la estadística de prueba para probar hipótesis acerca de la varianza de una población es:

$H_0 : \sigma^2 = \sigma_0^2$
$H_a : \sigma^2 \neq \sigma_0^2$.
Estadístico de Prueba:

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2},$$

donde σ_0^2 es la varianza de la población, s^2 es la varianza de la muestra y n el tamaño de la muestra. Cuando H_0 es verdadera, sigue una distribución χ^2 con $n - 1$ grados de libertad.

Prueba de hipótesis para la razón de las varianzas de dos poblaciones: Esta prueba de hipótesis compara las varianzas poblacionales, aunque este tipo de problemas surgen con poca frecuencia que los que tienen que ver con medias o proporciones. Los procedimientos de basan en la familia de distribuciones F ($F = \frac{X_1/v_1}{X_2/v_2}$, donde X_i son variables aleatorias chi-cuadradas independientes con v_1 y v_2 grados de libertad, respectivamente). Debido a que F tiene que ver con una relación y no con una diferencia, el estadístico de prueba es la

relación de las varianzas muéstrales; la afirmación de que $\sigma_1^2 = \sigma_2^2$ se rechaza si la relación difiere en gran medida de 1.

$H_0 : \sigma_1^2 = \sigma_2^2$
$H_a : \sigma_1^2 \neq \sigma_2^2$.
Estadístico de Prueba:

$$f = \frac{s_1^2}{s_1^2},$$

donde s_i^2 es la varianza de la muestra i.

Prueba de hipótesis de independencia (Chi-cuadrada): Es una prueba estadística no paramétrica para diferenciar entre dos o más muestras donde frecuencias esperadas son comparadas en relación con frecuencias obtenidas. Se utiliza para hacer comparaciones entre frecuencias y no entre valores medios [12]. La prueba de significación χ^2 se refiere esencialmente a la distinción entre frecuencias esperadas y frecuencias obtenidas. Las frecuencias esperadas f_e se refieren a los términos de la hipótesis nula, según la cual la frecuencia relativa (o proporción) se supone es la misma entre los dos grupos. Las frecuencias obtenidas f_o se refieren a los resultados obtenidos en el estudio y que, por consiguiente, pueden variar o no de un grupo a otro. Sólo si la diferencia entre las frecuencias observadas y obtenidas es suficientemente grande, se rechaza la hipótesis nula, y se concluye que existe una diferencia real en la población.

H_0 : La variables son independientes
H_a : La variables no son independientes
Estadístico de Prueba:

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

2.3. Ergonomía

Según la International Ergonomics Association [17] la ergonomía es la disciplina científica responsable de entender las interacciones entre los humanos y los elementos de los sistemas, así como la profesión que aplica teorías, principios, datos y métodos para diseñar con el objetivo de optimizar el bienestar de los humanos y el rendimiento global del sistema persona-máquina. En este sentido, la ergonomía estudia como diseñar sistemas persona-máquina teniendo en cuenta las necesidades de los humanos y centrándose especialmente en los entornos de trabajo, la eficiencia y la seguridad.

2.4. Usabilidad

La usabilidad es una característica muy importante que todo software debe poseer. Al contrario de lo que algunos puedan pensar, la usabilidad no es sólo la apariencia de la interfaz de usuario. En gran parte, lo que hace que algo sea usable es la ausencia de la frustración cuando éste es usado, en este sentido podemos decir que un producto o servicio realmente es usable cuando el usuario puede hacer lo que desea hacer, de la manera que espera hacerlo, sin obstáculos, dudas o preguntas.

Por otro lado, según la ISO 9241-11 [18], define la usabilidad como grado en que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso concreto, esta definición se centra en tres elementos críticos:

- **Usuarios Específicos:** Son los usuarios para quienes el producto fue diseñado.
- **Objetivos Específicos:** Estos usuarios específicos tienen que compartir los objetivos para con el producto, lo que significa que los objetivos del producto representa los objetivos de los usuarios.
- **Contexto Específico:** El producto tiene que ser diseñado para funcionar en el entorno en el cual los usuarios lo usarán.

Además de los elementos críticos expuestos también nos permite apreciar las medidas críticas de usabilidad:

- **Efectividad:** Exactitud e integridad con la que los usuarios alcanzan los objetivos especificados, y por tanto implica la facilidad de aprendizaje, la ausencia de errores del sistema o la facilidad del mismo para ser recordado.
- **Eficiencia:** Recursos empleados (esfuerzo, tiempo, etc.) en relación con la exactitud e integridad con la que los usuarios alcanzan los objetivos especificados.
- **Satisfacción:** Ausencia de incomodidad y existencia de actividades positivas hacia la utilización del producto

En [30] se definen cinco razones por las que un producto es difícil de usar o tiene poca usabilidad, estas son:

1. Un desarrollo basado en las computadoras o en el sistema.
2. El grupo de usuarios para quien es desarrollado el sistema cambia.
3. Es complicado desarrollar productos usables.
4. El grupo de especialistas de diferentes áreas no siempre trabaja en equipo.
5. El diseño y la implementación no siempre coinciden.

CAPÍTULO 3

Almacenes de Datos y Sistemas Manejadores de Bases de Datos Analíticos

En este capítulo se tratarán los puntos más importantes de los almacenes de datos así como de los sistemas manejadores de base de datos analíticos tanto columnares como por arreglos específicamente el caso de *Vertica* y *SciDB*.

3.1. Almacenes de Datos

Un almacén de datos es un repositorio de información recopilada de múltiples fuentes y almacenadas bajo un esquema unificado que usualmente reside en un solo lugar. Los almacenes de datos se construyen a través de un proceso de limpieza (data cleaning), integración (data integration), transformación (data transformation), carga (data loading) y actualización periódica (data refreshing) de datos [14].

Para facilitar la toma de decisiones, los datos en un almacén de datos están organizados entorno a los temas principales y almacenados para proveer información desde una perspectiva histórica, estos datos son típicamente agregados.

Características de un Almacén de Datos

Según William H. Inmon, un destacado arquitecto en la construcción de sistemas para almacenes de datos [16], un almacén de datos es una colección de datos orientados a temas, integrados, no volátiles y variantes en el tiempo, organizados para soportar necesidades organizacionales. De esta definición podemos apreciar las cuatro características principales que distinguen a un almacén de datos de cualquier otro tipo de repositorio.

Orientados a Temas: Los almacenes de datos están organizados y agregados en torno a los principales temas en lugar de concentrarse en las transacciones y operaciones del día a día de una organización, ya que se enfocan en el modelado y análisis de los datos para la toma de decisiones. Por lo tanto, los almacenes de datos suelen ofrecer una visión sencilla y concisa en cuestiones particulares excluyendo datos que no son útiles en el proceso de toma de decisiones [8].

Integrados: Los almacenes de datos son bases de datos centralizadas y consolidadas que integran datos provenientes de múltiples fuentes de información con diversos formatos. La integración de los datos implica que todas las entidades del negocio, elementos y características de los datos así como las métricas del negocio están descritas de alguna forma. Los datos en el almacén deben ser ajustados a formatos y estructuras uniformes para evitar conflictos [8].

No Volátiles: Una vez que los datos se encuentran en el almacén de datos, estos nunca se eliminarán, ya que representan la información histórica de la empresa. La información operacional que representa la historia a corto plazo de la empresa será continuamente agregada dejando al almacén en un continuo crecimiento [8].

Variantes en el Tiempo: Los datos en los almacenes de datos representan el flujo de datos a través del tiempo desde una perspectiva histórica en contraste con los datos operacionales, los cuales se enfocan en la transacción actual. Los almacenes de datos también pueden contener datos proyectados generados a través de modelos estadísticos o algún otro modelo. También se dice que es variante en el tiempo, ya que una vez que los datos son agregados al almacén, todas las agregaciones que son dependientes del tiempo son recalculadas. Dado que los datos en el almacén constituyen una toma instantánea de la historia

de la empresa, medida por sus variables, el componente de tiempo se vuelve crucial. Por lo que es importante adicionar una dimensión de tiempo para lograr que el análisis de los datos sea más sencillo así como las comparaciones [8].

3.2. Sistemas Manejadores de Bases de Datos Columnares

Los sistemas manejadores de base de datos (DBMS) orientados a columnas, como su nombre lo indica, almacenan los datos en forma de columnas y no por renglones, como lo haría un DBMS clásico [23]. En la Figura 3.1 se muestra un ejemplo de particionamiento columnar de una tabla. En este tipo de almacenamiento, cada elemento de la columna (o atributo) es almacenada de forma contigua de tal forma que se minimiza el tiempo de lectura de disco, que puede ser considerable cuando se realiza un gran volumen de cálculos. Esto presenta una ventaja en ambientes analíticos como lo son los almacenes de datos, en los cuales las consultas típicamente implican la realización de agregaciones sobre grandes cantidades de datos. Como resultado de una arquitectura basada en columnas solo aquellas columnas necesarias para una consulta en específico son leídas de disco. Debido a que en un ambiente analítico la mayoría de las consultas requieren de unas pocas columnas para ser procesadas. Esto hace que las bases de datos columnares presenten un mejor rendimiento en ambientes analíticos ya que su procesamiento no se ve afectado por la carga de datos innecesarios a memoria [23].

En los DBMS relacionales comerciales las tuplas de datos son almacenadas junto con índices auxiliares (árboles B) sobre atributos de la tabla. Tales índices pueden ser primarios, mediante los cuales se almacenan los registros de la tabla lo más cerca posible de forma ordenada de acuerdo al atributo especificado, o secundarios, en cuyo caso no se hace un intento por mantener los registros adyacentes de acuerdo al atributo especificado [34]. Estos índices son adecuados para sistemas transaccionales OLTP, pero no para ambientes optimizados para lectura (OLAP).

Dentro de los sistemas manejadores bases de datos columnares existen diferentes arquitecturas, una de las principales es C-Store [34] en la cual se hace uso de un almacenamiento híbrido mediante la implementación de un componente optimizado

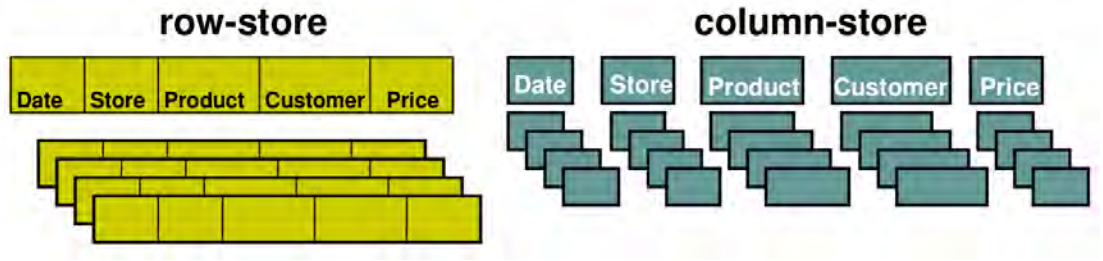

Figura 3.1: Almacenamiento columnar, tomado de [1]

de operaciones de escrituras (operaciones de inserción y actualización de datos), el cual permite soportar consultas tipo OLTP con un tiempo de respuesta razonable y un componente optimizado para lecturas para consultas tipo OLAP.

3.2.1. Compresión de Datos

A menudo los DBMS basado en columnas hacen uso de técnicas de compresión de datos sobre las diferentes columnas (atributos), lo que les permite reducir el espacio de disco necesario para su almacenamiento hasta 20 veces menor al requerido originalmente. Otra de las ventajas de la compresión de datos es que al reducirse la información, puede almacenarse una mayor cantidad por bloque de disco y en consecuencia leer mayor información en una sola lectura a este, esto provee un gran rendimiento y reduce el costo de almacenamiento.

Algunas técnicas de compresión de datos, tomadas principalmente de la arquitectura C-Store [34] mencionada anteriormente, se muestran a continuación.

Self-Order (Pocos Valores Distintos): Esta codificación es representada por una triada (v, f, n), donde v es un valor de la columna, f es el posición en la columna donde aparece por primera vez v y n es el número de veces que parece v en la columna. Por ejemplo, si un grupo de cuatros aparece en las posiciones 12-18 entonces la codificación está dada por la triada (4,12,7). Para columnas que están auto ordenadas se requiere una triada para cada valor distinto de la columna. Para soportar consultas sobre valores.

Foreign-Order (Pocos Valores Distintos): Esta codificación es representada por una secuencia de tuplas (v, b), donde v es el valor de la columna y b es un mapa de bits (Bitmap), el cual indica las posiciones en la cual se encuentra el valor almacenado. Por ejemplo, dada una columna con enteros 0,0,1,1,2,1,0,2,1

la codificación está dada por tres tuplas: (0, 110000100), (1, 001101001) y (2,000010010). Para encontrar el i-ésimo valor de una columna en esta codificación, se incluyen índices "Offset" que son árboles B los cuales hacen el mapeo de posiciones en la columna con los valores contenidos en esa columna.

Self-Order (Muchos Valores Distintos): En esta codificación se representa cada valor de la columna como un valor delta del valor anterior en la columna. Por ejemplo, una columna con valores 1,4,7,7,8,12 será representada por la secuencia 1,3,3,0,1,4, donde el primer elemento de la secuencia es el primer elemento de la columna y cada valor subsecuente es un valor delta de los valores previos.

Foreign-Order (Muchos Valores Distintos): Si existen muchos valores distintos en la columna, entonces lo mejor será dejar los valores como están, es decir sin codificación.

3.2.2. HP Vertica

Vertica es un sistema manejador de base de datos relacional (RDBMS) orientado a columnas que fue diseñado por Michael Stonebraker, uno de los investigadores más importantes en el área de base de datos. *Vertica* hace uso de una arquitectura de procesamiento paralelo masivo (MPP, por sus siglas en inglés) [21], donde los datos están distribuidos a lo largo de los nodos de la red (nodos esclavos) mientras que todos los metadatos residen en un solo nodo (nodo maestro). Esta arquitectura permite una mejor escalabilidad con tan solo agregar nuevos nodos a la red.

Muchas de las ideas implementadas en *Vertica* fueron tomadas de [34]. Dado que fue diseñado para optimizar las operaciones de disco, minimiza el espacio en este haciendo uso de algoritmos de compresión de datos elegidos según el tipo, la cardinalidad y el ordenamiento de los mismos [23]. Para cada columna, se elige el algoritmo apropiado de forma automática basado en un muestreo de los datos [27]. Además, como se trata de un sistema orientado al análisis, *Vertica* implementa una serie de algoritmos de aprendizaje automatizado para el análisis de datos como lo son: algoritmos de regresión lineal, regresión logística y algoritmos de clustering como k-means, así como, métodos de pre procesamiento de datos como normalización de

datos y métodos de evaluación como matrices de confusión, curvas ROC entre otros [27]. A continuación, se presentan algunas características importantes.

Modelo de Almacenamiento

Cada nodo en *Vertica* está organizado con un almacenamiento híbrido el cual se compone de dos elementos, uno llamado *Read Optimized Store* (ROS) y otro llamado *Write Optimized Store* (WOS), lo cual retoma las ideas establecidas en el artículo C-Store [34]. Generalmente el WOS reside en la memoria principal y está diseñado para soportar de manera eficiente operaciones de inserción y actualización de datos, así como permitir la carga masiva de datos. En esta área los datos se encuentran almacenados como una colección de columnas sin compresión ni ordenamiento. Por otra parte, el componente ROS es capaz de soportar grandes cantidades de información. Como su nombre lo indica el ROS, está optimizado para la lectura de datos y solo es compatible con un tipo especial de inserción, la cual se realiza cuando se mueve datos de un componente a otro. Los datos en el ROS se encuentran almacenados de una forma ordenada y comprimida por lo que pueden ser leídos y consultados de manera eficiente. El movimiento de datos entre el WOS y el ROS es llevado a cabo por un proceso asíncrono llamado *Tuple Mover* (TM), como se puede observar en la Figura 3.2. Es por esto que se dice que *Vertica* tiene un modelo de almacenamiento híbrido [26], porque se comporta como un sistema relacional cuando se realizan cargas y modificaciones de datos y se comporta como un sistema columnar cuando se hacen consultas y análisis de datos.

Para poder almacenar información, está primero entra al WOS que reside en memoria y mantiene los datos sin compresión alguna ni indexación. El TM se encarga de mover los datos existentes en el WOS y por consiguiente en memoria, al ROS que se encuentra en disco, este movimiento de datos se realiza en segundo plano y puede ser configurado el tiempo en que se realiza por el usuario [26]. También es posible, como se observa en la Figura 3.2, que se realice una carga directa al ROS solo que esta será más lenta y es recomendada solo para cuando se requiere hacer una inserción de muy grandes volúmenes de datos (gigas o terabytes de información). Las consultas y las actualizaciones no interfieren entre sí. Las actualizaciones residen en áreas basadas en el tiempo llamadas “epoch” hasta que la transacción haga commit. Los datos que no están en el epoch actual son candidatos para consultas y para ser enviados al área de ROS, en la Figura 3.3 se muestra un ejemplo de este proceso.

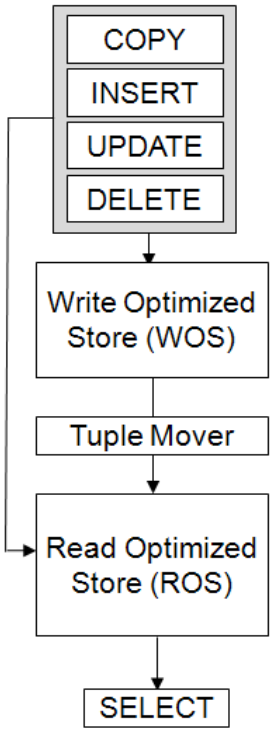

Figura 3.2: Modelo de Almacenamiento en *Vertica*, tomado de [26]

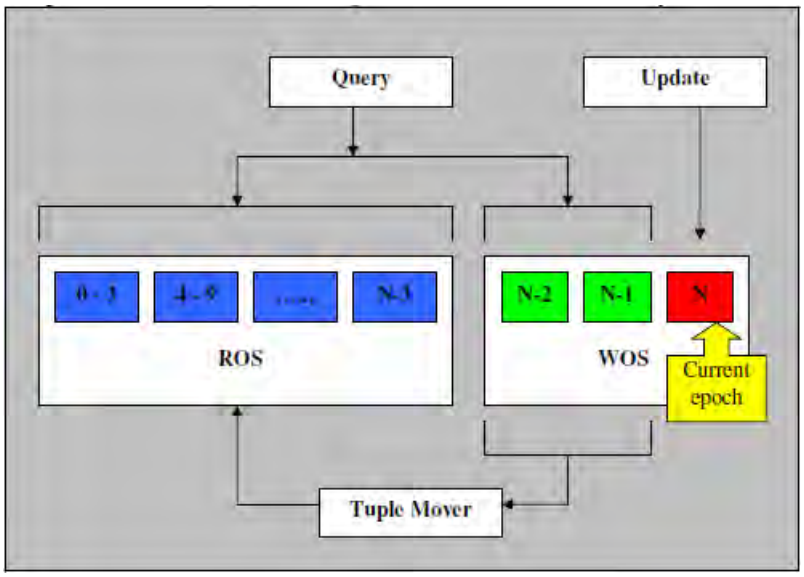

Figura 3.3: Modelo de Almacenamiento en *Vertica*, tomado de [23]

Modelo de Datos

Vertica hace uso de un modelo de datos por tablas de columnas (atributos) como cualquier sistema basado en SQL, aunque la manera de almacenamiento físico no es

la misma que la de los manejadores relacionales [21]. El lenguaje de manipulación de datos es SQL (*Structured Query Language*), esto representa una alta compatibilidad con sistemas existentes y estandarizados, además de tener una alta compatibilidad con MapReduce y Hadoop.

Vertica descompone las tablas lógicas y las almacena físicamente como grupos de columnas llamados "*proyecciones*". Estas proyecciones se almacenan de diferentes formas, algo similar a las vistas materializadas. Las proyecciones pueden ser vistas como una forma restringida de vista materializada, la diferencia radica en que una vista materializada puede contener agregaciones, reuniones y otro tipo de construcciones que las proyecciones de *Vertica* no. Cada proyección tiene un subconjunto de columnas de una o más tablas que están ordenadas por diferente atributo. Esta selección de atributos para las proyecciones se hace automáticamente de forma que se optimice el rendimiento de las consultas. Estas proyecciones también sirven como copias redundantes con lo cual asegura la tolerancia a fallos y mejorar el rendimiento de consultas concurrentes.

3.3. Sistemas Manejadores de Base de Datos por Arreglos

Los sistemas manejadores de base de datos por arreglos son sistemas en los cuales el modelo de datos es el de los arreglos multidimensionales. Estos sistemas a menudo están diseñados para dominios específicos de aplicación como la computación científica y procesamiento analítico en línea (OLAP). A continuación, se presentan las principales características de estos sistemas y la descripción del DBMS *SciDB*.

3.3.1. Arreglos

Un arreglo (array), desde el punto de vista de los lenguajes de programación, es una estructura de datos que almacena una colección de datos del mismo tipo. Aunque el concepto de arreglo en programación y el del modelo de datos es muy similar existen algunas desventajas las cuales presentan los arreglos de los lenguajes de programación, las cuales son:

- Todos los elementos de los arreglos deben tener el mismo tipo.

- En general, el tamaño del arreglo es fijo (no van creciendo dinámicamente como es el caso de las listas).
- Se ocupan principalmente para almacenar datos numéricos.
- Solo pueden almacenar un dato por celda.

Estos puntos no son un problema cuando hablamos de arreglos (arrays) multidimensionales en el contexto de los sistemas manejadores de bases de datos. Estos arreglos son estructuras de datos que almacenan los valores en más de una dimensión. Dicho de otro modo, un arreglo multidimensional es como un contenedor que guardará más valores para cada posición, es decir, como si los elementos del arreglo fueran a su vez otros arreglos. En [31] se presenta un estudio sobre el modelo de datos por arreglos, del cual se tomó la siguiente definición formal de un arreglo multidimensional (multidimensional array).

Sea un multiconjunto de dominio discreto $D_i = [l_i, u_i], i \in \{1, 2, \ldots, N\}$ donde cada dominio D_i contiene enteros entre l_i y u_i. Un arreglo N-dimensional con M atributos $A_j, j \in \{1, 2, \ldots, M\}$, puede ser visto como una función definida sobre dimensiones y tomando valores de los atributos tupla, como se muestra.

$$Array : D_1 \times D_2 \times \cdots \times D_N \mapsto (A_1, A_2, \ldots, A_M), \tag{3.1}$$

donde el tipo de los atributos puede ser cualquier tipo de datos encontrado en el modelo relacional o cualquier tipo de dato definido por el usuario haciendo uso de las mismas ideas detrás de los tipos de datos extendidos.

3.3.2. Chunks

La forma de organizar un arreglo en disco es muy importante para su funcionamiento óptimo, tanto para su almacenamiento como para su procesamiento e intercambio de información entre nodos. El tamaño de un arreglo está dado por $|D_1| * |D_2| * |D_N| * |sizeof(A_1, A_2, ..., A_N)|$ [31], sin embargo esto es muy grande para ser cargado a memoria, por lo que los arreglos están organizados en disco en bloques llamados "chunks"los cuales dividen el arreglo en pequeñas porciones. Cuando un elemento del chunk tiene que ser cargado a memoria, todo el bloque es leído en una sola lectura, de hecho los chunks en los manejadores de base de datos por arreglos representan las unidades de I/O de disco, esto es similar a las páginas en

los sistemas de archivos o los bloques de los sistemas relacionales. Al igual que los sistemas columnares, los DBMS por arreglos particionan el arreglo verticalmente para su almacenamiento en disco, esto le da muchas de las ventajas expuestas anteriormente. El proceso mediante el cual se hace el almacenamiento de los arreglos en *SciDB* es el siguiente [6]: considere un arreglo de dos dimensiones (I y J) con dos atributos (A y B) como el que se muestra en la Figura 3.4, primero se realiza un particionamiento vertical sobre los atributos del arreglo (Figura 3.5), dejando arreglos simples de un solo atributo, después el manejador toma estos arreglos simples y los particiona en bloques de igual tamaño que pueden traslaparse o no, dependiendo de las necesidades del usuario (Figura 3.6). En *SciDB* los chunks son las unidades de I/O, procesamiento y de intercomunicación con los nodos.

J \ I	[0]	[1]	[2]	[3]	[4]
[0]	(2, 0.7)	(5, 0.5)	(4, 0.9)	(2, 0.8)	(1, 0.2)
[1]	(5, 0.5)	(3, 0.5)	(5, 0.9)	(5, 0.5)	(5, 0.5)
[2]	(4, 0.3)	(6, 0.1)	(6, 0.5)	(2, 0.1)	(7, 0.4)
[3]	(4, 0.25)	(6, 0.45)	(6, 0.3)	(1, 0.1)	(0, 0.3)
[4]	(6, 0.5)	(1, 0.6)	(5, 0.5)	(2, 0.15)	(2, 0.4)

Figura 3.4: Estructura de un arreglo, tomado de [6].

{A}

J \ I				
2	5	4	2	1
5	3	5	5	5
4	6	6	2	7
4	6	6	1	0
6	1	5	2	2

{B}

J \ I				
0.7	0.5	0.9	0.8	0.2
0.5	0.5	0.9	0.5	0.5
0.3	0.1	0.5	0.1	0.4
0.25	0.45	0.3	0.1	0.3
0.5	0.6	0.5	0.15	0.4

Figura 3.5: Particionamiento del arreglo por atributos, tomado de [6].

$\{A_1\}$

J \ I		
2	5	4
5	3	5
4	6	6

$\{A_2\}$

J \ I		
4	2	1
5	5	5
6	2	7

$\{A_3\}$

J \ I		
4	6	6
4	6	6
6	1	5

$\{A_4\}$

J \ I		
6	2	7
6	1	0
5	2	2

Figura 3.6: Descomposición cada atributo en porciones de igual tamaño (chunks), tomado de [6].

3.3.3. Diferencia entre Arreglos y Tablas

Una tabla puede ser vista como un arreglo sin dimensiones (solo con atributos) si no existe una función de ordenamiento que permita identificar una tupla basado en índices. Para pasar de tablas a los arreglos es necesario que los atributos que representarán las dimensiones del arreglo formen una dependencia funcional con el resto de los atributos de la tabla. Debido a esto los atributos de una llave pueden ser transformados directamente a dimensiones; por el contrario , transformar dimensiones en atributos no es tan sencillo.

Convertir una dimensión en un atributo es equivalente a destruir las propiedades del arreglo y perder cualquier información ordenada. Como tal, un arreglo puede ser visto como un tipo particular de tabla organizada a lo largo de la dimensiones. En esencia, la expresión $Array[d_1, d_2, \ldots, d_N]$, donde $d_i \in [l_i, u_i]$, solo tiene sentido para un arreglo y es determinado de forma única [31]. Lo mismo es cierto para una tabla en la cual $(d_1, d_2, \ldots, d_N)$ representa la llave de la tabla . Lo que distingue un arreglo de una relación es que el arreglo está organizado de tal manera que realizar una búsqueda puede hacerse directamente a partir de los valores de los índices (la posición), sin tener en cuenta ninguna otra entrada. Esto no es posible en una tabla ya que no existe una correspondencia entre los índices y la posición actual en la representación física, por lo menos en el concepto abstracto del modelo relacional [31]. Por consiguiente, la principal diferencia entre arreglos y relaciones está en el nivel de organización física ya que los arreglos son un tipo particular de relación desde una perspectiva abstracta.

3.4. SciDB

SciDB es un sistema manejador de base de datos de almacenamiento masivo paralelo que es capaz de paralelizar algoritmos de procesamiento de arreglos a gran escala, está diseñado principalmente para aplicaciones de manejo de grandes volúmenes de datos (petabytes) científicos en arreglos, como lo son, aplicaciones relacionadas a la astronomía, modelado del clima, bio-ciencia, aplicaciones comerciales como sistemas de manejo de riesgo en el sector financiero, etc. En contraste con los sistemas de manejo de información comerciales, el análisis científico típicamente requiere

del procesamientos de métodos para datos matemáticamente y algorítmicamente sofisticados.

A diferencia de los modelos relacionales donde el ordenamiento no aporta información, sistemas donde el ordenamiento de la información debe ser explícitamente definida como parte del esquema y ordenada en la base de datos, un modelo de datos por arreglos puede ayudar en nociones de adyacencia y vecindad, estos conceptos son más deseados en el campo científico. En contraste con los paquetes estadísticos como R, MatLab o SAS, *SciDB* ofrece la escalabilidad de un DBMS. A continuación se explicara algunas de las características de *SciDB*, como:

- Arquitectura
- Modelo de Datos
- Lenguajes de Manipulación de Datos

3.4.1. Arquitectura

SciDB adopta una arquitectura shared-nothing que consiste en una arquitectura de cómputo distribuida en la cual cada nodo es independiente y autosuficiente. Básicamente consiste en máquinas con procesadores, memoria y discos independientes.

En esta arquitectura existen dos clases de nodos, un nodo coordinador (coordinator) el cual gestiona la ejecución de las consultas además de la recolección de los resultados y los nodos trabajadores (workers) quienes se encargan de la ejecución de las consultas. Es responsabilidad del coordinador mediar entre todas las comunicaciones que existen entre la base de datos y los clientes de consulta (*SciDB* Client). El procedimiento mediante el cual se ejecuta una consulta en esta arquitectura es el siguiente: el nodo coordinador toma una consulta y lo divide entre los demás nodos (workers) en cargas de trabajo similares, una vez que los workers obtienen los resultados de su tarea son enviados al coordinador quien se encarga de recolectar los resultados generados por los workers y regresarlos al cliente.

Los procesos de *SciDB* se ejecutan en cada nodo compartiendo acceso a un catálogo (lógico) de sistema centralizado el cual contiene información acerca de los nodos, la distribución de los datos entre los nodos, entre otras cosas. Este catálogo reside en una base de datos de PostgreSQL. En la Figura 3.7 se observa como esta compues-

ta la arquitectura de *SciDB*. Esta arquitectura provee las siguientes ventajas: bajo costo, alta disponibilidad y alta escalabilidad.

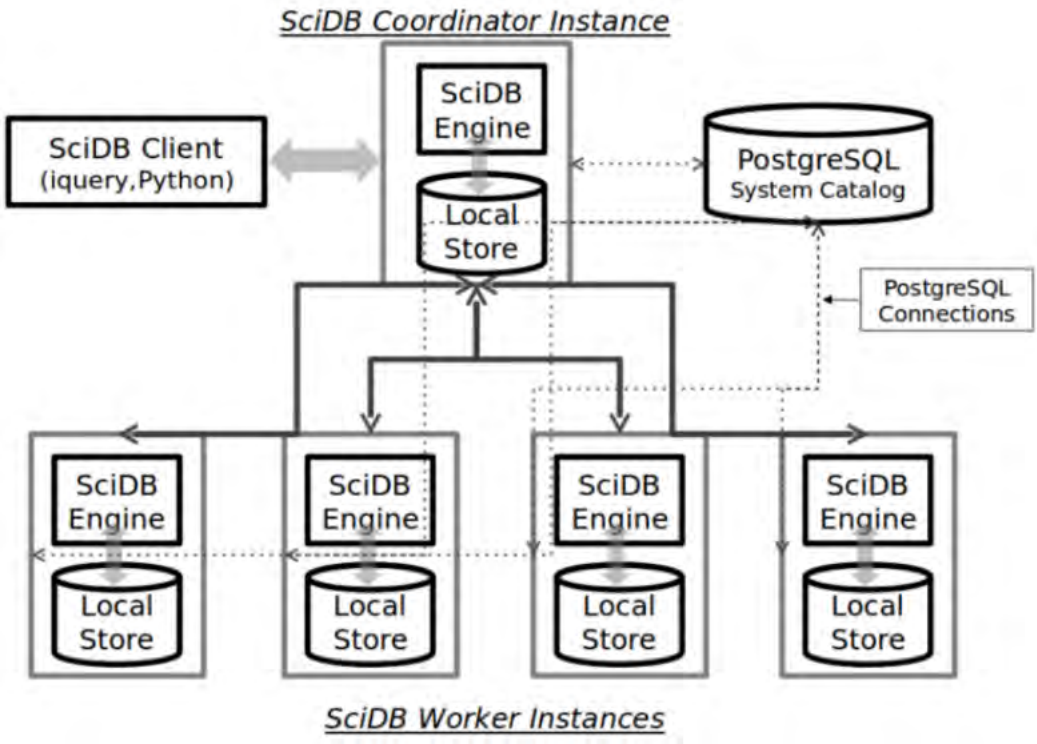

Figura 3.7: Arquitectura *SciDB*, tomado de [28]

3.4.2. Modelo de Datos

SciDB se basa en un modelo de datos por arreglos [9] (en lugar de uno por tablas) ya que los arreglos son los objetos de datos natural en gran parte de las ciencias. Por otra parte, la mayoría de los análisis complejos de la comunidad científica se basan en operaciones básicas de álgebra lineal [5] (por ejemplo, multiplicación de matrices, cálculo de la covarianza, cálculo de la inversa, solución de ecuaciones lineales, etc.). Estas operaciones en un modelo de datos basado en tablas requeriría de una conversión a arreglos y viceversa para poder ser realizadas. En resumen el modelo de datos por arreglos permite tener:

- Estructuras optimizadas para el análisis.
- Estructuras eficientes para datos donde el ordenamiento es una característica inherente.
- Un modelo sensible a operaciones para análisis y manipulación de arreglos.

Dimensiones

Un arreglo n dimensional tiene $d_1, d_2, \ldots, d_n$ dimensiones. *SciDB* permite tener cualquier número de dimensiones para un arreglo. Estas pueden ser dimensiones enteras (tradicionales), iniciando y terminando en cualquier punto e incluso no tener definido un límite (dimensión ilimitada) en cualquier dirección. Por otra parte, muchos arreglos son más naturales con dimensiones no enteras por lo que *SciDB* también soporta dimensiones de punto flotante (1.2, 2.76, 4.3,...) y dimensiones por cadenas (alpha, beta, gamma,...), aunque estas últimas fueron descontinuadas a partir de la versión 13.12. En la Figura 3.8 se muestra una representación de un arreglo de dos dimensiones con tres atributos en cada celda.

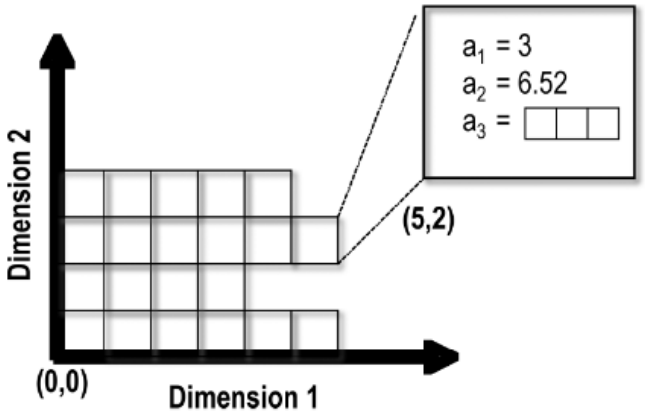

Figura 3.8: Representación de un arreglo de 2 dimensiones, tomado de [22]

Atributos

Cada combinación de valores de las dimensiones identifican a una celda o elemento dentro de un arreglo, el cual puede contener múltiples valores llamados atributos $(a_1, a_2, \ldots, a_m)$. Cada atributo pertenece a un tipo de dato. En la Figura 3.9 se muestra un ejemplo de la definición de un arreglo de dos dimensiones (I y J) con tres atributos (v_1, v_2 y v_3). En este ejemplo podemos observar que los atributos pueden ser de diferente tipo y que a la dimensión I no se le define un límite por lo que puede crecer indefinidamente.

```
CREATE ARRAY Simple_Array
< v1 : double,
  v2 : int64,
  v3 : string >
[ I = 0:*, 5, 0, J = 0:9, 5, 0 ];
```

Attributes	Dimensions	Dimension size	Chunk	Chunk
v1, v2, v3	I, J	* is unbounded	size	overlap

Figura 3.9: Definición de un arreglo de 2 dimensiones con 3 atributos, tomado de [22]

3.4.3. Lenguajes de Manipulación de Datos

SciDB soporta dos tipos de lenguajes, uno funcional y otro similar a SQL. El funcional es llamado AFL (Array Functional Language); el lenguaje similar a SQL es llamado AQL (Array Query Language). Las consultas AQL son compiladas y transformadas a AFL. El lenguaje funcional (AFL) incluye una colección de operaciones las cuales el usuario puede encapsular para obtener el resultado esperado, a continuación se muestra dos consultas, una en AQL y otra en AFL, con las cuales se obtiene el mismo resultado. Sea A y B dos arreglos de dos dimensiones (I y J) y c un atributo de A, entonces las consultas equivalentes son:

```
# AQL Query
AQL% SELECT *
     FROM   A,B
     WHERE  A.I = B.I AND A.J = B.J AND A.c = value;

# AFL Query
AFL% JOIN(B, FILTER(A, c = value), I, J);
```

Si bien el lenguaje de consulta AQL es similar a SQL se tienen que considerar algunas restricciones y condiciones antes de realizar una consulta. Una consulta SELECT tradicional en un modelo relacional de SQL tiene la siguiente sintaxis

```
SELECT   column_name, ...
FROM     table_name
WHERE    column_name operator value;
```

Esto es muy similar para el caso de AQL solo que en lugar de proyectar columnas se hace la proyección de atributos y dimensiones de un arreglo, como se muestra a continuación.

```
SELECT   [attribute_name | dimension_name], ...
FROM     array_name
WHERE    [attribute_name | dimension_name] operator value;
```

Las reuniones y agregaciones en AQL son similares a las de SQL con la restricción de que solo pueden hacerse mediante las dimensiones.

```
# SQL sintax
SELECT   column_name, aggregate_function(column_name), ...
FROM     table_name
WHERE    column_name operator value
GROUP BY column_name ;

# AQL sintax
SELECT   [attribute_name | dimension_name],
         aggregate_function(attribute_name),...
FROM     array_name
WHERE    [attribute_name | dimension_name] operator value
GROUP BY dimension_name;
```

CAPÍTULO 4

Pruebas de Usuario

La comprensión e identificación de los modelos asociados al diseño y desarrollo de estas interfaces nos posibilitará un acercamiento a la lógica del usuario y a cómo facilitar la consecución de sus objetivos. En este capítulo se hará una descripción del diseño centrado en el usuario y la evaluación de interfaces gráficas con la finalidad de generar un producto que satisfaga las necesidades del usuario.

4.1. Breve descripción del Desarrollo Centrado en el Usuario (DCU)

El Diseño Centrado en el Usuario (UDC del inglés *User-Centred Design*) es una filosofía de diseño y un proceso que toma en cuenta al usuario, sus necesidades, requerimientos y limitaciones a lo largo del proceso de desarrollo y en donde la tecnología se ajusta a este y no al contrario. Al diseño centrado en el usuario (DCU) también se le suele conocer por Diseño Centrado en el Humano (Human-Centred Design), Diseño Centrado en Personas (People-Centred Design) y Diseño Orientado al Usuario/Cliente (User/Client-Oriented Design). Según la ISO 9241-210 [19] define DCU como un enfoque para el desarrollo de sistemas interactivos que tiene por objetivo hacer sistemas usables y útiles, centrándose en los usuarios, sus necesidades y requerimientos, mediante la aplicación de factores ergonómicos. En ocasiones se tiende a confundir usabilidad con DCU, pero aunque la usabilidad es un concepto central e inherente al DCU, es evidente que podemos señalar diferencias entre ambos conceptos. La usabilidad es un atributo de calidad del diseño, mientras que el DCU

es una vía para alcanzar y mejorar empíricamente la usabilidad del producto. Es decir, la usabilidad representa el "qué", mientras el DCU representa el "como"[15]. Existen otros enfoques o filosofías de diseño que diferencian al DCU, como las que describe Kalbach [20]:

Diseño Centrado en el Diseñador (Designer-centered design): En esta perspectiva el diseñador, a partir de su visión personal, sabe qué es lo mejor en cada momento. Esta filosofía podría tener éxito para las empresas más pequeñas, sin embargo cómo el diseño es guiado por la perspectiva y experiencia del diseñador, los objetivos de negocio pueden verse eclipsados por intereses personales.

Diseño Centrado en la Empresa (Enterprise-centered design): Esta es la filosofía más común. El diseño está centrado en la estructura y las necesidades de las partes interesadas en la compañía u organización.

Diseño Centrado en el Contenido (Content-centered design): En esta perspectiva la información contenida es la base para organizar y estructurar la navegación. En cierto sentido esta es una perspectiva natural y está siempre presente. Sin embargo, la cantidad y el tipo de contenido disponible no deben ser la única fuerza determinante en el diseño.

Diseño Centrado en la Tecnología (Technology-centered design): El diseño puede ser determinado por la forma más fácil de implementar una solución. La atención se centra en la aplicación y en llegar a un producto final. Esta perspectiva presenta las ventajas de ser económica y eficiente ya que se puede cumplir con una fecha límite de proyecto. Sin embargo se corre el riesgo de que el usuario no sea capaz de usar o entender el producto final siendo contraproducente a largo plazo para los objetivos del proyecto y del negocio.

Aunque el DCU se centra en el usuario, no implica que se deba de hacer todo lo que el usuario solicite o ignorar otras restricciones del proyecto. Por supuesto, los objetivos de negocio y la tecnología son importantes, pero el diseño centrado en el usuario nos indica que la experiencia del usuario es la meta principal y que los demás puntos de vista son secundarios. Dado que el objetivo del DCU es establecer un conocimiento profundo de los usuarios y sus necesidades, todo diseño

debe comenzar con una compresión de los usuarios, incluyendo perfiles que reflejen sus capacidades físicas y cognitivas, educación, antecedentes culturales o étnicos, formación, motivación y objetivos.

4.2. Evaluación de Interfaces sin Usuarios

Una vez que se ha descrito la herramienta base y con la finalidad de corregir fallas y situaciones complejas, las cuales pueden ser detectadas mediante una prueba de usabilidad, se realizó una evaluación y corrección de la interfaz de la herramienta. Para la evaluación de la interfaz se ha decidido primeramente una prueba de usabilidad sin usuarios y posteriormente una prueba con usuarios. Para la prueba sin usuarios se realizó una evaluación siguiendo criterios ergonómicos desarrollados por Bastien, JM Christian y Scapin, Dominique L.[4], los cuales serán explicados más adelante. Para la prueba con usuarios se realizaron pruebas con usuarios reales en un laboratorio controlado.

4.2.1. Pruebas de Usabilidad

Las pruebas de usabilidad aportan datos tanto cuantitativos como cualitativos sobre usuarios reales que llevan a cabo tareas reales con el software. Durante estas pruebas se recolectan datos empíricos mientras se observan a usuarios finales realizando tareas con el producto. Las pruebas se dividen principalmente en dos tipos de enfoque. El primer enfoque engloba la pruebas formales las cuales tienen el propósito de confirmar o refutar hipótesis. El segundo enfoque, emplea un ciclo iterativo de pruebas ideadas para exponer las deficiencias de usabilidad y poco a poco formar o moldear el producto en cuestión. El objetivo de estas pruebas es identificar y corregir las deficiencias de usabilidad existentes en los productos antes de ser liberados.

4.2.2. Criterios Ergonómicos

A continuación se lista una serie de criterios desarrollados en [4], los cuales pueden guiarnos en la evaluación de un producto desde el punto de vista que no es el técnico. Los criterios están organizados en ocho puntos principales (algunos pueden a su vez están subdivididos en criterios más específicos), y tienen dos objetivos principales:

- Definir o formalizar las diferentes dimensiones que conforman el concepto de "utilizable", sustento de lo que denominamos "Software de Calidad".
- brindar una herramienta que facilite, mejore y documente el proceso de evaluación de las interfaces-usuario.

Estos criterios constituyen una herramienta para la evaluación rápida y económica de interfaces con el objetivo de identificar aquellos errores comunes de diseño. Sin embargo, estos criterios no sustituyen la pruebas de usabilidad con usuarios ya que con estas podemos identificar problemas más complejos o nuevos, los criterios son:

1. **Guía:** Se refiere al conjunto de medios que permiten orientar, informar, instruir o guiar al usuario a través de su interacción con la computadora (mensajes, alarmas, etiquetas, etc.). Se divide en cuatro criterios:

   a) **Incitación:** Agrupa todos aquellos mecanismos que permiten encaminar a los usuarios para que realicen acciones específicas. También engloba las acciones que indican al usuario el conjunto de operaciones posibles, así como aquellas que le ayudan a identificar el lugar donde se encuentra dentro de la aplicación.

   b) **Agrupación/Distinción de elementos:** Evalúa la estructura visual de los diferentes elementos que se encuentran en la interfaz. Este criterio toma en cuenta la topología y la distribución espacial de las informaciones desplegadas, su pertenencia a una misma clase, o la diferenciación entre elementos diferentes.

   El agrupamiento o distinción de elementos puede ser realizado con base a dos criterios diferentes: agrupación/distinción por localización; y agrupación/distinción por formato.

   c) **Retroalimentación Inmediata:** Se refiere a las respuestas que el sistema brinda para cada acción del usuario.

   d) **Legibilidad:** Se refiere a las características de la información en pantalla que puedan facilitar o dificultar su lectura.

2. **Carga de Trabajo:** Concierne a todos los elementos de la interfaz que juegan un rol en la percepción del usuario o en la carga cognitiva, así como en una mejor eficiencia en la interacción. Se divide en dos criterios:

a) **Brevedad:** Corresponde al hecho de limitar la lectura, las acciones de entrada y en general el número de acciones necesarias para realizar una tarea. Este criterio se subdivide a su vez en dos criterios:

- **Concisión:** Se refiere a que las etiquetas, los comandos y las regiones sensibles, deben ser cortos y claros. En el caso de captura de datos, el sistema debe interpretar y completar la información cuando sea posible (ex. ceros a la izquierda, guiones intermedios, etc.).
- **Acciones Mínimas:** Se refiere al número de pasos que un usuario debe realizar para llegar a su objetivo. Entre menor sea este número, más eficiente será el sistema.

b) **Densidad de Información:** Se refiere a la carga de trabajo perceptual y cognitiva ocasionada por grupos de elementos, y no por elementos aislados como en el caso de Brevedad.

3. **Control Explícito:** Concierne al procesamiento por parte del sistema de acciones explícitas del usuario, así como el control que debe tener el usuario sobre un proceso. Se divide en dos criterios:

   a) **Acciones Explícitas del Usuario:** El sistema debe realizar únicamente aquellas acciones señaladas por el usuario.

   b) **Control del Usuario:** El usuario debe tener el control de la aplicación todo el tiempo.

4. **Adaptabilidad:** Se refiere a la capacidad de un sistema para comportarse de manera contextual y de acuerdo a las necesidades y preferencias del usuario. Se divide en dos criterios:

   a) **Flexibilidad:** Es la capacidad de la interfaz para adaptarse a las necesidades particulares de los usuarios. Una buena flexibilidad permite al usuario adaptar la interfaz a sus necesidades.

   b) **Experiencia del usuario:** Mecanismos que permiten respetar el nivel de experiencia del usuario en cuanto al uso del sistema.

5. **Manejo de Errores:** Se refiere a los medios disponibles para prevenir o reducir errores, y para recuperar la información cuando éstos ocurran. Se divide en tres criterios:

a) **Protección contra los Errores:** Evitar en lo posible que el usuario cometa errores.

b) **Calidad de los Mensajes de Error:** Los mensajes de error deben de ser claros y concisos. Además, deben indicar cómo corregir el error.

c) **Corrección de los Errores:** Una vez que el usuario ha cometido un error, el sistema debe ofrecer mecanismos que permitan la recuperación del estado inmediato anterior.

6. **Consistencia:** Se refiere a la manera en que el diseño de una interfaz se mantiene para contextos similares, y se diferencia para contextos diferentes.

7. **Significado de Códigos:** Califica la relación entre un término y/o un signo, y el objeto o comando al que hace referencia. Los códigos y los nombres son importantes para los usuarios cuando existe una relación clara entre tales códigos y las acciones.

8. **Compatibilidad:** Se refiere a la relación que hay entre las características del usuario (memoria, capacidad cognitiva, capacidad perceptual, experiencia, preferencias, etc.) y su tarea (qué hace, cómo lo hace, que objetos utiliza, en qué momento, etc.), con respecto a la organización de las entradas/salidas, y el diálogo de la aplicación.

4.3. Evaluación del Sistema

Como se mencionó en la sección anterior, los criterios ergonómicos representa una herramienta muy importante para la evaluación de una interfaz que se le presenta a un usuario con el objetivo de saber que tan usable es. A continuación se presentará una evaluación que se realizó a cada una de las interfaces de la herramienta base aplicando los criterios ergonómicos y mencionando cuales son los principales problemas que presenta, para posteriormente presentar una propuesta de interfaz, la cual corrija o minimice estos problemas.

Figura 4.1: Interfaz de selección de DBMS.

Criterio	**Sub-Criterio**	**Descripción**
Guía	Incitación	• No es claro que es lo que se espera que ingrese por parte del usuario, el número o el nombre del sistema manejador de base de datos. • No hay una indicación que muestre una invitación a teclear una palabra en el área de entradas.
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico.
Manejo de Errores	Protección Contra los Errores Calidad de los Mensajes de Error Corrección de los Errores	No existe nada que evite que el usuario ingrese algo que no esperamos. No cuenta con ningún mensaje de Error. La interfaz no permite al usuario regresar a un estado anterior una vez que se ha cometido un error.
Significado de Códigos		• No es claro que significa el signo de interrogación. • No es claro lo que el botón "Cancelar" hace.

Continúa en la siguiente hoja

Tabla 4.1 – *Continuación*

Criterio	Sub-Criterio	Descripción
		• El título de la interfaz dice "Seleccionar" pero se espera que el usuario ingrese el tipo de manejador de base de datos.

Tabla 4.1: Tabla de análisis de la interfaz de selección de DBMS.

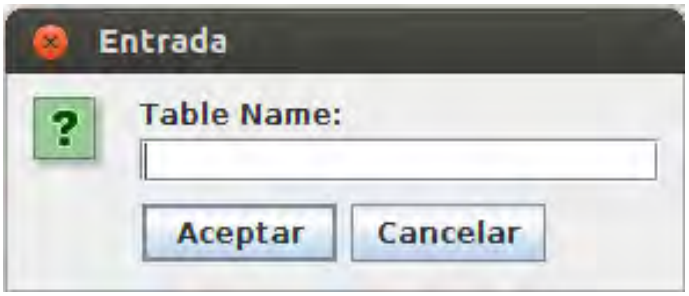

Figura 4.2: Interfaz de selección de tabla.

Criterio	Sub-Criterio	Descripción
Guía	Incitación	• No es claro que es lo que se espera que ingrese por parte del usuario. • No hay una indicación que muestre una invitación a teclear una palabra en el área de entradas.
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico.
Manejo de Errores	Protección Contra los Errores Calidad de los Mensajes de Error	No existe nada que evite que el usuario ingrese algo que no esperamos. No cuenta con ningún mensaje de Error.

Continúa en la siguiente hoja

Tabla 4.2 – *Continuación*

Criterio	**Sub-Criterio**	**Descripción**
	Corrección de los Errores	La interfaz no permite al usuario regresar a un estado anterior una vez que se ha cometido un error.
Significado de Códigos		• No es claro que significa el signo de interrogación. • No es claro lo que el botón "Cancelar" hace.

Tabla 4.2: Tabla de análisis de la interfaz de selección de tabla.

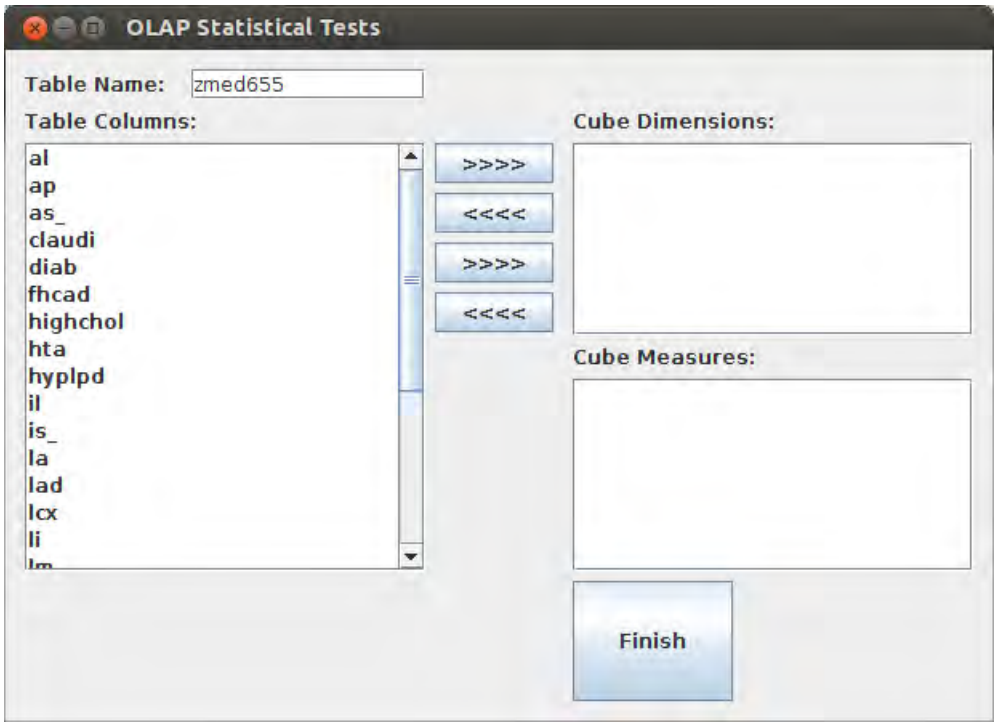

Figura 4.3: Interfaz de selección de dimensiones y medidas.

Criterio	Sub-Criterio	Descripción
Guía	Incitación	• No es claro que es lo que se espera que ingrese por parte del usuario. • No se guía al usuario para que seleccione los atributos dimensión y los atributos medidas. • No se indica el lugar en donde se encuentra dentro del sistema.
	Agrupación/ Distinción de Elementos	• Los botones para agregar o quitar atributos deberían ser de diferente tipo al de "Finish" ya que pertenecen a diferente clase. • Los botones de agregar atributos medida deberían separados de los que agregan dimensiones.
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico.
Manejo de Errores	Protección Contra los Errores	El campo de texto con el nombre de la tabla debería estar deshabilitado ya que no se espera que el usuario lo cambie.
	Calidad de los Mensajes de Error	No existen mensajes de error
	Corrección de los Errores	La interfaz no permite al usuario regresar a un estado anterior una vez que se ha cometido un error.
Significado de Códigos		• No es claro lo que el botón "Finish" hace, este puede interpretarse como terminar la aplicación o terminar la selección de atributos. • No es claro lo que los botones con etiquetas ">>>>" y "<<<<" hacen.

Tabla 4.3: Tabla de análisis de la interfaz de selección de dimensiones y medidas.

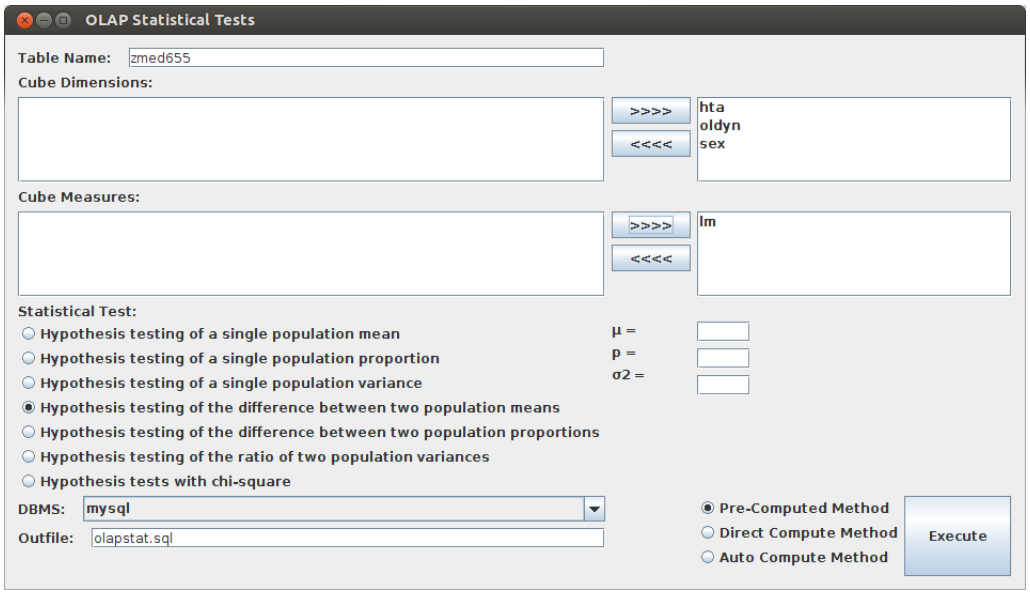

Figura 4.4: Interfaz de selección de parámetros para la prueba.

Criterio	**Sub-Criterio**	**Descripción**
Guía	Incitación	• No se indica que es lo que se espera que haga el usuario o para que se tiene que hacer un filtrado de atributos seleccionados. • No hay mensajes que le digan al usuario que seleccione opciones de análisis o tipo de método a usar.
	Agrupación/ Distinción de Elementos	Los botones de agregar o quitar atributos deberían ser de diferente tipo al de del botón "Finish" ya que pertenecen a diferente clase.
Carga de Trabajo	Concisión	Las etiquetas no son claras
Control Explícito	Control del usuario	No existe un mecanismo que permita regresar
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico

Continúa en la siguiente hoja

Tabla 4.4 – *Continuación*

Criterio	**Sub-Criterio**	**Descripción**
Manejo de Errores	Protección Contra los Errores	• El campo de texto con el nombre de la tabla debería estar deshabilitado ya que no se espera que el usuario lo cambie. • El usuario no debería poder cambiar el tipo de DBMS ya que puede que la tabla no exista en otro manejador. • El usuario bien podría no seleccionar ningún atributo. • El usuario podría ingresar caracteres no esperados o no permitidos en las cajas de texto.
	Calidad de los Mensajes de Error	No se cuenta con ningún mensaje de error.
	Corrección de los Errores	No permite que el usuario regrese a un estado anterior.
Significado de Códigos		• No es claro lo que el botón "Finish"hace. • No es claro lo que los botones con etiquetas ">>>>" y "<<<<" hacen.

Tabla 4.4: Tabla de análisis de la interfaz de selección de parámetros.

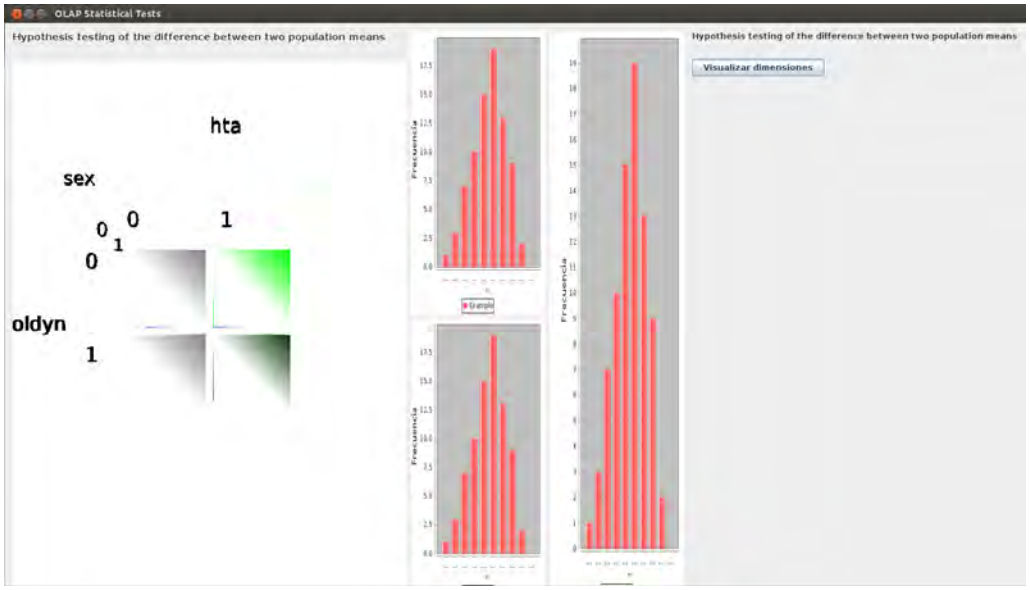

Figura 4.5: Interfaz de visualización del cubo OLAP.

Criterio	**Sub-Criterio**	**Descripción**
Guía	Incitación	No hay nada que indique al usuario que es lo que puede hacer o sobre la interpretación de resultados.
	Agrupación/ Distinción de Elementos	No hay una buena distribución de los elementos.
	Legibilidad	No es posible una buena legibilidad debido al tamaño de los textos.
Control Explícito	Control del Usuario	No existe ningún mecanismo que le permita al usuario regresar a la interfaz de selección de parámetros.
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico
Significado de Códigos		No es claro lo que el botón "Finish" hace.

Tabla 4.5: Tabla de análisis de la interfaz de visualización del cubo OLAP.

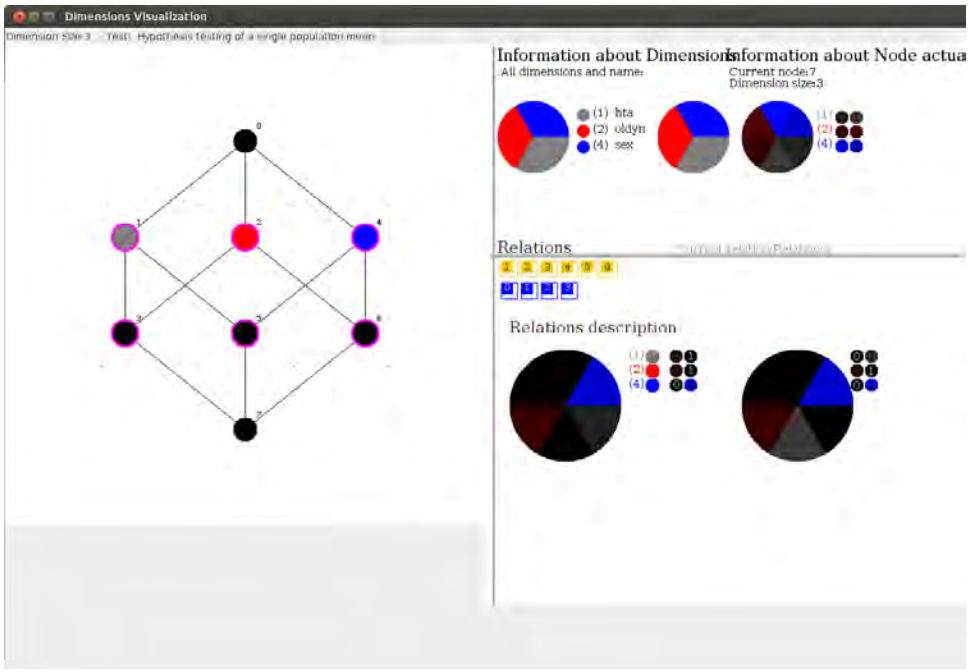

Figura 4.6: Interfaz de visualización en forma de retícula.

Criterio	Sub-Criterio	Descripción
Guía	Incitación	No hay nada que indique al usuario que es lo que puede hacer o sobre la interpretación de resultados.
	Legibilidad	No presenta buena legibilidad ya que las etiquetas de las gráficas de pastel de enciman.
Control Explícito	Control del Usuario	No existe ningún mecanismo que le permita al usuario regresar a la interfaz de selección de parámetros.
Adaptabilidad	Flexibilidad	No se permite que la interfaz se adapte a un usuario específico

Tabla 4.6: Tabla de análisis de la interfaz de visualización en forma de lattice.

Además de la evaluación de cada interfaz por separado, la herramienta en general presenta problemas de brevedad en su apartado de acciones mínimas ya que se necesita navegar a través de muchas interfaces para poder realizar una tarea. Para el caso de errores que suceden cuando el usuario ingresa una opción no solicitada o incorrecta el programa simplemente no hace nada o se cierra, esta pantalla no se muestra.

4.4. Evaluación de Interfaces con Usuarios

La evaluación de interfaces sin usuarios y el uso de los criterios ergonómicos comprende una herramienta importante y económica para la evaluación de la usabilidad, como se establece en secciones anteriores, pero aún es necesario realizar pruebas con usuarios reales, y que estos realicen tareas reales con la herramienta. Con estas pruebas podemos resolver problemas que pasamos por alto durante las etapas de diseño y evaluación sin usuarios, muchos de estos problemas ocurren ya que asumimos cosas como obvias o por conocer demasiado nuestro sistema, a continuación se describe la metodología a seguir durante las pruebas con usuarios y los elementos necesarios para su realización así como la descripción del laboratorio en el cual se realizó la prueba con usuarios de la herramienta con la interfaz propuesta, la cual será explicada más adelante.

4.4.1. Laboratorio

Existe una gran variedad de configuraciones posibles para este tipo de laboratorios [30], se escoge la configuración llamada *Observación Electrónica Simple*. Esta configuración permite a los observadores estar en un cuarto de observación, anexo al cuarto de prueba, en el cual pueden observar el desarrollo de la prueba pero no pueden interactuar con el moderador de la misma, en la Figura 4.7 se muestra un ejemplo de esta configuración.

Una de las ventajas principales de esta configuración es que, el usuario no tiene porqué saber de antemano que está siendo observado, ni que sus acciones y su persona serán grabadas, esto representa un punto importante porque no introduce

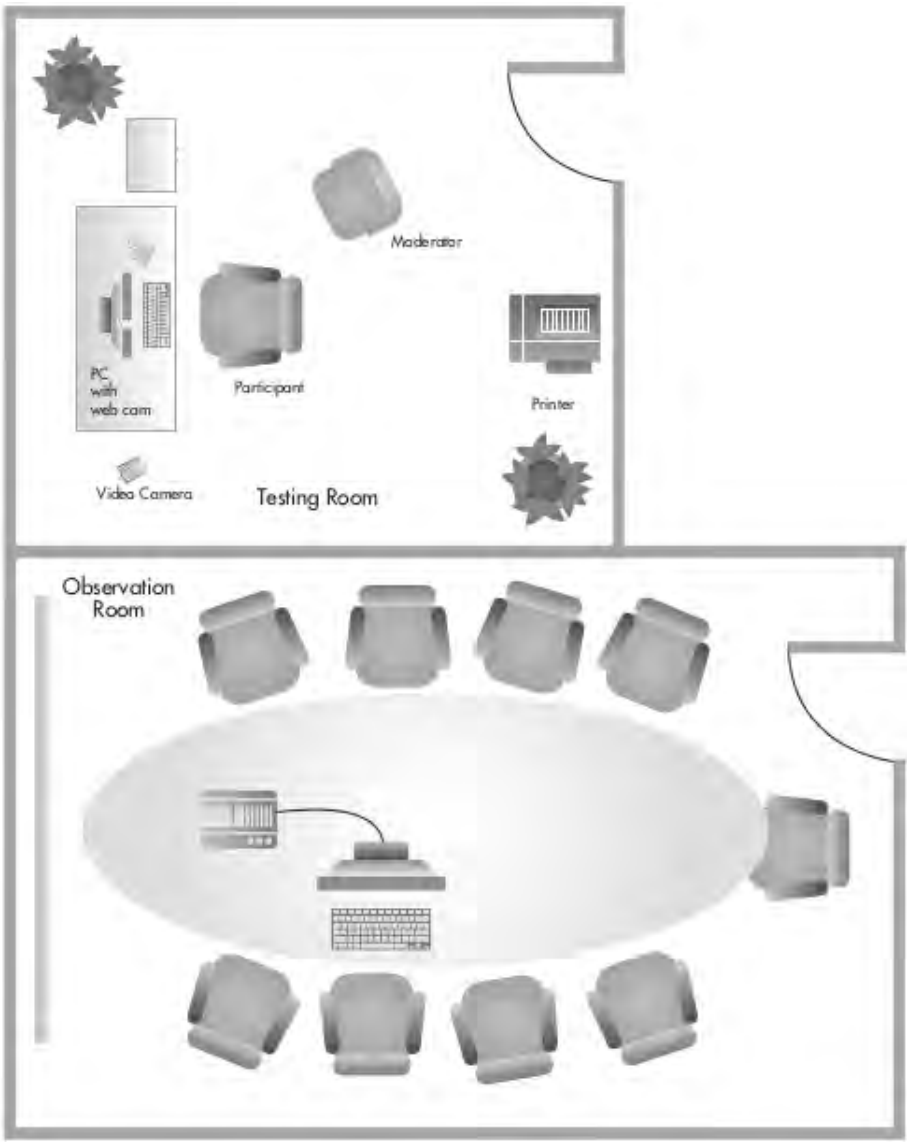

Figura 4.7: Ejemplo del laboratorio de pruebas con usuario, tomado de [30].

nerviosismo al usuario. Otra ventaja es que el resultado de la prueba no quedará sesgado por la influencia del observador ya que el moderador es el único que interactúa con el usuario, proporcionándole todo lo que éste necesite.

La única desventaja que presenta es que el comportamiento del moderador de la prueba puede afectar en forma negativa al resultado de la misma. El moderador podría proporcionar ayuda al usuario o incomodar a éste con algunos comentarios o con su comportamiento durante la realización de la prueba.

4.4.2. Etapas de la Prueba

A continuación, se hace una descripción de las etapas en las que se divide las pruebas con usuarios y en qué consiste cada una de ellas:

- Elaboración del plan de pruebas.
- Elección de los participantes.
- Creación de los Artefactos.
- Realización de la prueba.
- Despedir al usuario.
- Analizar la prueba.

Elaboración del Plan de Pruebas

Cuando se requiere una prueba con usuarios es necesario contar con ciertas hipótesis acerca del prototipo actual las cuales guiarán la prueba, los documentos requeridos que conforman el plan de pruebas, disponibles en el Apéndice A, son:

Protocolo de Bienvenida: Este documento es muy importante ya que es el texto que lee el moderador de la prueba, el cual será igual para todos los participantes, y contendrá la propia presentación del moderador, instrucciones generales y hará hincapié en que solo se está evaluando al prototipo y no al usuario.

Actividades del Plan de Pruebas: Son las tareas que se le solicitarán al usuario que realice. Estas tareas tienen que estar dirigidas a verificar la validez de las hipótesis planteadas anteriormente.

Cuestionario de Perfil del Usuario: Este cuestionario es una serie de preguntas, las cuales ayudan a asegurar que los usuarios cumplen con el perfil deseado para la prueba.

Cuestionario de Usabilidad: Es un cuestionario, el cual permite capturar la experiencia subjetiva del usuario para con el sistema.

Elección de Usuarios

La selección de los participantes, cuyos conocimientos y habilidades son representativos de los usuarios para quienes se desarrolla un producto, es crucial en el

proceso de prueba dado que los resultados de las mismas serán válidos solamente si los participantes son o serán usuarios típicos del producto. De esta manera, si se realizan pruebas con usuarios no representativos, no importara que tanto esfuerzo se ponga en la prueba o que tan bien esté diseñada, los resultados serán cuestionables o sin valor.

El usuario final es a menudo difícil de identificar y de describir, pero deben realizarse las pruebas con personas que representen una aproximación cercana a lo que nosotros pensamos como nuestro usuario ideal.

Creación de los Artefactos

Los artefactos no son lo mismo que los materiales o los documentos que se usarán en la prueba. Suponga que queremos que nuestro usuario ingrese información en diferentes secciones o que añada un texto, no sería adecuado si el moderador le dictase todos los datos. Además recordemos que estamos tratando de simular condiciones reales. Una forma para solucionar esto es darle al usuario información impresa. Este material impreso forman parte del tipo de material que se podría necesitar para la prueba. Podría ser que el usuario necesite papel, discos vírgenes, etc.

Realización de la prueba

En una prueba clásica, como se menciona en [3], ocurren los siguientes eventos:

- El usuario previamente pudo haber contestado el cuestionario de perfil del usuario.
- El moderador recibe al usuario, le explica brevemente en lo que consistirá la prueba y lo instala en el laboratorio.
- Cuando el usuario está listo y se sienta preparado, el moderador o entrevistador lee el protocolo de bienvenida y le indica al usuario las tareas a realizar.

Despedir al usuario

Al término de la prueba, se le da las gracias al usuario y se le entrega el cuestionario de usabilidad. En este punto el moderador ya podrá responder a todas las preguntas que no podía responder durante la prueba o aclarar dudas que tenga el usuario acerca de la misma.

Analizar la prueba

Ya que se han realizado las pruebas con todos los usuarios, se procede a la revisión por parte del o de los observadores de todo el material recabado durante las pruebas y analizar las respuestas de los usuarios a los cuestionarios. Con esto se hace un reporte de la evaluación a la interfaz incluyendo qué problemas fueron detectados y los comentarios por parte de los usuarios.

En este capítulo se resaltó la importancia del diseño centrado en el usuario y de las evaluaciones de interfaces con y sin usuarios, además, se realizó un análisis de los principales problemas en términos de usabilidad que presenta la herramienta base.

CAPÍTULO 5

Herramienta OLAP

La integración de las herramientas de análisis de datos con los sistemas manejadores de base de datos analíticos resulta un punto importante ya que como vimos en capítulos anteriores nos permiten optimizar tiempos, manejar grandes volúmenes de datos y un almacenamiento eficiente.

En este capítulo se presentan las mejoras que se realizaron a la herramienta base. Primero se presentan las mejoras en su parte funcional con la descripción del procedimiento e implementación de las consultas para la realización de las diferentes pruebas estadísticas definidas en el Capítulo 2 ya que esto potenciará el uso de la aplicación en ambientes analíticos y permitirá que la misma pueda analizar grandes volúmenes de información con un mejor tiempo de respuesta. Aunque se realizó la implementación de las diferentes pruebas estadísticas tanto en *Vertica* como en *SciDB*, sólo se muestra el caso de implementación con arreglos en *SciDB* debido a que *Vertica* usa SQL como lenguaje de manipulación de datos y que la herramienta base hace uso de sentencias del estándar de SQL para la realización de las pruebas estadísticas, el presentarlas sería hasta cierto punto redundante. Posteriormente se presenta la propuesta de interfaz siguiendo los criterios ergonómicos y contemplando los problemas detectados durante la evaluación de la interfaz sin usuarios detallado en el capítulo anterior.

5.1. Consultas AQL para la Generación de Cubos

Como se revisó en el Capítulo 2 los almacenes de datos usan modelos de datos multidimensionales los cuales típicamente están conformados por hechos y dimensiones. Los hechos contienen columnas numéricas a las cuales se les puede aplicar funciones de agregación llamadas medidas. En los ambientes analíticos los usuarios necesitan analizar los hechos agregados en varios niveles de abstracción. La consulta básica que obtiene hechos con base a una granularidad deseada por una lista de categorías es conocida como una vista de cubo. Una vista de cubo puede definirse, utilizando una función de agregación, como lo muestra la consulta siguiente.

```
#SQL
SELECT <Dimensión(es)>, <Función(Medida(s))>
FROM <Tabla>
GROUP BY <Dimensión(es)>;
```

En SQL se puede definir un cubo de datos como el conjunto de todas la posibles vistas del cubo definidas sobre una lista de dimensiones, una tabla base, y las medidas agregadas. En el contexto de un cubo de datos, se puede denotar una vista de cubo simplemente como CV $[G]$, donde G es una granularidad [12]. Como se revisó en capítulos anteriores, las dimensiones de un arreglo multidimensional forman una dependencia funcional con los atributos del mismo, es por esto que no es necesaria la definición de las dimensiones como parámetro de la sentencia "SELECT"de AQL, como lo muestra la siguiente consulta.

```
#AQL
SELECT <Función(Medida(s))>
FROM <Arreglo>
GROUP BY <Dimensión(es)>;
```

Por otra parte, en *SciDB* no se puede asignar el resultado de una consulta a un arreglo que no está definido, a diferencia de manejadores de base de datos relacionales con la sentencia "SELECT ... INTO ..." de SQL, en este caso primero se define la estructura del arreglo y después se define la consulta con la que será llenado. A continuación, se detallan las consultas y las definiciones de los arreglos involucrados en la realización de las pruebas de hipótesis, donde M_i representa las medidas, D_i las dimensiones con $i \in \{1, 2, \ldots, N\}$, V es el valor contra el que se está haciendo

la comparación y A es el nombre del arreglo original que contiene la información. Los elementos en las consultas que se encuentren entre <> serán sustituidos por las dimensiones, medidas o parámetros seleccionados por el usuario, por ejemplo $<D_1>$ será sustituido por el nombre de la primera dimensión seleccionada para el análisis.

5.1.1. Arreglo base

Existe un arreglo base del cual parten las pruebas estadísticas, este arreglo contiene las estadísticas reales de las poblaciones a las cuales se les aplicará la prueba estadística, las poblaciones son obtenidas mediante la agrupación de los valores de las dimensiones. El arreglo base es nombrado tempMSV. En este arreglo se almacenan las estadísticas necesarias como la media, la varianza y la desviación estándar de cada medida (M_i) para cada población por dimensión seleccionada (D_i). Los parámetros con el carácter "?" son los valores correspondientes a la configuración de las dimensiones del arreglo original (A) el cual contiene la información que será analizada, estos valores son obtenidos mediante una consulta al catálogo de datos que se encuentra en una base de datos de PostgreSQL, esto es importante ya que la estructura de los arreglos, los cuales contendrán información proveniente del arreglo original, deben tener la misma estructura. La definición general del arreglo tempMSV es la siguiente.

```
CREATE ARRAY tempMSV <
    N :uint64 ,
    MEAN_<M1> :double ,
    VARIANCE_<M1> :double ,
    STD_<M1> :double ,
    ... ,
    MEAN_<MN> :double ,
    VARIANCE_<MN>:double ,
    STD_<MN> :double >
[<D1>=?:?,?,?,
 ... ,
 <DN>=?:?,?,?];
```

Este arreglo es llenado a partir de la siguiente sentencia AQL que hace uso de funciones nativas del manejador, a diferencia de la herramienta base en la cual los cálculos se hacen con sumas y multiplicaciones, para el cálculo de las estadísticas

como se muestra a continuación.

```
INSERT INTO tempMSV
SELECT * FROM
    AGGREGATE(<A>,
    SUM(1) as N,
    AVG(<M1>) as MEAN_<M1>,
    VAR(<M1>) as VARIANCE_<M1>,
    STDEV(<M1>) as STD_<M1>,
    ...,
    AVG(<MN>) as MEAN_<MN>,
    VAR(<MN>) as VARIANCE_<MN>,
    STDEV(<MN>) as STD_<MN>
    <D1>, ..., <DN>);
```

Una vez que se ha definido el arreglo base se procede a la elaboración de las pruebas estadísticas. A continuación, se detalla las consultas y los arreglos creados para cada una de las pruebas.

5.1.2. Media de una sola población

Después de haber creado el arreglo base se crea un nuevo arreglo nombrado <A>_TempZ en el cual se almacenará los valores para el estadístico de prueba en cuestión, donde los atributos del arreglo contendrán los valores para cada población analizada.

```
CREATE ARRAY <A>_TempZ <
    ZTest_<M>:double,
    N :uint64 >
[<D1>=?:?,?,?,
 <D2>=?:?,?,?,
 <D3>=?:?,?,? ];
```

En este arreglo se guardan los valores correspondientes a los valores del estadístico de prueba z calculados a partir del arreglo tempMSV. La siguiente consulta muestra un ejemplo del llenado del arreglo <A>_TempZ donde <V> es el valor contra el que se está haciendo la comparación y el cual es ingresado desde la aplicación,

las dimensiones son tomadas a partir de las dimensiones del arreglo tempMSV, por último, en la cláusula "WHERE" se filtran aquellas poblaciones donde existan 25 o más integrantes, esto debido a la suposición de una distribución normal.

```
INSERT INTO <A>_TempZ
SELECT
    ((MEAN_<M> - <V>)/(STD_<M> / sqrt(N) )) as ZTest_<M>,
    N
FROM   tempMSV
WHERE ( N >=25 );
```

Para finalizar se crea el arreglo <*A*>_Final mediante la siguiente consulta la cual hace uso de los resultados del arreglo anterior, además se incluye un atributo extra que nos ayudará a tomar una decisión acerca de la población analizada. Dependiendo del resultado del estadístico z se compara con valores extraídos de la tabla de distribución para z. En las poblaciones donde se asigne un valor de "3" en el atributo Pvalue_<*M*> serán las poblaciones donde se rechazará la hipótesis nula y se aceptara la hipótesis alternativa.

```
INSERT INTO <A>_Final
SELECT
    iif(abs(ZTest_<M>) < 1.88, '0',
    iif(abs(ZTest_<M>) < 1.96, '1',
    iif(abs(ZTest_<M>) < 2.04, '2', '3'))) as Pvalue_<M>,
    ZTest_<M>,
    N
FROM <A>_TempZ;
```

5.1.3. Diferencia entre medias de dos poblaciones

Para esta prueba se crea un arreglo el cual contendrá las estadísticas de aquellas poblaciones que difieren en una sola dimensión, tomando como base el arreglo tempMSV, la definición de este arreglo es la siguiente.

```
CREATE ARRAY tempCompare <
    N1:uint64 NULL,
    MEAN<M>1:double NULL,
```

```
    VARIANCE_<M>1:double NULL,
    STD_<M>1: double NULL,
    N2:uint64 NULL,
    MEAN_<M>2:double NULL,
    VARIANCE_<M>2:double NULL,
    STD_<M>2: double NULL>
[<D1>1=0:?,?,?, ···, <DN>1=?:?,?,?,
<D1>2=0:?,?,?, ···, <DN>2=?:?,?,?];
```

Para obtener la información de este arreglo se toman dos instancias del arreglo tempMSV (representadas por a1 y a2) tomando solo aquellas poblaciones de difieren en una sola dimensión, esto se logra fijando el valor para una de las dimensiones de una de las instancias y forzando que esta misma dimensión sea diferente en la otra instancia. Este procedimiento es realizado para cada una de las dimensiones seleccionadas por el usuario, la siguiente consulta muestra un ejemplo del procedimiento descrito.

```
INSERT INTO tempCompare
SELECT
    a1.N AS N1
    a1.MEAN_<M> as MEAN_<M>1
    a1.VARIANCE_<M> as VARIANCE_<M>1
    a1.STD_<M> as STD_<M>1
    a2.N AS N2
    a2.MEAN_<M> as MEAN_<M>2
    a2.VARIANCE_<M> as VARIANCE_<M>2
    a2.STD_<M> as STD_<M>2
FROM cross join (tempMSV as a1,tempMSV as a2)
WHERE (
(a1.<D1> <> a2.<D1> AND a1.<D1>=0 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1> <> a2.<D1> AND a1.<D1>=1 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=0
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=1
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
    AND a1.<D3>=0)
OR (a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
```

```
    AND a1.<D3>=1));
```

Al igual que la prueba anterior se crea un arreglo nombrado <*A*>_TempZ el cual almacenará los valores del estadístico de prueba para cada una de las poblaciones analizadas, este valor es el que nos servirá para poder tomar una decisión acerca del rechazo o aceptación de la hipótesis establecida ($H_0 : \mu_1 = \mu_2$). Los atributos con sufijo 1 son los pertenecientes a la primera población y los atributos con sufijo 2 son los pertenecientes a la segunda, la siguiente consulta muestra un ejemplo del cálculo del estadístico de prueba z.

```
INSERT INTO <A>_TempZ
SELECT
    (MEAN<M>1-MEAN<M>2)/(sqrt((VARIANCE<M>1/N1)+
    (VARIANCE<M>2/N2))) as ZTest_<M>,
    N1,
    <D1>1, <D2>1, <D3>1,
    N2,
    <D1>2, <D2>2, <D3>2
FROM tempCompare
WHERE ( N1>=25 AND N2>=25 );
```

Por último se crea el arreglo llamado <*A*>_Final el cual almacenará los valores finales de la prueba, es un arreglo multidimensional de las poblaciones comparadas para las dimensiones seleccionadas por el usuario, los atributos de este arreglo corresponden al número de elementos de cada población (N1 y N2), el valor del estadístico de prueba y la decisión sobre la hipótesis establecida.

```
CREATE ARRAY <A>_Final<
    PValue_<M>:string NULL,
    ZTest_<M> :double NULL,
    N1:uint64 NULL,
    N2:uint64 NULL>
[<D1>1=?:?,?,?,
<D2>1=?:?,?,?,
<D3>1=?:?,?,?,
<D1>2=?:?,?,?,
<D2>2=?:?,?,?,
<D3>2=?:?,?,? ];
```

Este arreglo es llenado a partir de la siguiente consulta, en donde dependiendo del resultado se compara con los valores extraídos de la tabla de distribución para z de acuerdo al grado de significancia, para aquellas poblaciones con un valor "3" serán donde la hipótesis nula ha sido rechazada y es donde se ha encontrado una relación significativa entre las poblaciones.

```
INSERT INTO <A>_Final
SELECT
      iif(abs(ZTest_<M>)<1.88,'0',
      iif(abs(ZTest_<M>)<1.96,'1',
      iif(abs(ZTest_<M>)<2.04,'2','3'))) AS PValue_<M>,
ZTest_<M>,
N1,
<D1>1, <D2>1, <D3>1,
N2,
<D1>2, <D2>2, <D3>2
FROM <A>_TempZ;
```

5.1.4. Proporción de una población

Como se revisó en el Capítulo 2, las pruebas de hipótesis que involucran proporciones solo toman en cuenta los totales, debido a esto se hace uso de un arreglo base (nombrado de igual manera tempMSV) con una estructura distinta ya que no es necesario el cálculo de la media, la varianza y la desviación estándar. Este arreglo contendrá los totales para cada población y un atributo (NT) que representa la suma de los totales de las poblaciones, dicha estructura es la siguiente.

```
CREATE ARRAY tempMSV<
      N :uint64 NULL,
      NT :int64 NULL >
[<D1>=?:?,?,?,
 <D2>=?:?,?,?,
 ...
 <DN>=?:?,?,?];
```

Este arreglo base, el cual será utilizado para esta prueba y para la siguiente, es llenado a partir de la siguiente consulta.

```
INSERT INTO tempMSV
SELECT
      N,
      1 as NT
FROM aggregate(<A>,
     COUNT(*) as N,
     <D1>,
     <D2>,
     <D3>);
```

El atributo NT es inicialmente asignado a 1 y después es actualizado con las siguientes consultas, en donde el valor de la sentencia UPDATE (<*Total*>) es el valor obtenido de la consulta SELECT.

```
SELECT
     NT as <Total>
FROM aggregate(tempMSV, SUM(N) as NT);

UPDATE tempMSV
     SET NT = <Total>;
```

De forma similar a las pruebas anteriores se crea el arreglo <*A*>_TempZ donde se asignan los valores del estadístico obtenido, siendo éste parametrizado por el valor <*P*>, el cual representa el parámetro (proporción) ingresado por el usuario desde la aplicación.

```
INSERT INTO <A>_TempZ
SELECT
     ((double(N)/double(NT)) - (<P>)) /
     sqrt((<P>) * (1 - (double(N)/double(NT)))) / (double(N)) ),
     N as N1
FROM tempMSV
WHERE ( N>=25 );
```

De igual manera se concluye con el llenado del arreglo multidimensional <*A*>_Final, con el cual hacemos las comparaciones necesarias para determinar si existe evidencia suficiente para rechazar o aceptar la hipótesis nula.

```
INSERT INTO <A>_Final
SELECT
     iif(abs(ZTest_<M>)<1.88, '0',
     iif(abs(ZTest_<M>)<1.96, '1',
     iif(abs(ZTest_<M>)<2.04, '2', '3'))) as Pvalue_<M>,
     ZTest_<M>,
     N1
FROM <A>_TempZ;
```

5.1.5. Diferencia entre las proporciones de dos poblaciones

Al igual que la prueba anterior se hace uso de un arreglo base distinto (definido en la sección anterior) y siguiendo el mismo procedimiento establecido en la prueba para la diferencia entre las medias de dos poblaciones, se crea un arreglo que contenga aquellas poblaciones que difieren en una dimensión el cual es nombrado tempComparePP, este arreglo es calculado a partir de dos instancias del arreglo tempMSV, la estructura de dicho arreglo es la siguiente.

```
CREATE ARRAY tempComparePP <
     N1:uint64 NULL,
     N2:uint64 NULL,
     NT1:int64 NULL,
     NT2:int64 NULL
[<D1>1=?:?,?,?,
 <D2>1=?:?,?,?,
 <D3>1=?:?,?,?,
 <D1>2=?:?,?,?,
 <D2>2=?:?,?,?,
 <D3>2=?:?,?,? ];
```

La información que se guardara en este arreglo se calcula a partir de la siguiente consulta. En esta consulta los valores dependen del dominio de las dimensiones elegidas por el usuario.

```
INSERT INTO tempComparePP
SELECT
     a1.N as N1,
     a2.N as N2,
```

```
    0 as NT1,
    0 as NT2
FROM  cross_join(tempMSV as a1 ,tempMSV as a2)
WHERE
((a1.<D1> <> a2.<D1> AND a1.<D1>=0 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1> <> a2.<D1> AND a1.<D1>=1 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=0
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=1
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
    AND a1.<D3>=0 )
OR ( a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
    AND t1.<D3>=1 ));
```

Una vez que se ha creado el arreglo que contiene aquellas poblaciones que difieren en una dimensión se actualizan los valores totales NT1 y NT2 de cada relación de poblaciones, estos totales fueron inicializados con 0 en la consulta anterior, el proceso de actualización es el siguiente. Primero se toman los valores (<$v1a$> y <$v1b$>) de la dimensión en la cual varían <D> y se obtiene los totales para cada una de ellos, los totales obtenidos de las consultas son nombrados <Na> y <Nb> y serán usados en la actualización de NT1 y NT2 dejando las demás dimensiones con sus respectivos valores.

```
SELECT
      SUM(N) as <Na>
FROM  tempMSV
WHERE (<D> = <v1a>);

SELECT
      SUM(N) as <Nb>
FROM  tempMSV
WHERE (<D> = <v1b>);

UPDATE
     tempComparePP
SET NT1 = <Na>, NT2 = <Nb>
```

```
WHERE <D1>1 = <v1a> AND <D2>1 = <v2> AND <D3>1 = <v3>
  AND <D1>2 = <v1b> AND <D2>2 = <v2> AND <D3>2 = <v3>;
```

Por último se crea los arreglos <A>_TempZ y <A>_Final al igual que las pruebas anteriores.

```
CREATE ARRAY <A>_TempZ <
        ZTest_<M>:double NULL,
        N1:uint64 NULL,
        N2:uint64 NULL>
 [<D1>1=?:?,?,?,
 <D2>1=?:?,?,?,
 <D3>1=?:?,?,?,
 <D1>2=?:?,?,?,
 <D2>2=?:?,?,?,
 <D3>2=?:?,?,?];
```

Este arreglo contendrá los valores del estadístico de prueba y en el cual se hace un filtrado de aquellas poblaciones que tengan más de 25 elementos debido al supuesto de distribución normal. La siguiente consulta muestra un ejemplo del llenado de este arreglo.

```
INSERT INTO <A>_TempZ
SELECT
 ((N1/double(NT1)) - (N2/double(NT2))) / sqrt( ((1.0/double(NT1)) +
 (1.0/double(NT2))) * ((N1+N2)/(double(NT1+NT2))) *
 (1.0-((N1+N2)/double((NT1+NT2)))) ) as ZTest_<M> ,
 N1,
 N2
FROM  tempComparePP
WHERE ( N1>=25 AND N2>=25 );
```

Una vez que tenemos el valor del estadístico de prueba se crea el arreglo <A>_Final el cual contendrá la decisión acerca de la hipótesis planteada.

```
CREATE ARRAY <A>_Final <
      PValue_<M>:string NULL,
      ZTest_<M>:double NULL,
      N1:uint64 NULL,
      N2:uint64 NULL>
```

```
[<D1>1=?:?,?,?,
<D2>1=?:?,?,?,
<D3>1=?:?,?,?,
<D1>2=?:?,?,?,
<D2>2=?:?,?,?,
<D3>2=?:?,?,?];
```

Este arreglo se llena con la siguiente consulta las celdas que tengan un valor de "3" en su atributo Pvalue_<M> será donde se rechace la hipótesis nula.

```
INSERT INTO <A>_Final
SELECT
        iif(abs(ZTest_<M>)<1.88, '0',
        iif(abs(ZTest_<M>)<1.96, '1',
        iif(abs(ZTest_<M>)<2.04, '2', '3'))) as Pvalue_<M>,
        ZTest_<M>,
        N1,
        N2
FROM <A>_TempZ;
```

5.1.6. Varianza de una sola población

Una vez creado el arreglo base se procede al llenado del arreglo <A>_TempZ, tomando en cuenta el parámetro (varianza) ingresado por el usuario para el cálculo del estadístico z, en donde <V> representa la varianza con la cual se está comparando.

```
INSERT INTO <A>_TempZ
SELECT
     ( ((N - 1) * (VARIANCE_<M>)) /<V>) as ZTest_<M>,
     N
FROM tempMSV
WHERE ( N>=25 );
```

Por último se hace el cálculo del arreglo <A>_Final, el cual es llenado por la consulta.

```
INSERT INTO <A>_Final
SELECT
    '' as PValue_<M>,
    ZTest_<M>,
    N1
FROM <A>_TempZ;
```

Debido a los grados de libertad involucrados en cada prueba de acuerdo al número de dimensiones que puede variar, se toman los datos del arreglo <A>_Final que ha sido llenado previamente y se hacen las debidas comparaciones haciendo uso de las siguientes consultas para determinar el resultado de la prueba.

```
SELECT
PValue_<M>,
ZTest_<M>,
N,
<D1>,
<D2>,
<D3>
FROM <A>_Final;

UPDATE <A>_Final
SET PValue_<M> = <V>
WHERE
<D1>1=<v1> AND <D2>1=<v2> AND <D3>1=<v3>;
```

5.1.7. Razón de las varianzas de dos poblaciones

Al igual que la prueba de la proporción de dos poblaciones, se crean dos atributos los cuales representan a cada una de las muestras N1 y N2. Para esta prueba se crea un arreglo que contiene las poblaciones que difieren en una dimensión.

```
INSERT INTO tempCompare
SELECT
      a1.N as N1,
      a2.N as N2,
      a1.VARIANCE_<M> as VARIANCE_<M>1,
      a2.VARIANCE_<M> as VARIANCE_<M>2
FROM cross join (tempMSV as a1 ,tempMSV as a2)
```

```
WHERE
((a1.<D1> <> a2.<D1> AND a1.<D1>=0 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1> <> a2.<D1> AND a1.<D1>=1 AND a1.<D2>=a2.<D2>
    AND a1.<D3>=a2.<D3>)
OR (a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=0
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1>=a2.<D1> AND a1.<D2> <> a2.<D2> AND a1.<D2>=1
    AND a1.<D3>=a2.<D3>)
OR ( a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
    AND a1.<D3>=0 )
OR ( a1.<D1>=a2.<D1> AND a1.<D2>=a2.<D2> AND a1.<D3> <> a2.<D3>
    AND t1.<D3>=1 ));
```

El cual sirve como base para la creación del arreglo <*A*>_TempZ.

```
CREATE ARRAY <A>_TempZ <
    ZTest_<M>:double NULL,
    N1:uint64 NULL,
    N2:uint64 NULL>
[<D1>1=?:?,?,?,
 <D2>1=?:?,?,?,
 <D3>1=?:?,?,?,
 <D1>2=?:?,?,?,
 <D2>2=?:?,?,?,
 <D3>2=?:?,?,? ];
```

Debido a que en esta prueba la varianza de la población con mayor valor tiene que ser la que se coloque en el lugar del numerador para el cálculo del estadístico z, se hace una comparación de las varianzas mediante la cláusula iif, la siguiente consulta muestra un ejemplo del llenado de este arreglo.

```
INSERT INTO <A>_TempZ
SELECT
    iif(VARIANCE_<M>1 > VARIANCE_<M>2 ,
        VARIANCE_<M>1 / VARIANCE_<M>2 ,
        VARIANCE_<M>2 / VARIANCE_<M>1 ) AS Ztest_<M>,
    iif(VARIANCE_<M>1 > VARIANCE_<M>2 , N1, N2) as N1,
    N2
FROM tempCompare
```

```
WHERE ( N1>=25 AND N2>=25 );
```

Por último se crea el arreglo <A>_Final el cual es llenado a partir de la siguiente consulta.

```
INSERT INTO <A>_Final
SELECT
    '' as PValue_<M>,
    ZTest_<M>,
    N1,
    N2
FROM <A>_TempZ;
```

Al igual que para la prueba anterior, se realiza un proceso de consulta del valor del estadístico de prueba y se actualiza el valor del atributo PValue_<M> con la decisión acerca de la hipótesis establecida, esto se debe al grado de libertad el cual puede variar de acuerdo al número de dimensiones seleccionadas.

```
SELECT
PValue_<M>,
ZTest_<M>,
N1,
<D1>1,
<D2>1,
<D3>1,
N2,
<D1>2,
<D2>2,
<D3>2
FROM <A>_Final;

UPDATE <A>_Final
SET PValue_<M> = <V>
WHERE
<D1>1=<v1a> AND <D2>1=<v2a> AND <D3>1=<v3a> AND
<D1>2=<v1b> AND <D2>2=<v2b> AND <D3>2=<v3b>;
```

5.1.8. Chi-cuadrada (prueba de independencia)

Para esta prueba se crea un arreglo nombrado tempCompare<D_1><D_2> donde <D_1> y <D_2> son un par de dimensiones seleccionadas desde la aplicación y son aquellas dimensiones que están siendo analizadas. Este procedimiento se sigue para cada combinación de dimensiones (de dos en dos) del conjunto de todas las dimensiones seleccionadas para saber si éstas guardan alguna relación.

```
CREATE ARRAY tempCompare<D1><D2> <
        N:uint64 NULL,
        T1:int64 NULL,
        T2:int64 NULL,
        FE:double NULL>
[<D1>=?:?,?,?,
 <D2>=?:?,?,?];
```

En este arreglo FE será el atributo que nos sirva para almacenar las frecuencias esperadas de la prueba. Posteriormente se calcula el total y la frecuencias esperadas con las siguientes consultas.

```
SELECT
    COUNT(1) as <Total>
FROM <A>
WHERE (<D1> is NOT NULL AND <D2> is NOT NULL );
```

Una vez que se ha obtenido el total se procede la llenado del arreglo nombrado tempCompare<D_1><D_2> con la siguiente consulta donde <$Total$> es el valor obtenido de la consulta anterior.

```
INSERT INTO tempCompare<D1><D2>
SELECT
        N,
        int64(T1),
        int64(T2),
        (double(T1) * double(T2))/<Total> as FE
FROM cross_join(
        cross_join(
            aggregate(<A>, COUNT(*) as N, <D1>, <D2>) as a1,
            aggregate(<A>, COUNT(*) as T1, <D1>) as a2,
```

```
        a1.<D1>, a2.<D1> ) as t<D1>,
    aggregate(<A>, COUNT(*) as T2, <D2>) as a3,
    t<D1>.<D2>, a3.<D2>);
```

Por último, al igual que las otras pruebas, se crea el arreglo <A>_Final el cual contendrá los resultados finales de la prueba y la decisión acerca de la hipótesis establecida (H_0 : Las variables son independiente).

```
INSERT INTO <A>_Final<D1><D2>
SELECT
        iif(abs(ZTest_<M>)<4.709,'0',
        iif(abs(ZTest_<M>)<5.024,'1',
        iif(abs(ZTest_<M>)<5.412,'2', '3'))) as PValue_<M>,
        ZTest_<M>,
        N1, <D1>1, <D2>1,
        N2, <D1>2, <D2>2
FROM (SELECT
        SUM((N–FE)*(N–FE)/FE) as ZTest_<M>
      FROM tempCompare<D1><D2>);
```

5.2. Interfaz Propuesta

Una vez que se han descrito las mejoras a la parte funcional de la aplicación se explicará el diseño de interfaz propuesto en el cual se tomaron como base los criterios ergonómicos y los problemas de usabilidad detectados durante la evaluación a la herramienta base. Esta propuesta será evaluada con pruebas de usabilidad con usuarios.

La interfaz de la herramienta optimizada se divide principalmente en dos interfaces, una interfaz para la selección de parámetros que va desde la selección del sistema manejador de base de datos, la tabla o arreglo a utilizar según sea el caso, la selección de las dimensiones y medidas hasta la selección de la prueba estadística y sus parámetros. La segunda interfaz principal que compone esta propuesta es la interfaz de visualización de resultados de las pruebas estadísticas (visualización del cubo OLAP y visualización de la retícula multidimensional). Además de estas dos

partes principales que componen la nueva interfaz se añadió otra interfaz que no está contemplada en la herramienta base, ésta es una interfaz para la configuración de parámetros de conexión con el sistema manejador de base de datos misma que puede ser accesada desde el menú File en la interfaz de inicio.

Interfaz de configuración de parámetros

Al iniciar la herramienta optimizada se presenta la interfaz de selección de parámetros (Figura 5.1), esta interfaz corrigen los problemas de las primeras cuatro interfaces, en la Figura 5.2 se muestra la distribución de los elementos en la interfaz de tal forma que elementos comunes están en la misma sección.

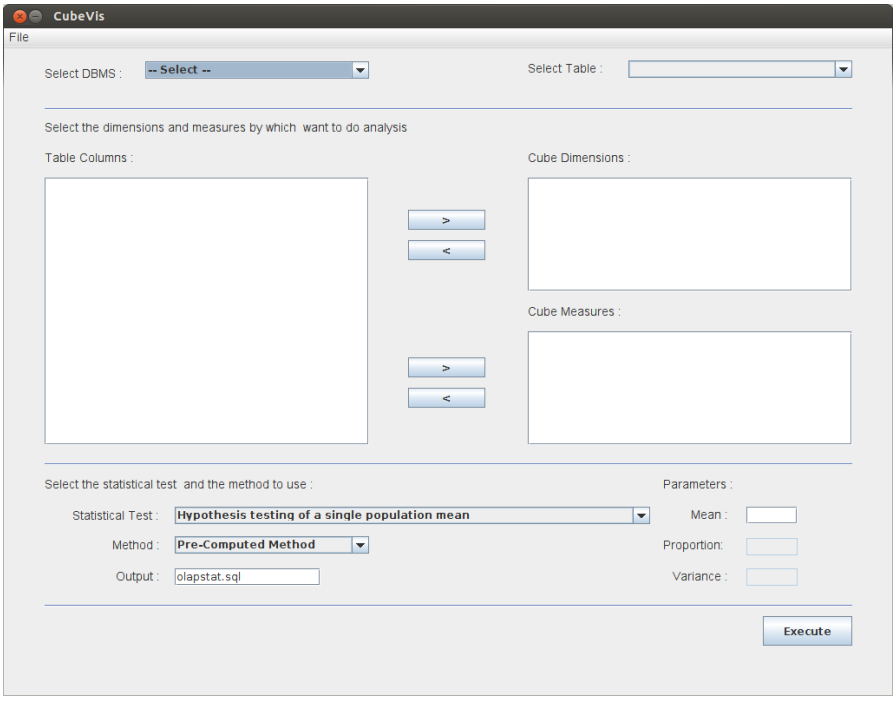

Figura 5.1: Interfaz de inicio.

Con estas mejoras también se resuelve el problema de brevedad (acciones mínimas), ya que no es necesario navegar entre varias interfaces para poder seleccionar y configurar los parámetros de la prueba estadística a realizar. Como ya se mencionó la optimización a la herramienta base con la implementación de bases de datos analíticas permite el análisis de grandes volúmenes de datos, es por esto que se agregó una barra de progreso al diseño para mostrar al usuario el progreso del análisis y evitar que el usuario sufra frustración por no saber si es que la aplicación sigue trabajando

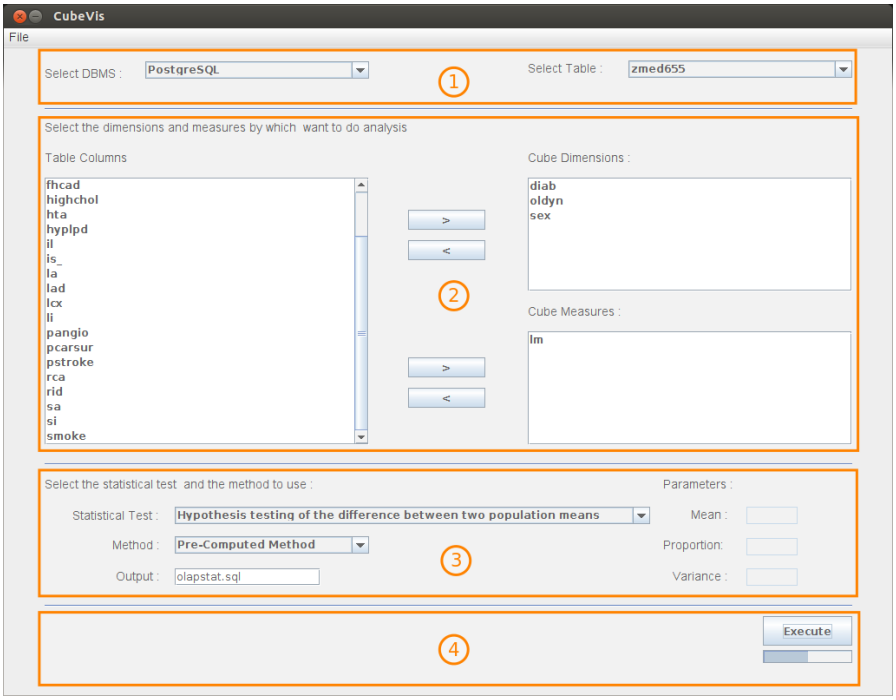

Figura 5.2: Interfaz de inicio, agrupación de elementos.

o es que tuvo problemas durante la ejecución, por otra parte el manejo de errores en la herramienta base no está contemplado por lo que la aplicación se cierra inesperadamente o tiene pausas indeterminadas cuando llega a ocurrir un error.

En la Figura 5.2, se muestran las secciones en la que se dividió la interfaz. En sección 1 están los elementos referentes a la selección del sistema manejador de base de datos (DBMS) y la selección de la tabla o arreglo, según sea el caso. En esta sección se emplearon listas desplegables con las cuales se evita que el usuario cometa algún error ya sea por la errónea selección del DBMS o por la selección de una tabla inexistente, también se incluyeron mensajes de error en caso de que no sea posible establecer conexión con el DBMS, Figura 5.3. En la sección 2 están los elementos referentes a la selección de las dimensiones y medidas, en sección 3 se encuentran aquellos elementos referentes a la elección de la prueba estadística y sus parámetros de configuración los cuales están solo habilitados en aquellas pruebas en las que éstos son requeridos evitando así que el usuario cometa errores. Por último, en sección 4 se encuentran los elementos referentes a mensajes de error por datos faltantes o por parámetros de no numéricos (Figura 5.4), el botón de ejecución de la prueba estadística y la barra de progreso que muestra el avance de la ejecución.

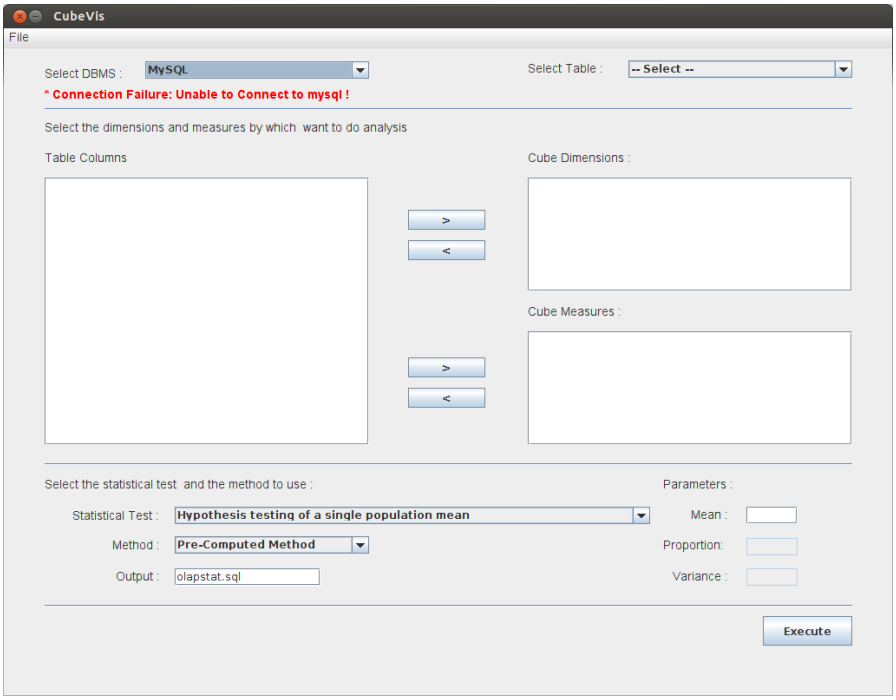

Figura 5.3: Interfaz de inicio con mensaje de error de conexión al sistema manejador de base de datos.

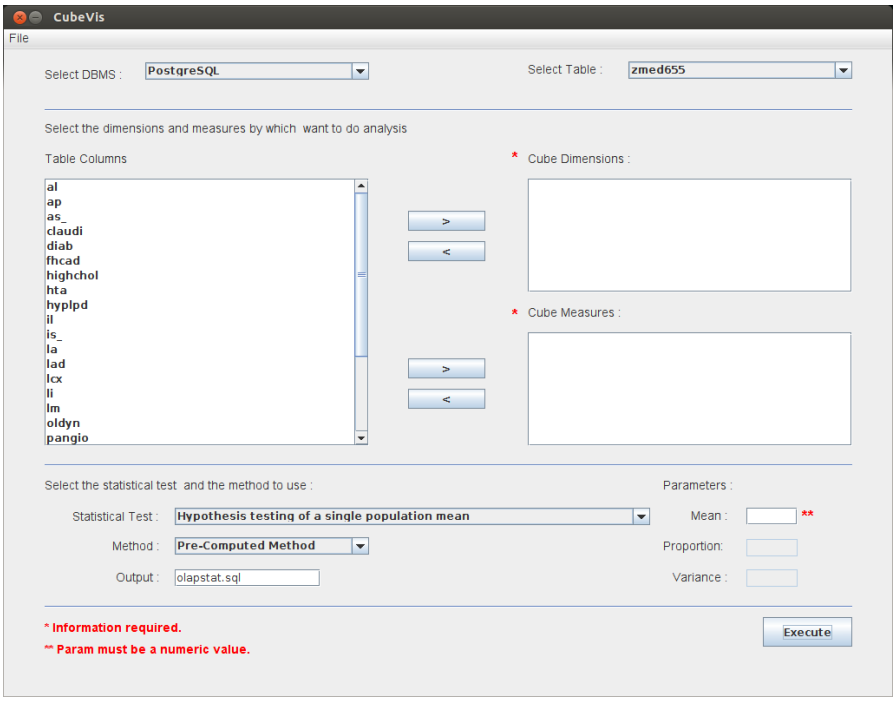

Figura 5.4: Interfaz de inicio con mensajes de error por parámetros faltantes o incorrectos.

Para la configuración de parámetros de conexión desde la interfaz de usuario se implementó una interfaz, la cual es accedida a través del menú File, en ella se solicitan los parámetros necesarios para la conexión al DBMS mostrando una configuración de URL base la cual el usuario solo tiene que cambiar de acuerdo a sus necesidades, Figura 5.5.

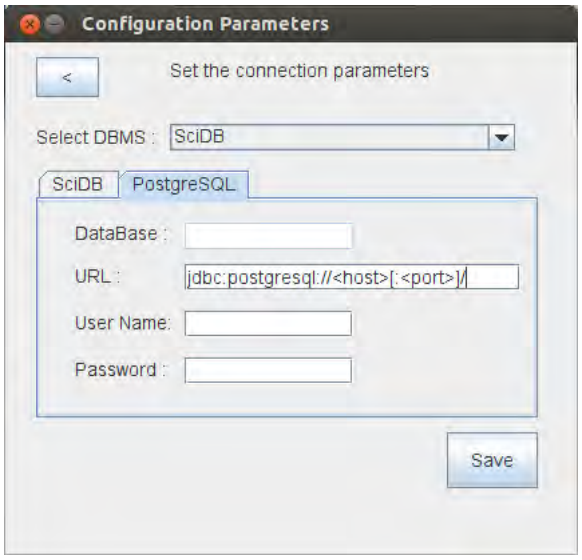

Figura 5.5: Interfaz de configuración de parámetros de conexión con el sistema manejador de base de datos.

Una vez que se ha configurado y ejecutado la prueba estadística se despliegan los resultados, primero se muestra la visualización del cubo OLAP (en la primera pestaña del panel de visualización, Figura 5.6) junto con gráficas de distribución de frecuencias. En esta interfaz también se hace el resumen de la prueba que se realizó, el número de dimensiones y tiempo de ejecución de la misma, también se agregó un mecanismo para poder regresar a la interfaz anterior, lo cual era uno de los problemas en la herramienta base, con lo que se puede cambiar los parámetros sin tener que salir de la aplicación. En la siguiente pestaña del panel se muestra la visualización en forma de retícula, este manejo de pestañas permite que el usuario no tenga que navegar entre interfaces para cambiar el modo de visualización de resultados, Figura 5.7.

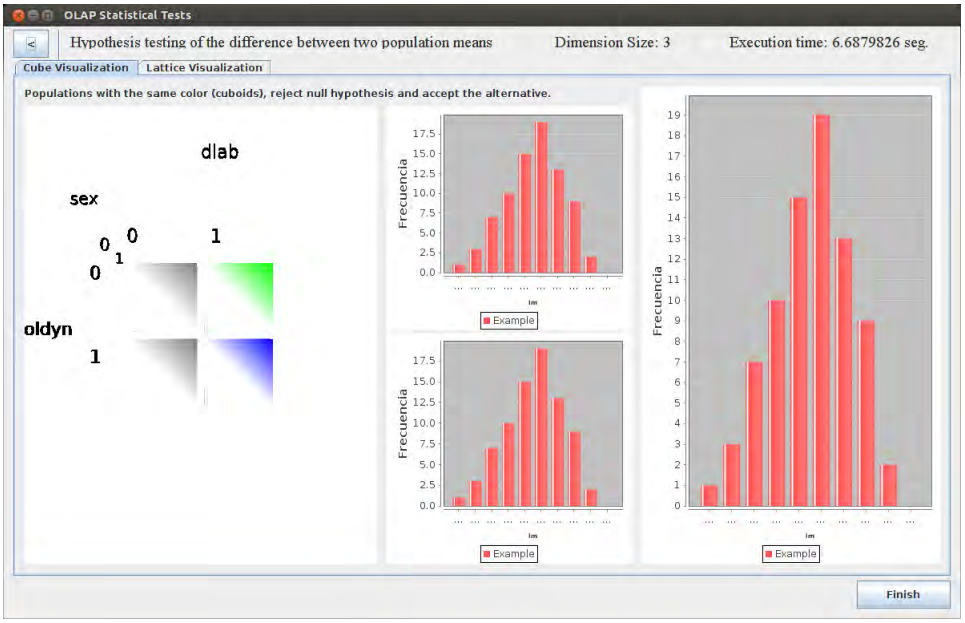

Figura 5.6: Interfaz de visualización de resultados, sección cubo OLAP.

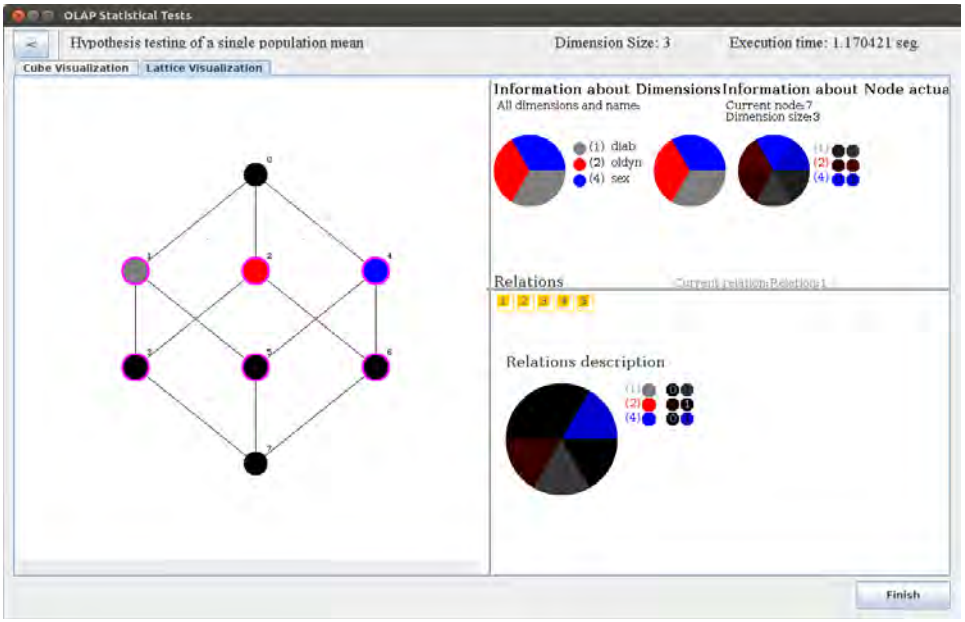

Figura 5.7: Interfaz de visualización de resultados, sección retícula multidimensional.

En este capítulo se presentaron las mejoras que se realizaron a la herramienta base tanto funcionales como en la usabilidad de su interfaz. Estas mejoras serán validadas en el siguiente capítulo con pruebas de tiempos de respuesta (para la parte funcional) y con pruebas con usuarios para la evaluación de la interfaz propuesta de la aplicación.

CAPÍTULO 6

Experimentación y Pruebas con Usuarios

El éxito en el uso de una herramienta OLAP se basa en que ésta sea de fácil uso y fácil entendimiento además de proporcionar un tiempo de respuesta considerablemente bueno. Con esta motivación en mente, se presenta un análisis comparativo de tiempos de respuesta haciendo uso de sistemas manejadores de bases de datos analíticos, tanto por tablas como por arreglos, y una base de datos basada en renglones así como los resultados de la prueba con usuarios que se hizo a la interfaz propuesta en el capítulo anterior.

6.1. Comparación de Tiempos de Ejecución de Pruebas Estadísticas

En esta sección se presenta los resultados obtenidos del análisis de tiempos de respuesta entre los sistemas manejadores de base de datos *PostgreSQL*, *HP Vertica* y *SciDB* para el manejo de grandes volúmenes de datos. Los tiempos presentados son los reportados por el DBMS, además, se borró la memoria cache para evitar que el manejador de base de datos haga uso de ésta y optimice los tiempos de respuesta. Para la realización de las pruebas se utilizó el siguiente Hardware y Software:

Hardware:

- Procesador Intel Core i7 a 2.40GHz con 4 núcleos.
- 16 GB de Memoria RAM.

Software:

- *SciDB* 14.12 con 3 instancias de forma local.
- *HP Vertica*.
- *PostgreSQL* 8.4 .
- Ubuntu 12.04 64 bits.
- Java 7.

6.1.1. Definición Tabla y Arreglo

Para la prueba de tiempos de respuesta de las pruebas estadísticas sobre los sistemas manejadores de base de datos analíticos descritos en la Sección 3, se cuenta con una tabla la cual contiene datos de la historia clínica de algunos paciente con tuberculosis (tbbm), el historial contiene características de los pacientes como el sexo, si fuma o no, si padece diabetes, etc. Esta tabla contiene 132 atributos y 41,385,984 de registros la cual fueron cargadas en *PostgreSQL* y *HP Vertica*, para el caso de *SciDB* se tomaron 10 atributos candidatos a llave primaria y se creó un arreglo de 10 dimensiones con 123 atributos. sobre estos datos se realizaron las pruebas hipótesis para la media de una población, para la diferencia entre la media de dos poblaciones y de independencia, de las cuales se compararon los tiempos para la creación de la diferentes tablas o arreglos. En la Figura 6.1 se muestran algunos de los atributos representativos de esta tabla y en la Figura 6.2 la estructura del arreglo equivalente.

En esta estructura de arreglo se encuentran algunos atributos y las dimensiones que atributos de por los que se desea hacer el análisis, dado que las dimensiones en un arreglo forman una dependencia funcional con los atributos de este, se agrega una dimensión extra llamada n la cual hace que las diferentes combinaciones de las dimensiones sean únicas.

tbbm	
folio	character varying(7)
sexo	numeric(10)
escolrec	numeric(10)
peso	numeric(73)
vih	numeric(10)
derechoh	numeric(10)
fiebre	numeric(10)
pdia3	numeric(10)
expua2	numeric(10)
alcohol	numeric(10)
fuma	numeric(10)

Figura 6.1: Atributos relevantes de la tabla tbbm.

```
create array tbbm<
  folio    : string NULL,
  pauci    : double NULL,
  pauci2   : double NULL,
  rfc      : string NULL,
  tf_relac : string NULL,
  f_relac  : string NULL,
  td_relac : string NULL,
  bmtb     : double NULL,
  vecesfol : double NULL,
... >
[ sexo      =0:?,?,0,
  alcohol   =0:?,?,0,
  pdia3     =0:?,?,0,
  fuma      =0:?,?,0,
  vih       =0:?,?,0,
  fiebre    =0:?,?,0,
  escolrec  =0:?,?,0,
  expua2    =0:?,?,0,
  derechoh  =0:?,?,0,
  n         =0:?,?,0 ];
```

Figura 6.2: Atributos relevantes del arreglo tbbm .

6.1.2. Tiempos

La siguiente tabla muestra los tiempos de ejecución, para la ejecución de las pruebas estadísticas sobre el manejador base de datos *PostgreSQL*, *HP Vertica* y *SciDB*, estas pruebas fueron realizadas contemplando tres dimensiones (sexo,alcohol y fuma) y una medida (pdia3). En la Figura 6.3 se muestra la visualización de los resultados obtenidos para la prueba de hipótesis para la media de una población donde los cuboides cuyo color es diferente de gris son los cuboides donde la hipótesis

nula fue rechazada y por consiguiente se aceptó la alternativa. En la Figura 6.4 se muestra la visualización en forma de lattice en donde los pares de cuboides en los cuales se rechazó la hipótesis nula están representados en el panel inferior derecho en cuadros de color azul. Como se puede observar en la Tabla 6.1, la mayor parte del tiempo de ejecución de las pruebas se concentra en el cálculo de las estadísticas necesarias para la ejecución de las mismas (media, varianza y desviación estándar, para cada población). En la comparación de los manejadores de bases de datos, *SciDB* y *Vertica* muestra un mejor tiempo de respuesta en el procesamiento de grandes cantidades de datos ya que en todos los casos presentaron un mejor tiempo de respuesta que *PostgreSQL* y en el procesamiento analítico de éstos.

Prueba Estadística	**Tabla/Arreglo**	**Tiempos de Ejecución**		
		PostgreSQL	*SciDB*	*Vertica*
Prueba de hipótesis para la media de una población	tempMSV	198.105	7.66	6.056
	tbbm_TempZ	0.0186	0.01	0.081
	tbbm_Final	0.0249	0.03	0.044
	Tiempo Total	198.144	7.7	6.182
Prueba de hipótesis para la diferencia entre las media de dos poblaciones	tempMSV	219.964	6.56	1.407
	tempCompare	0.041	0.11	0.687
	tbbm_TempZ	0.22	0.02	0.042
	tbbm_Final	0.029	0.02	0.034
	Tiempo Total	220.057	6.71	2.172
Prueba de hipótesis de independencia	tempMSV	180.739	7.56	4.680
	tempComparePP	181.049	37.28	1.168
	tbbm_TempZ	0.035	0.02	0.032
	tbbm_Final	0.029	0.02	0.037
	Tiempo Total	361.853	44.88	5.918

Tabla 6.1: Tabla comparativa de tiempos de respuesta (segundos).

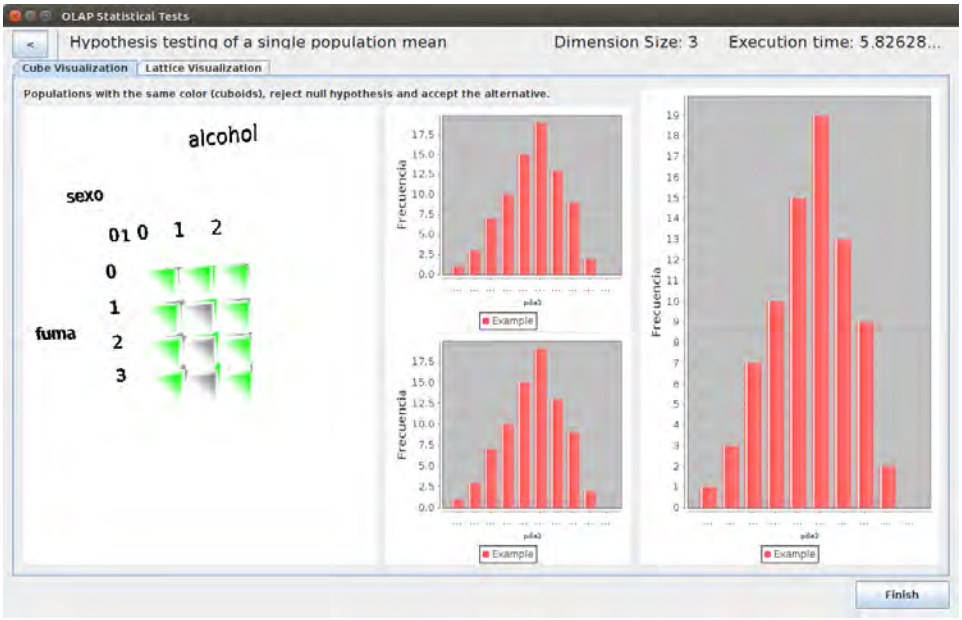

Figura 6.3: Visualización de resultados en forma de cubo OLAP.

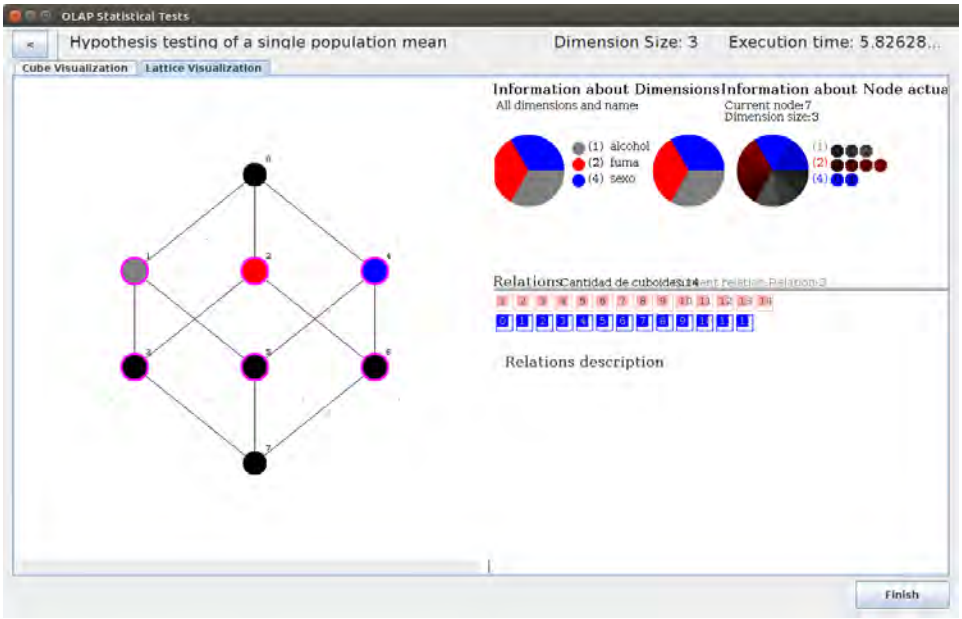

Figura 6.4: Visualización de resultados en forma de retícula multidimensional.

Esta reducción de tiempos se debe en gran medida al tipo de almacenamiento de cada manejador, ya que como vimos en capítulos anteriores tanto *Vertica* como *SciDB* hacen uso de un almacenamiento de tipo columnar a diferencia de *PostgreSQL*, el cual hace uso de un almacenamiento por registros. Sin embargo, *Vertica* presenta un mejor rendimiento ya que no solo tiene un almacenamiento columnar,

sino que también toma ventaja de las diferentes proyecciones que almacena, lo cual representa una gran ventaja ya que estas proyecciones son subconjuntos de atributos almacenados de manera redundante. A continuación, se presenta lo referente a la prueba de la interfaz gráfica con usuarios.

6.2. Pruebas con Usuarios

Como vimos en capítulos anteriores, se debe contar con una herramienta que esté en armonía con las necesidades del usuario y para estar seguro de que se cumple lo mejor posible con esto, se realizan pruebas con usuarios siguiendo las distintas etapas descritas en la Sección 4.4.2, con la finalidad de validar las hipótesis bajo las cuales fueron diseñadas las distintas tareas. En esta sección se muestran los resultados obtenidos de dicha prueba partiendo del análisis del perfil de los usuarios participantes y las hipótesis bajo las cuales guió la prueba, presentando lo que ocurrió durante la prueba para cada usuario y finalizando con la evaluación subjetiva que el mismo usuario hizo del sistema.

6.2.1. Perfil del Usuario

El perfil de usuario constituye un elemento básico para diseñar servicios de información, esto nos ayuda a entender para quien se está diseñando un producto y es clave a la hora de validar las actividades de usabilidad y de experiencia de usuario. Para generar el perfil de usuario se sometió a los usuarios a un cuestionario anónimo del cual podemos obtener la siguientes conclusiones:

- El 75 % de los usuarios tienen un nivel educativo de Maestría.
- 3 de 4 usuarios conocen alguna prueba estadística de los cuales 2 afirman conocer las pruebas de hipótesis y su utilidad.
- El 100 % de los usuarios usa o ha usado algún paquete estadístico.
- Los usuarios tienen conocimiento sobre análisis multidimensional de datos.
- El 75 % de los usuarios están familiarizados con las bases de datos tipo OLAP y su terminología.

Dadas estas conclusiones podemos decir que el perfil de los usuarios dentro de esta prueba es el un usuario con conocimiento de bases de datos y estadística, por

lo cual cumplen con el perfil de usuario para el que se planeó la aplicación.

6.2.2. Hipótesis

Un primer paso para efectuar la una prueba de usabilidad con usuarios es la identificación de las hipótesis las cuales guiarán la prueba y bajo las cuales se diseñarán las tareas que deberá realizar el usuario a través de la prueba para probar la validez de cada hipótesis establecida. Las hipótesis sobre las cuales se realizó esta prueba son las siguientes:

1. El usuario entiende los elementos de la primer interfaz.
2. El usuario puede seleccionar la tabla o arreglo del DBMS indicado.
3. El usuario puede seleccionar las dimensiones y medidas para el análisis.
4. El usuario puede seleccionar la prueba estadística, el método y cambiar el nombre del archivo de salida.
5. El usuario puede configurar los parámetros de la prueba estadística.
6. El usuario puede ejecutar la prueba estadística.
7. El usuario entiende la interfaz de visualización de resultados.
8. El usuario puede navegar entre las formas de visualización.
9. El usuario puede regresar y realizar otra prueba.
10. El usuario puede terminar la aplicación en cualquier momento.

A continuación se presentan una tabla por usuario de las actividades, su calificación, explicación de lo sucedido y propuesta de solución a su problema.

Actividad	**Evaluación**	**Calificación**	**Que Sucedió**	**Propuesta Solución**
Por favor describa todos los elementos de la primer interfaz y para que sirven	El usuario describe satisfactoriamente todos los elementos de la primer interfaz	Con Errores = 17 a 13 de los elementos	No quedo muy claro que hacia cada método	Agregar texto de ayuda para recordarle al usuario que el lo que hace cada método
Por favor seleccione la tabla zmed655 del manejador SciDB	El usuario seleccionó satisfactoriamente el manejador y la tabla correspondiente	Bien = Seleccionó a la primera el manejador y la tabla		

Continúa en la siguiente hoja

Tabla 6.2 – *Continuación*

Actividad	**Evaluación**	**Calificación**	**Que Sucedió**	**Propuesta Solución**
Por favor seleccione las dimensiones y medidas y agregelas a la lista correspondiente	El usuario selecciona y agrega las dimensiones y medidas satisfactoriamente	Bien= Seleccionó y agregó a la primera las dimensiones y atributos		
Por favor seleccione la prueba con el método indicados en la ficha y cambie el nombre del archivo a salida.sql	El usuario selecciona correctamente la prueba estadística, el método y cambia el nombre del archivo de salida	Bien = Seleccionó a la primera la prueba estadística, el método y cambió el nombre del archivo al solicitado		
Por favor asigne el valor de 20 para el parámetro "Mean"	El usuario asigno correctamente el parámetro "Mean"	Bien = Asignó correctamente el valor		
Por favor describa todos los elementos de la interfaz de visualización y para que sirven	El usuario describe satisfactoriamente todos los elementos de la interfaz de visualización	Bien= Todos los elementos (11)		
Por favor cambie de la vista de cubo multidimensional a la de lattice	El usuario cambio correctamente entre las vistas	Bien = Cambio entre las vistas a la primera		
Por favor regrese a la interfaz anterior y realice la prueba con los parámetros, ambos indicados en la ficha	El usuario regreso a la interfaz anterior y ejecutó el nuevo test	Bien = Regreso correctamente y realizo la prueba		
Por favor finalice la aplicación desde la pantalla en que se encuentre	El usuario cerro correctamente la aplicación	Bien = El usuario cerro la aplicación sin demoras		

Tabla 6.2: Análisis de Actividades Usuario 1

Nota: Agregar algo que muestre un mensaje o estado de procesamiento durante la ejecución. Separar las dimensiones y atributos de los arreglos para el caso de SciDB.

Solución: Agregar una barra de progreso de las tareas y separar las dimensiones y atributos de los array den don componentes distintos .

Actividad	**Evaluación**	**Calificación**	**Que Sucedió**	**Propuesta Solución**
Por favor describa todos los elementos de la primer interfaz y para que sirven	El usuario describe satisfactoriamente todos los elementos de la primer interfaz	Con Errores = 16 a 13 de los elementos	No fue claro de primera vista como es que se selecciona y agregan las dimensiones. No fue claro que representa la salida y donde es que se almacena este archivo	Hacer mas énfasis en como se agregan las dimensiones y medidas. Poner mensaje donde se indique donde se va a guardar el archivo
Por favor seleccione la tabla zmed655 del manejador SciDB	El usuario seleccionó satisfactoriamente el manejador y la tabla correspondiente	Bien = Seleccionó a la primera el manejador y la tabla		
Por favor seleccione las dimensiones y medidas y agregelas a la lista correspondiente	El usuario selecciona y agrega las dimensiones y medidas satisfactoriamente	Bien= Seleccionó y agregó a la primera las dimensiones y atributos		
Por favor seleccione la prueba con el método indicados en la ficha y cambie el nombre del archivo a salida.sql	El usuario selecciona correctamente la prueba estadística, el método y cambia el nombre del archivo de salida	Bien = Seleccionó a la primera la prueba estadística, el método y cambió el nombre del archivo al solicitado		
Por favor asigne el valor de 20 para el parámetro "Mean"	El usuario asigno correctamente el parámetro "Mean"	Bien = Asignó correctamente el valor		
Por favor describa todos los elementos de la interfaz de visualización y para que sirven	El usuario describe satisfactoriamente todos los elementos de la interfaz de visualización	Con Errores = 10 a 6 de los elementos	No quedo claro cual es la diferencia entre las diferentes gráficas de frecuencia (Histogramas)	Indicar cual es la diferencia entre ellas con un icono de ayuda
Por favor cambie de la vista de cubo multidimensional a la de lattice	El usuario cambio correctamente entre las vistas	Bien = Cambio entre las vistas a la primera		
Por favor regrese a la interfaz anterior y realice la prueba con los parámetros, ambos indicados en la ficha	El usuario regreso a la interfaz anterior y ejecutó el nuevo test	Bien = Regreso correctamente y realizo la prueba		
Por favor finalice la aplicación desde la pantalla en que se encuentre	El usuario cerro correctamente la aplicación	Bien = El usuario cerro la aplicación sin demoras		

Tabla 6.3: Análisis de Actividades Usuario 2

Actividad	Evaluación	Calificación	Que Sucedió	Propuesta Solución
Por favor describa todos los elementos de la primer interfaz y para que sirven	El usuario describe satisfactoriamente todos los elementos de la primer interfaz	Bien= Todos los elementos		
Por favor seleccione la tabla zmed655 del manejador SciDB	El usuario seleccionó satisfactoriamente el manejador y la tabla correspondiente	Bien = Seleccionó a la primera el manejador y la tabla		
Por favor seleccione las dimensiones y medidas y agregelas a la lista correspondiente	El usuario selecciona y agrega las dimensiones y medidas satisfactoriamente	Bien = Seleccionó a la primera la prueba estadística, el método y cambió el nombre del archivo al solicitado		
Por favor seleccione la prueba con el método indicados en la ficha y cambie el nombre del archivo a salida.sql	El usuario selecciona correctamente la prueba estadística, el método y cambia el nombre del archivo de salida	Bien = Seleccionó a la primera la prueba estadística, el método y cambió el nombre del archivo al solicitado		
Por favor asigne el valor de 20 para el parámetro "Mean"	El usuario asigno correctamente el parámetro "Mean"	Bien = Asignó correctamente el valor		
Por favor describa todos los elementos de la interfaz de visualización y para que sirven	El usuario describe satisfactoriamente todos los elementos de la interfaz de visualización	Con Errores = 10 a 6 de los elementos	No quedo claras las gráficas de frecuencia (Histogramas)	Indicar cual es la diferencia entre ellas con un icono de ayuda
Por favor cambie de la vista de cubo multidimensional a la de lattice	El usuario cambio correctamente entre las vistas	Bien = Cambio entre las vistas a la primera		
Por favor regrese a la interfaz anterior y realice la prueba con los parámetros, ambos indicados en la ficha	El usuario regreso a la interfaz anterior y ejecutó el nuevo test	Bien = Regreso correctamente y realizó la prueba		
Por favor finalice la aplicación desde la pantalla en que se encuentre	El usuario cerro correctamente la aplicación	Bien = El usuario cerro la aplicación sin demoras		

Tabla 6.4: Análisis de Actividades Usuario 3

Actividad	**Evaluación**	**Calificación**	**Que Sucedió**	**Propuesta Solución**
Por favor describa todos los elementos de la primer interfaz y para que sirven	El usuario describe satisfactoriamente todos los elementos de la primer interfaz	Bien= Todos los elementos		
Por favor seleccione la tabla zmed655 del manejador SciDB	El usuario seleccionó satisfactoriamente el manejador y la tabla correspondiente	Bien = Seleccionó a la primera el manejador y la tabla		
Por favor seleccione las dimensiones y medidas y agregelas a la lista correspondiente	El usuario selecciona y agrega las dimensiones y medidas satisfactoriamente	Con Errores = Tardo más de 5 min en agregar las dimensiones y medidas o no le fue claro	Le costo trabajo buscar las dimensiones solicitadas en la lista de atributos	Ordenar alfabéticamente los atributos
Por favor seleccione la prueba con el método indicados en la ficha y cambie el nombre del archivo a salida.sql	El usuario selecciona correctamente la prueba estadística, el método y cambia el nombre del archivo de salida	Bien = Seleccionó a la primera la prueba estadística, el método y cambió el nombre del archivo al solicitado		
Por favor asigne el valor de 20 para el parámetro "Mean"	El usuario asigno correctamente el parámetro "Mean"	Bien = Asignó correctamente el valor		
Por favor describa todos los elementos de la interfaz de visualización y para que sirven	El usuario describe satisfactoriamente todos los elementos de la interfaz de visualización	Con Errores = 10 a 6 de los elementos	No quedo claras las gráficas de frecuencia (Histogramas)	Poner mas claros los rangos
Por favor cambie de la vista de cubo multidimensional a la de lattice	El usuario cambio correctamente entre las vistas	Bien = Cambio entre las vistas a la primera		
Por favor regrese a la interfaz anterior y realice la prueba con los parámetros, ambos indicados en la ficha	El usuario regreso a la interfaz anterior y ejecutó el nuevo test	Bien = Regreso correctamente y realizó la prueba		
Por favor finalice la aplicación desde la pantalla en que se encuentre	El usuario cerro correctamente la aplicación	Bien = El usuario cerro la aplicación sin demoras		

Tabla 6.5: Análisis de Actividades Usuario 4

6.2.3. Evaluación Subjetiva

Durante el desarrollo de la prueba se pudo verificar que el uso de la interfaz es intuitivo, aunque existen algunos detalles con los que podría mejorar la usabilidad como el ordenamiento de los elementos para evitar que el usuario esté buscando dentro de una lista larga de atributos. En cuanto a la experiencia subjetiva del usuario, la mayoría usaría el sistema frecuentemente como se observa en la Figura 6.6 donde la mitad de los usuarios piensan que no es necesaria la ayuda de un técnico para poder hacer uso del sistema

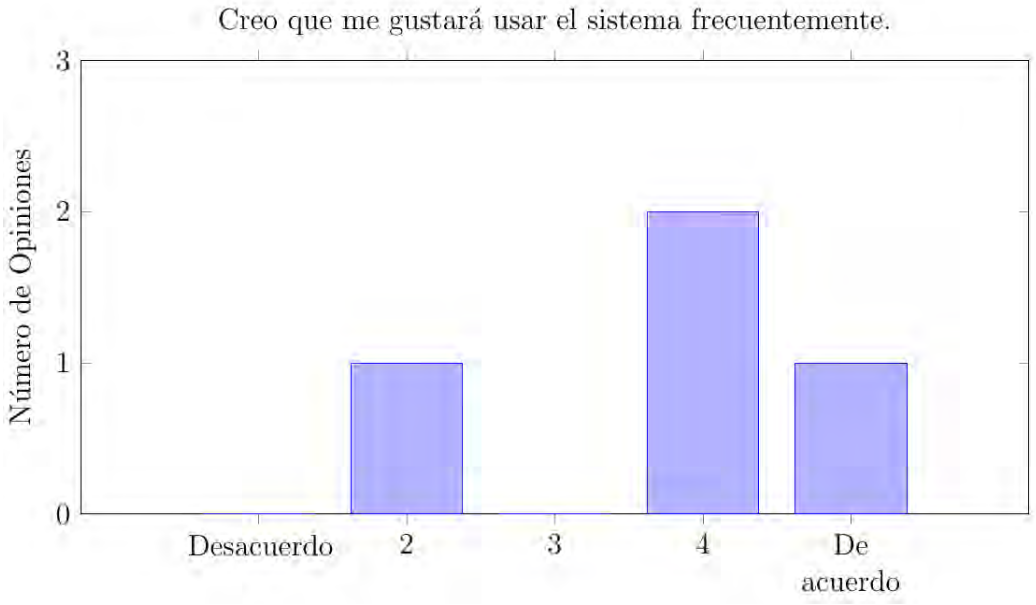

Figura 6.5: Respuestas de los usuarios a la pregunta 1 del cuestionario de evaluación subjetiva.

Por otra parte todos los usuarios creen que las distintas funciones del sistema estaban bien integradas y que sería rápido de aprender para otros usuarios, finalmente los usuarios se sintieron seguros como se puede observar en la Figura 6.7.

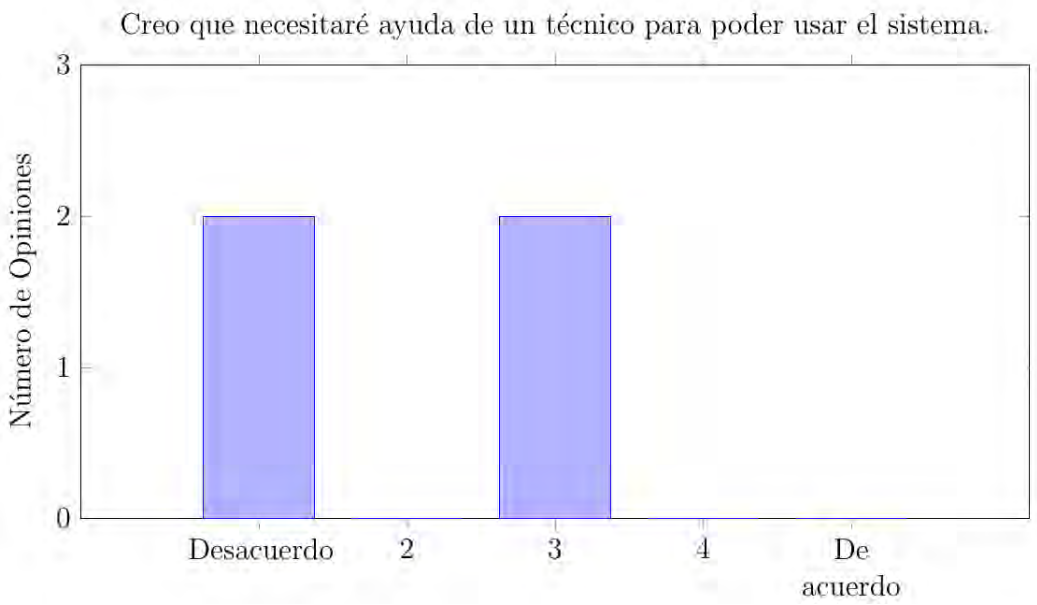

Figura 6.6: Respuestas de los usuarios a la pregunta 4 del cuestionario de evaluación subjetiva.

Figura 6.7: Respuestas de los usuarios a la pregunta 9 del cuestionario de evaluación subjetiva.

6.2.4. Propuesta de mejora de interfaz

Una vez que se ha realizado las pruebas y analizado que fue lo que sucedió durante la misma se procede a implementar mejoras que ayuden a corregir los problemas encontrados durante la evaluación de los usuarios. A continuación se muestran las mejoras para la siguiente versión de la interfaz en la cual se tomaron en cuenta los problemas que los usuarios tuvieron:

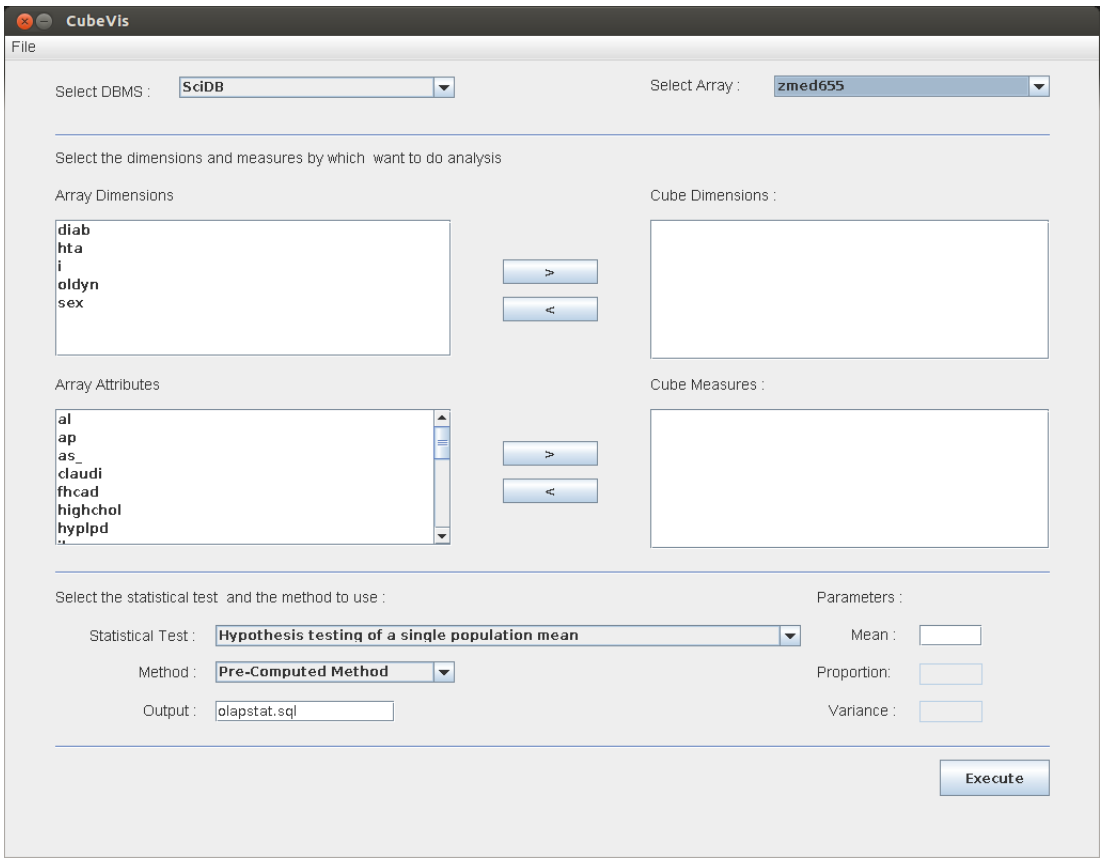

Figura 6.8: Interfaz de inicio.

Para esta interfaz (Figura 6.8) se separaron las dimensiones de los atributos, para el caso de *SciDB*, evitando que los usuarios comentan errores en la selección de los parámetros de la prueba y a su vez facilitando la búsqueda de estos haciendo un ordenamiento alfabético antes de ser desplegados, también se agregó una barra de progreso para indicar el nivel de avance de la prueba. Para la siguiente interfaz (Figura 6.9) se agregaron los elementos de ayuda para explicar más a detalle los

resultados y valores mostrados así como el tiempo que duró la generación de resultados mas no el de visualización de los mismos.

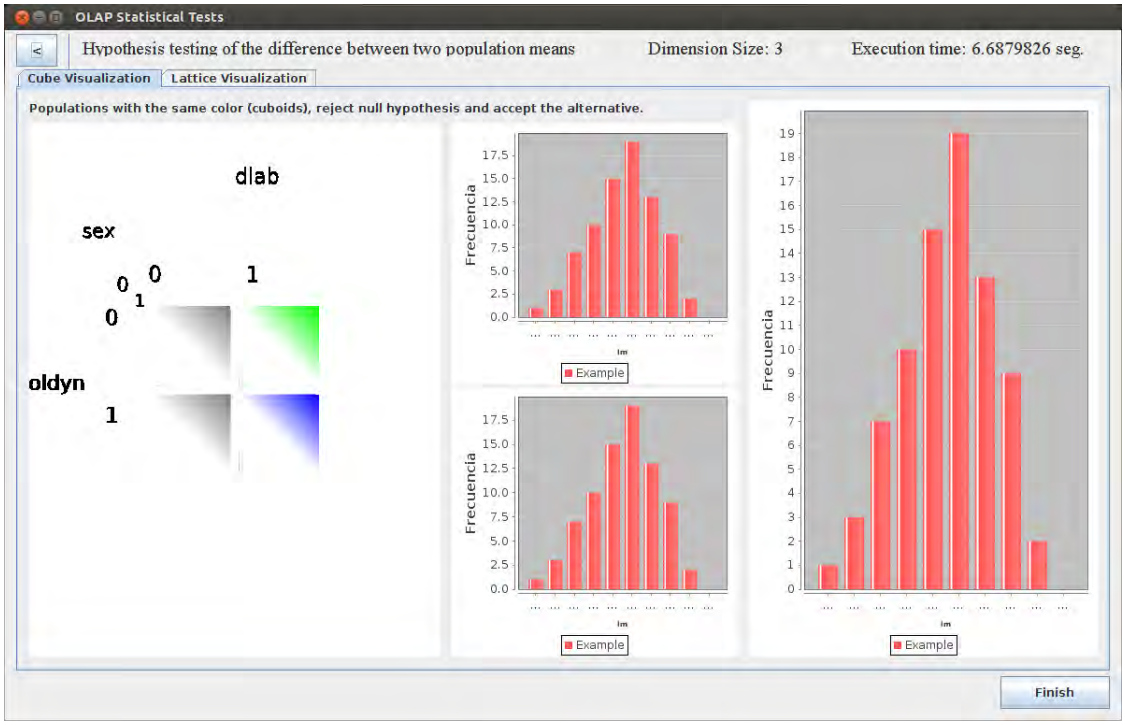

Figura 6.9: Interfaz de visualización de resultados.

CAPÍTULO 7

Conclusiones y Trabajo Futuro

En este trabajo de investigación se presentó el desarrollo de una herramienta de análisis de resultados de pruebas estadísticas haciendo uso de técnicas OLAP, así como de visualización de cubos multidimensionales y su representación matemática en forma de retícula. Esta herramienta se mejoró incluyendo en ella el manejo de sistemas del tipo analíticos (*SciDB* y *HP Vertica*) lo cual, como se vió en el capítulo anterior, proporciona una gran ventaja cuando hablamos del análisis de grandes cantidades de datos. También, como vimos se realizó una mejora a la interfaz de usuario lo que le permite a la herramienta adaptarse mejor a las necesidades del usuario y está pueda ser usada por más usuarios, que es el objetivo final de toda herramienta de análisis de datos.

Mediante la experimentación se comprobaron las ventajas de contar una herramienta que nos permita tomar ventaja de los sistemas manejadores de base de datos analíticos y así poder analizar grandes cantidades de datos, que son los que se manejan en ambientes analíticos y de minería de datos, en un tiempo considerable. Uno de los aspectos a resaltar es que la elaboración de las pruebas estadísticas se realiza dentro del DBMS por lo que no es necesario realizar una exportación de los mismos para su posterior análisis, lo cual puede ser muy costoso y tardado en ambientes analíticos.

En lo que se refiere a trabajos futuros, se puede analizar la manera de incluir otro tipo de pruebas estadísticas o modelos estadísticos que presenten otras opciones de análisis que aprovechen las ventajas de estos sistemas y así poder sacar otro tipo de conclusiones sobre los datos analizados, otra de las mejoras que podrían realizarse es la integración con otras herramientas de análisis de datos que existen en el mercado.

BIBLIOGRAFÍA

[1] Daniel J Abadi, Samuel R Madden, and Nabil Hachem. Column-stores vs. row-stores: How different are they really? In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 967–980. ACM, 2008.

[2] Alberto Abelló and Oscar Romero. On-line analytical processing. 2009.

[3] Rodrigo Fernando Alonso Pinzón. *Evaluación de Interfaces de Usuario en el Proceso Unificado*. Tesis de Maestría. Universidad Nacional Autónoma de México, 2002.

[4] JM Christian Bastien and Dominique L Scapin. Ergonomic criteria for the evaluation of human-computer interfaces. 1993.

[5] Paul G Brown. A survey of scientific applications using scidb. 2005.

[6] Paul G. Brown. Overview of scidb: large scale array storage, processing and analysis. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 963–968. ACM, 2010.

[7] CIMAT. Otra cara de las matemáticas. http://www.cimat.mx/ciencia_para_jovenes/otra_cara/sesion4.html, 2010.

[8] Carlos Coronel and Steven Morris. *Database Systems: Design, Implementation & Management*. Cengage Learning, 9th edition, 2009.

[9] Judith Bayard Cushing, James French, and Shawn Bowers. *Scientific and Statistical Database Management: 23rd International Conference, SSDBM 2011, Portland, OR, USA, July 20-22, 2011. Proceedings*, volume 6809. Springer, 2011.

[10] Wayne W. Daniel. *Bioestadística: Base para el análisis de las ciencias de la salud*. Number 04; HA29, D3. Editorial Limusa S.A. De C.V., 3 edition, 1980.

[11] Jay L. Devore. *Probabilidad y estadística para ingenierías y ciencias*. Cengage Learning Editores, 6th revised edition edition, 2008.

[12] Edgar Durán Muñoz. *Análisis y visualización de resultados OLAP*. Tesis de Maestría. Universidad Nacional Autónoma de México, Junio 2013.

[13] Guadalupe Vanessa Carolina Gutiérrez Hernández. *Herramienta Gráfica para la Visualización de Cubos Multidimensionales OLAP*. Tesis de Maestría. Instituto Politécnico Nacional, Enero 2015.

[14] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[15] S. Hassan-Montero, Y. & Ortega-Santamaría. Informe apei de usabilidad. http://www.nosolousabilidad.com/manual/index.htm, 2009.

[16] William H. Inmon. *Building the data warehouse*. John wiley & sons, 2005.

[17] International Ergonomics Association. http://www.iea.cc/.

[18] ISO. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Technical report, International Organization for Standardization, 1998.

[19] ISO. Iso 9241-210 (2009). ergonomics of human system interaction-part 210: Human-centered design for interactive systems (formerly known as 13407). *International Organization for Standardization (ISO). Switzerland*, 2009.

[20] James Kalbach. *Designing web navigation*. O'Reilly Media, Inc., 2007.

[21] Andrew Lamb, Matt Fuller, Ramakrishna Varadarajan, Nga Tran, Ben Vandiver, Lyric Doshi, and Chuck Bear. The vertica analytic database: C-store 7 years later. *Proceedings of the VLDB Endowment*, 5(12):1790–1801, 2012.

[22] Bryan Lewis and Alex Poliakov. Big analytics without big hassles. http://es.slideshare.net/Paradigm4_/paradigm4-sci-db-ieee-webinar, 2013.

[23] Gheorghe Matei and Romanian Commercial Bank. Column-oriented databases, an alternative for analytical environment. *Database Systems Journal*, 1(2):3–16, 2010.

[24] Carlos Ordonez and Zhibo Chen. Evaluating statistical tests on olap cubes to compare degree of disease. *Information Technology in Biomedicine, IEEE Transactions on*, 13(5):756–765, 2009.

[25] Carlos Ordonez, Zhibo Chen, and Javier García-García. Interactive exploration and visualization of olap cubes. In *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*, pages 83–88. ACM, 2011.

[26] Hewlett Packard. Hybrid storage model. https://my.vertica.com/docs/7.1.x/HTML/Content/Authoring/ConceptsGuide/Components/HybridStorageModel.htm, 2014.

[27] Hewlett Packard. *HP Vertica Complete Documentation*. 7.2 edition, 2016.

[28] Paradigm 4. *SciDB User's Guide*. 12.3 edition, 2012.

[29] Nigel Pendse. What is olap?, an analysis of what the increasingly misused olap term is supposed to mean. http://dssresources.com/papers/features/pendse04072002.htm, 2002.

[30] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests*. Wiley Publishing, 2nd edition, 2008.

[31] Florin Rusu and Yu Cheng. A survey on array storage, query languages, and systems. *arXiv preprint arXiv:1302.0103*, 2013.

[32] Ben Shneiderman. Leonardo's laptop: Human needs and the new computing technologies. *Cambridge: The MIT Press*, 270:15, 2002.

[33] Sinnexus. Bases de datos oltp y olap. http://www.sinnexus.com/business_intelligence/olap_vs_oltp.aspx, 2007.

[34] Mike Stonebraker, Daniel J Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O'Neil, et al. C-store: a column-oriented dbms. In *Proceedings of the 31st*

international conference on Very large data bases, pages 553–564. VLDB Endowment, 2005.

[35] Vladimir Andreevich Uspenskii. *Pascal's Triangle. Popular Lectures in Mathematics.* Popular Lectures in Mathematics. University of Chicago Press, 1975.

[36] Ronald E Walpole, Raymond H Myers, and Sharon L Myers. *Probabilidad y Estadística para Ingeniería y Ciencias.* Pearson Educación, 9 edition, 2012.

APÉNDICE A

Documentos Plan de Pruebas

Prueba:_____
Fecha:_____
Nota:_____

Protocolo de bienvenida para prueba de usabilidad del proyecto CubeVis

Buenos días, mi nombre es <Nombre_Moderador> y estaré con usted en esta sesión. Permítame explicarle porqué está usted aquí.

Estamos probando el prototipo de un sistema que permitirá a los investigadores el análisis de resultados de pruebas estadísticas sobre bases de datos multidimensionales.

Mediante el estudio de sus acciones trataremos de determinar los defectos o virtudes de este sistema en términos de la facilidad con que usted interactúe con el mismo. Es importante enfatizar que el que será evaluado es el prototipo y no usted, por ello le pedimos que actué con naturalidad, pero sobre todo, que responda con honestidad.

Debido a que no es un sistema final existe la posibilidad de que no funcione adecuadamente, por favor no se extrañe si el sistema hace algo inesperado.

Se le presentará un prototipo y se le pedirá que realicé algunas tareas típicas para las cuales está diseñado el sistema, la sesión consistirá en que usted las efectué y describa en voz alta sus acciones así como cualquier opinión que tenga ya que esto será de mucha ayuda para nosotros.

Una vez comenzada la sesión siéntase en total libertad de hacer cualquier pregunta aunque no podré contestar algunas de ellas, ya que el objetivo es simular la situación real en la cual operará el sistema de manera autónoma.

¿Tiene alguna duda?

Actividades del Plan de Pruebas
para el Sistema CubeVis

Ir tachando las actividades realizadas con el fin de no perderse durante la prueba. NO se realizará ninguna actividad si antes no ha concluido la anterior (a menos que se le instruya lo contrario).

Nota: En caso de que el usuario cometa cualquier error preguntar: *¿El sistema le informó con claridad qué fue lo que pasó?*

1. ___ Por favor describa todos los elementos de la primer interfaz y para qué sirven.
2. ___ Por favor seleccione la tabla del manejador especificados en la ficha #1 .
3. ___ Por favor selecciones y agregue las dimensiones y medidas, especificadas en la ficha #2.
4. ___ Por favor seleccione la prueba _____ con el método y cambie el nombre del archivo a _____ , indicados en la ficha #3.
5. ___ Por favor asigne el valor especificado en la ficha #4 a la media para la prueba _____ y ejecutela.
6. ___ Apuntando con el mouse, describa lo que ve en la pantalla de visualización de resultados e indique para qué cree que es cada elemento que observa en ella.
7. ___ Por favor cambie la vista de resultados de cubo multidimensional a la visualización en forma de lattice.
8. ___ Por favor regrese a la interfaz anterior y realice la prueba estadística _____ con los parámetros especificados en la ficha #5
9. ___ Por favor finalice la aplicación desde la interfaz donde se encuentre.

Prueba:_____
Fecha:_____
Nota:_____

Cuestionario de perfil del usuario

1. ¿Cuál es su nivel académico?

   _____

2. ¿Cuales pruebas estadísticas conoce ?

   _____

3. ¿Conoce el objetivo de las pruebas de Hipótesis?

   _____

4. ¿Ha utilizado algún paquete estadístico?

   _____

5. ¿Con qué frecuencia?

   _____

6. ¿Tiene conocimientos de análisis de datos multidimensional?

   _____

7. ¿Tiene conocimiento tiene de bases de datos tipo OLAP?

   _____

8. ¿Esta familiarizado con la terminologia de modelo de almacenamiento de datos para bases de datos OLAP?

   _____

*Prueba:*_____
*Fecha:*_____
Nota:_____

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas.

	Fuertemente en desacuerdo				**Fuertemente de acuerdo**
1. Creo que me gustará usar el sistema frecuentemente.	1	2	3	4	5
2. Encuentro el sistema innecesariamente complejo.	1	2	3	4	5
3. Pensé que el sistema era fácil de usar.	1	2	3	4	5
4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema.	1	2	3	4	5
5. Encontré que las distintas funciones del sistema estaban bien integradas.	1	2	3	4	5
6. Pienso que había mucha inconsistencia en el sistema.	1	2	3	4	5
7. Me imagino que la gente aprenderá a usar el sistema bastante rápido.	1	2	3	4	5
8. Encuentro al sistema engorroso de usar.	1	2	3	4	5
9. Me sentí muy seguro usando el sistema.	1	2	3	4	5
10. Necesito aprender muchas cosas antes de poder utilizar el sistema.	1	2	3	4	5