Diseño e Implementación de una Arquitectura de Datawarehouse Escalable

P. Garzón, C. Rojas y M. Almache.

Departamento de Ciencias de la Computación, Escuela Politécnica del Ejército, Sangolquí, Ecuador
pablogg.ec@hotmail.com; crrojas@espe.edu.ec; gmalmache@espe.edu.ec

RESUMEN: El desarrollo de un Datawarehouse es una alternativa utilizada a nivel empresarial en soluciones de Business Intelligence (BI), proporcionando información de soporte para la toma de decisiones. Los beneficios de una solución de BI, pueden aprovecharse, anticipadamente, a la finalización de un proyecto de desarrollo, cuando se utiliza una metodología por ciclos iterativos; proporcionando, incrementalmente, los recursos para análisis de información. El presente artículo, explora el desarrollo de una solución de BI, basándose en el diseño de una arquitectura de Datawarehouse, con características de escalabilidad, posibilitando la integración progresiva de sistemas de origen de datos y, la construcción de aplicaciones de BI. El diseño de esta arquitectura se ajusta a un proceso de desarrollo por ciclos iterativos; para realizarlo, se definieron los elementos principales y funciones, que determinan las características de escalabilidad. El desarrollo de un Datawarehouse, para el Proceso de Mesa de Servicios de la Empresa TATA CONSULTANCY SERVICES, utiliza las bondades de BI a través de un desarrollo incremental en tres ciclos, cuyos resultados permitieron determinar que, los recursos de información para BI, pueden ser generados desde el primer ciclo de desarrollo, incrementando su alcance en los ciclos posteriores.

ABSTRACT: Data warehousing is an alternative used in enterprise business intelligence solutions to provide information and support for decision making. The benefits of a BI solution can be used before the completion of a development project when using a methodology for incrementally iterative cycles, providing resources to users for information analysis. This article explores the development of a BI solution, aligned to the mentioned background, based on the design of a Datawarehouse architecture with scalability features for the progressive integration of data source systems and building of business intelligence applications. This architecture design conforms to a development process by iterative cycles. For the development of it, were defined the main elements of the architecture and functions that determine the characteristics of scalability. For validation of this approach, was implemented a case study which is based on the incremental development of a Datawarehouse in three cycles; its results shows that information resources for business intelligence can be generated from the first cycle of development by increasing its scope in subsequent cycles.

1 INTRODUCCIÓN

El desarrollo de un Datawarehouse (almacén de datos) es una alternativa utilizada, a nivel empresarial, para soluciones de BI que proporcionan, a nivel directivo, aplicaciones de presentación de información para una oportuna toma de decisiones. Los beneficios de una solución de BI normalmente no pueden percibirse antes, durante, e incluso al finalizar su desarrollo, cuando se utiliza una metodología estructurada, sino, posteriormente a la liberación del producto. Por el contrario, una metodología por ciclos iterativos, permite adelantar los recursos para análisis de información e incrementarlos progresivamente. La revista electrónica GestioPolis, en su publicación "Implementación incremental para Datawarehouse", menciona que, la metodología incremental, nace con la finalidad de definir indicadores que entreguen información relevante para la toma de decisiones, agilizando la puesta en marcha del proyecto de implementación de un Data Warehouse [1]. En concordancia con este concepto, la comunidad de personas interesadas en compartir conocimiento sobre temas relacionados con datos (Dataprix.com), se refiere a la construcción e implantación de un Datawarehouse como un proceso evolutivo, haciendo referencia a la metodología rápida de desarrollo iterativo del mismo RWM [2].

Por su naturaleza, un Datawarehouse requiere ser versátil para adaptarse a los cambios que ocurren en las organizaciones y, satisfacer las demandas permanentes de información para análisis de BI. Como contribución, el presente artículo, trata de la implementación de un Datawarehouse basado en un diseño de arquitectura escalable, aplicando una metodología de desarrollo por ciclos iterativos enfocado en: *i*) la integración de nuevos sistemas fuente de datos; *ii*) el desarrollo incremental de aplicaciones de BI. Para realizarlo, se definen los elementos principales de una arquitectura Datawarehouse y su función, para obtener un modelo de solución escalable desde el punto de vista de diseño. También, se analiza la relación existente entre las tendencias de crecimiento a nivel de número de objetos de las bases de datos intervinientes. La meta final es, demostrar que, las aplicaciones de BI, como informes automáticos y, cubos de información, reducen la necesidad de generación manual de información; y, los resultados, muestran que ésta (la información) puede ser generada de manera automática desde el primer ciclo de desarrollo, incrementando su alcance en los ciclos posteriores.

El resto del artículo ha sido organizado de la siguiente forma: La sección 2, describe el modelo de arquitectura de un Datawarehouse escalable. La sección 3, detalla la implementación iterativa realizada, en base al diseño propuesto. En la sección 4, se muestran los resultados obtenidos de la implementación. En la sección 5, se analizan algunos trabajos relacionados al tema. Finalmente, en la sección 6, se presentan las conclusiones y trabajos.

2 DISEÑO DE ARQUITECTURA DE UN DATAWAREHOUSE ESCALABLE.

Al hablar de un diseño de arquitectura Datawarehouse escalable, puede hacerse referencia a varios aspectos, como por ejemplo: *i)* el crecimiento de los datos en la estructura de un Datawarehouse; *ii)* el número de usuarios que utilizan las aplicaciones de BI; *iii)* la integración de nuevos orígenes de información al Datawarehouse, *iv)* el desarrollo de aplicaciones de BI a partir de un Datawarehouse existente. Este artículo se centra en los dos últimos ítems.

2.1 Elementos del diseño.

La Figura No.1 ilustra los elementos básicos que forman parte del modelo conceptual de una solución Datawarehouse [3], mismos que son explicados a continuación:

- **SISTEMA DE ORIGEN.** Son uno o varios sistemas informáticos utilizados por la organización para sus funciones operativas o de negocio. Para un Datawarehouse, los sistemas de origen, son los proveedores de los datos, mismos que serán transformados y preparados para su explotación en aplicaciones de análisis.
- **ETL.** Proceso de Extracción Transformación y Carga de datos desde los Sistemas de Origen a las bases de datos del Datawarehouse. La responsabilidad de mantener la consistencia de los datos, en un Datawarehouse, está en la programación correcta de los procesos ETL. Un proceso ETL puede utilizar una base de datos (BD) auxiliar, siendo sus funciones principales limpiar, depurar y homologar los datos; sobre todo, cuando provienen de diferentes fuentes. Las diferentes plataformas, para el desarrollo de soluciones de BI, proveen herramientas y servicios integrados para las funciones de ETL.

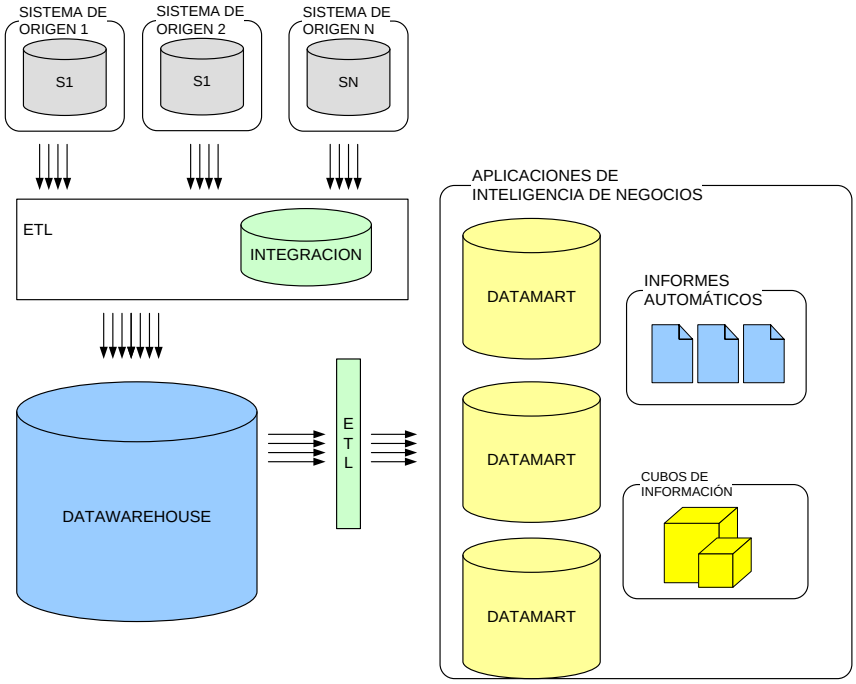

Figura No 1. Diseño de arquitectura de un Datawarehouse escalable.

- **BASES DE DATOS DE DATAWAREHOUSE.** Una o varias bases de datos, que constituyen el medio lógico y físico, encargado de guardar los datos procesados por los ETL desde los sistemas de Origen. La BD Datawarehouse es el centro del diseño de la arquitectura; su modelo de datos debe ser orientado a las características propias de un Datawarehouse: Orientación a entidades, Integración, No volatilidad, Carga y consulta masiva de datos.
- **APLICACIONES DE BI.** Son las aplicaciones o herramientas destinadas a la explotación de los datos, proveyendo un aspecto analítico de la información como soporte al proceso de toma de decisiones. Normalmente, las aplicaciones de BI, pueden ser desarrolladas sobre plataformas integradas para análisis de datos multidimensionales OLAP y servicios para la generación de informes automáticos. Para la presentación de datos, en estas aplicaciones, el uso de bases de datos DATAMART son una alternativa adecuada por sus características, ya

que, resume grandes cantidades de datos en información específica que el usuario quiere conocer, pudiendo visualizarla en la aplicación para realizar análisis específicos, sin tener que escudriñar manualmente los grandes bloques de datos.

2.2 Escalabilidad en la integración de sistemas de origen de datos.

Consiste en integrar nuevos sistemas de origen de datos a un Datawarehouse existente. La clave para alcanzar esta característica se encuentra en la arquitectura de los procesos ETL. El uso de una BD auxiliar, como se ilustra en la figura 1 (BD "INTEGRACION"), permite depurar, homologar, transformar y, finalmente, integrar datos, de las fuentes existentes con la nueva fuente de datos. En esta fase, de la arquitectura, se realizan todas las transformaciones de datos necesarias, para que, la información, sea entregada al Datawarehouse de manera íntegra y consistente.

2.3 Escalabilidad en el desarrollo de aplicaciones de BI.

Es una característica necesaria por la naturaleza del tipo de aplicación. A causa de la generación de conocimiento, la perspectiva de análisis de los usuarios, es ampliada y, son extendidos los requerimientos funcionales, generando nuevas necesidades. Una solución de BI, puede quedar obsoleta si su Datawarehouse no es un proveedor de datos central, consolidado, depurado y sobre todo escalable. Los Datamart, que proporcionan datos para BI, pueden requerir modificaciones y desarrollos nuevos, por los cambios a los que, generalmente, suceden en una organización. Por lo mencionado, es preferible modificar un Datamart o crear uno nuevo para satisfacer las necesidades de información, en lugar de modificar una BD central de un Datawarehouse, de donde se pueden extraer datos para diferentes soluciones de BI, pudiendo existir impactos colaterales.

3 IMPLEMENTACIÓN

El siguiente caso práctico resume la implementación del Datawarehouse de Mesa de Servicios de la empresa Tata Consultancy Services en Ecuador, el cual, ha sido desarrollado en un proceso iterativo, aplicando la metodología MSF [4], alineando el proceso de desarrollo al diseño de arquitectura presentado.

3.1 Proceso de desarrollo. Primera iteración.

Se realizaron: el análisis inicial de requerimientos, el diseño de la solución en base al modelo de arquitectura presentado en este artículo, los modelos de datos y, procesos ETL. En cuanto a la construcción, se desarrollaron los procesos de almacenamiento del Datawarehouse y, transformación de los datos del sistema de origen (ver Figura 2).

3.2 Proceso de desarrollo. Segunda iteración.

Se implementaron las aplicaciones de BI para el esquema de análisis, correspondiente al requerimiento funcional RF2.1: "Administración de atención de requerimientos". Para complementar la información requerida, se incorporó el sistema Septimus como una nueva fuente de datos; en este punto, se puso en prueba la característica de escalabilidad para la integración de nuevos orígenes de datos (ver Figura 3).

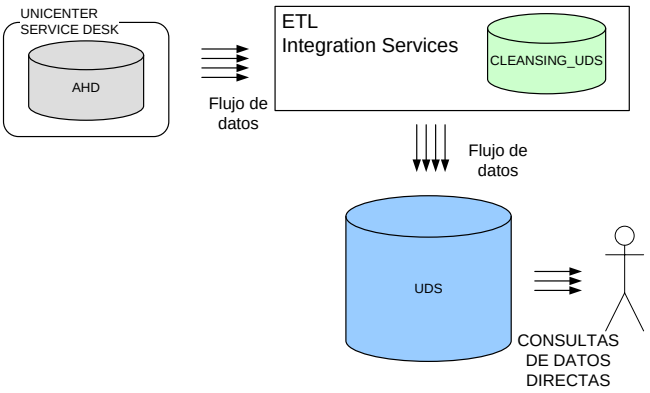

Figura No 2. Modelo conceptual de la solución en la primera iteración.

3.3 Proceso de desarrollo. Tercera iteración.

Se desarrollaron las aplicaciones de BI para el esquema de análisis, correspondiente al requerimiento funcional RF2.2: "Administración de nivel de servicio externo e interno". La Figura 3 ilustra lo indicado.

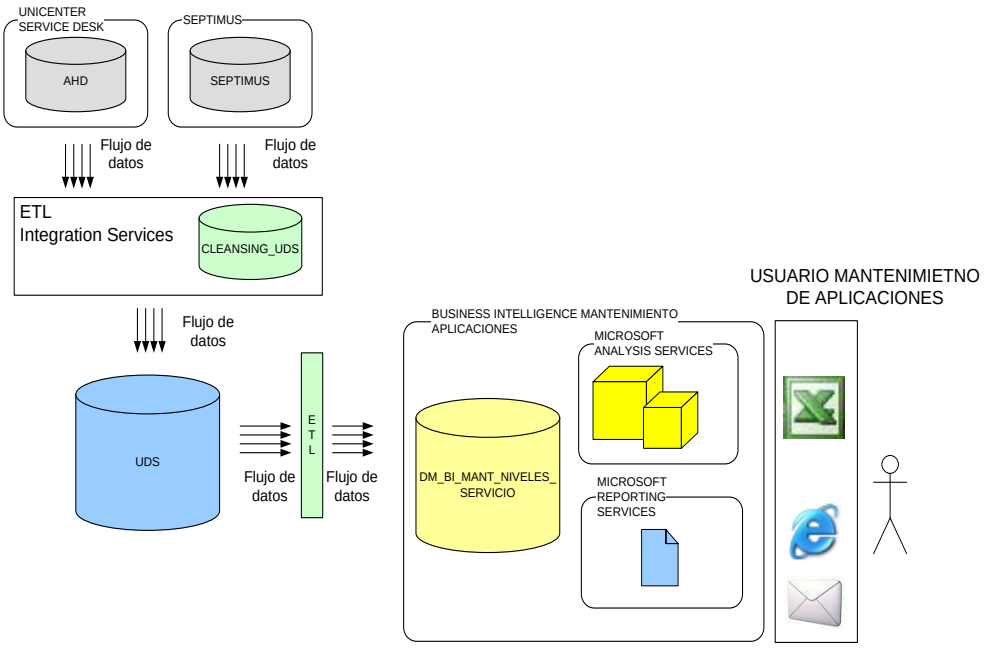

Figura No 3. Modelo conceptual de la solución en la segunda y tercera iteración.

Finalmente, el modelo conceptual de la solución, quedó conformado por los siguientes elementos:

- UNICENTER SERVICE DESK. Sistema de origen de datos.
- AHD. Nombre de la BD de Unicenter Service Desk
- SEPTMUS. Sistema y nombre de la BD de origen.
- ETL. Proceso de extracción, transformación y carga de datos.
- CLEANSING_UDS. BD de integración utilizada para depuración de datos, transformaciones complejas y, homologación de datos entre los sistemas de origen.
- Integration Services. Plataforma de procesos ETL de SQL Server 2008.
- UDS. Nombre de la BD del Datawarehouse de Mesa de Servicios.

- DM_BI_MANT_NIVELES_SERVICIO. Datamart del área de Mantenimiento de Aplicaciones. BD para herramientas de BI.
- Usuario Mantenimiento de Aplicaciones. Usuario de las herramientas de BI.

3.4 Modelos de bases de datos y programación de procesos ETL.

Complementariamente al diseño de arquitectura planteado, se realizó la clasificación de los objetos de bases de datos (tablas y stored procedures) para funciones específicas, desde la perspectiva de transformación de datos. La figura No. 4, ilustra el flujo de los datos en los procesos ETL, desde el origen hasta el Datamart. Cada una de las columnas del diagrama, representa las bases de datos; sus elementos son explicados a continuación:

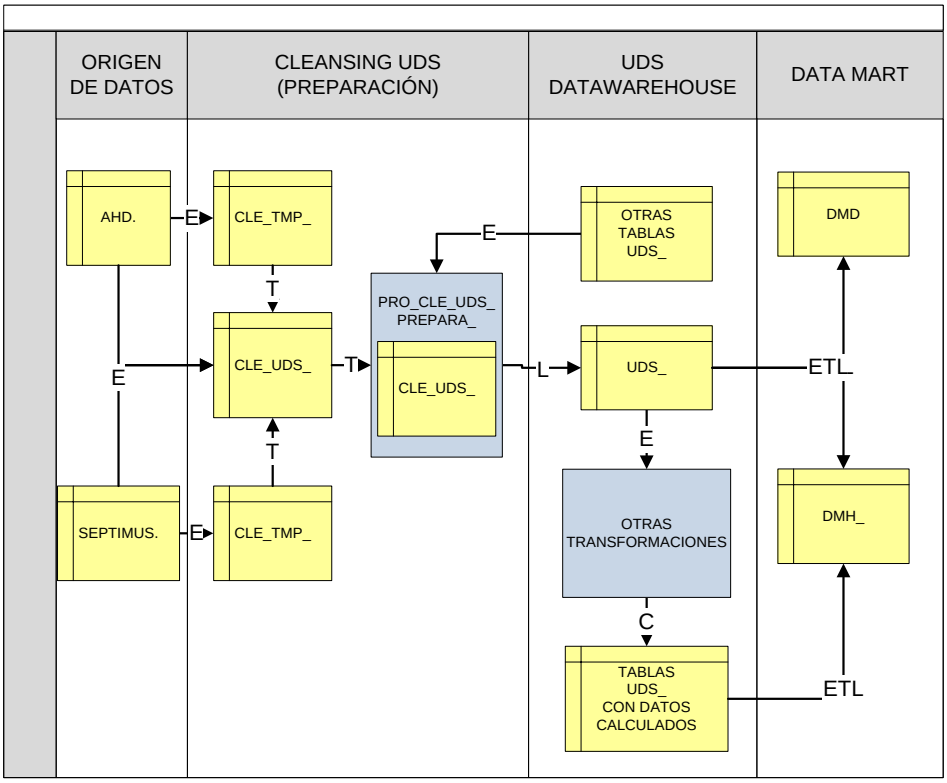

Figura No 4. Flujo de los datos en los procesos ETL.

- **AHD, SEPTIMUS.** Tablas en las bases de datos de los sistemas de origen.
- **CLE_TMP_*.** Tablas de la BD CLEANSING_UDS, con prefijo CLE_TMP_ que corresponden a tablas, con estructura igual a la del origen. Se utilizan estas tablas para evitar la ejecución de varias consultas similares al origen de datos; a partir de estas tablas se extraen nuevamente los datos, localmente, para que entren en el proceso de preparación, transformación y formato, posibilitando así, la carga en la tabla final. No en todos los casos se utiliza una tabla de estas características, sino que, cuando la tabla de origen y la de destino tienen una correspondencia una a una y, no es necesario aplicar filtros y transformaciones complicadas, entonces los datos son cargados directamente a la tabla con el prefijo CLE_UDS_.
- **CLE_UDS_*.** Tablas de la BD CLEANSING_UDS con prefijo CLE_UDS_. La estructura de estas tablas es similar a las tablas finales de los datos, es decir, a las tablas UDS_, pero con características particulares como: omisión de condicionantes y uso de tipo de dato caracter en todos los campos (incluso numéricos y fechas). Estas características permiten

depurar, homologar y preparar los datos para el formato final. Es importante la existencia de estas tablas intermedias para facilitar la integración de los sistemas al Datawarehouse; la integración de nuevos sistemas que, contengan información relativa y complementaria a la existente en el Datawarehouse, debe ser depurada y homologada para mantener las características de integración y orientación a entidades.

- **PRO_CLE_UDS_PREPARA_*.** PROCEDIMIENTOS ALMACENADOS con el prefijo PRO_CLE_UDS_PREPARA, que denota la preparación de los datos de una tabla”CLE_UDS_”, para la tabla final en el Datawarehouse. El procedimiento puede requerir datos complementarios en la depuración y transformación de datos, por lo cual, puede consultar directamente a la BD Datawarehouse UDS.
- **UDS_*.** Tabla del Datawarehouse de Mesa de Servicios en la BD UDS, con el prefijo UDS_. Almacena, de manera histórica, los datos consolidados y depurados en el formato final.
- **OTRAS TRANSFORMACIONES.** A partir de los datos cargados en las tablas UDS, se realizan otras transformaciones de información, creándose tablas de datos calculados a partir de otros cargados, sin descartar el uso de éstos, sino más bien como un proceso de explotación interna de datos, para que sean consumidos de forma más sencilla, mediante la ejecución de consultas o el desarrollo de herramientas de BI.
- **DMD_*.** Tablas de la BD DM_BI_MANT_NIVELES_SERVICIO que tienen una estructura dimensional para ser utilizada en informes automáticos y en modelos multidimensionales de datos OLAP.
- **DMH_*.** Tablas de registros de hechos de la BD DM_BI_MANT_NIVELES_SERVICIO, que se utilizan para generación de informes automáticos y procesamiento de datos, en conjunción con los modelos multidimensionales OLAP.

4 RESULTADOS

4.1 Uso y generación de recursos de información versus ciclos de desarrollo.

La incorporación de un sistema de origen de datos, en el segundo y tercer ciclo de desarrollo, permitió la generación de cubos de información y la automatización de informes de manera muy notable, con un resultado final de 12 informes hasta la tercera iteración indicada en este artículo. En consecuencia, se redujo la generación de informes de manera manual. En la figura No. 5, se puede apreciar la tendencia de uso y generación de recursos de información en los ciclos iterativos de desarrollo.

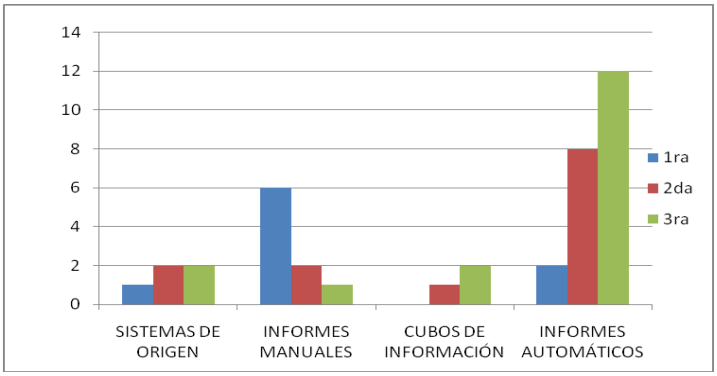

Figura No 5. Recursos de información versus ciclos de desarrollo.

4.2 Crecimiento del número de objetos en las bases de datos de la solución.

Se puede apreciar la característica de escalabilidad de la arquitectura para el desarrollo de aplicaciones de BI, entre la segunda y tercera iteración de desarrollo; el número de objetos del Datamart, con una tendencia de crecimiento mayor a su similar, relacionada al número de objetos de la base de Integración. Esta tendencia refleja que, el diseño de arquitectura realizado, permitió incorporar aplicaciones de BI, mediante la adición de datos al Datamart, el cual explota datos pre-existentes en un Datawarehouse, independientemente de la integración de nuevos sistemas de origen de datos (ver figura 6).

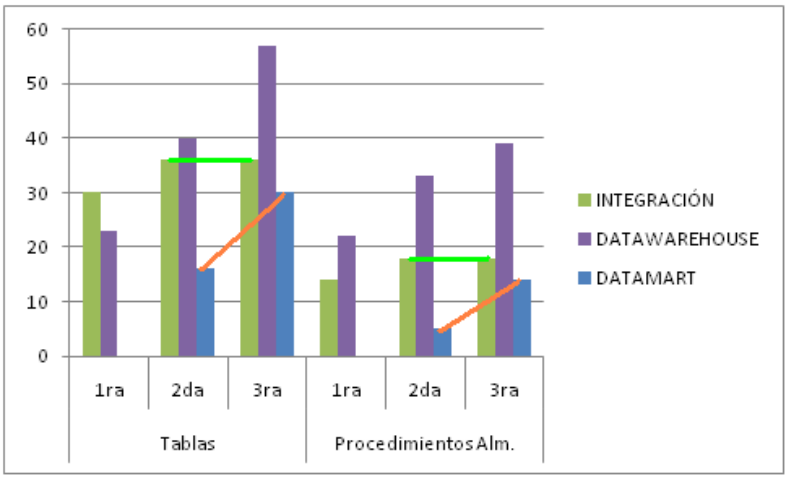

Figura No 6. "Número de objetos de bases de datos en ciclos de desarrollo".

Los datos que generan las tendencias ilustradas en la figura 6, se encuentran en la Tabla No 1.

Tabla 1. "Número de objetos de bases de datos en ciclos de desarrollo"

	Tablas			**Procedimientos Alm.**		
Base de datos / Iteración	**1ra**	**2da**	**3ra**	**1ra**	**2da**	**3ra**
INTEGRACIÓN	30	36	36	14	18	18
DATAWAREHOUSE	23	40	57	22	33	39
DATAMART	0	16	30	0	5	14

5 TRABAJOS RELACIONADOS

De la investigación realizada se puede destacar al artículo publicado por Edgard Benitez-Guerrero et al., en [5], en donde explica el desarrollo de un Datawarehouse como un proceso evolutivo utilizando el enfoque "Whes", que se centra en la adaptación de modelos multidimensionales de datos a las necesidades de cambio en los esquemas de análisis. En el artículo citado, se hace referencia específicamente al modelamiento utilizando el Lenguaje Multidimensional de Datos (MDL) sobre una serie de componentes basados en JavaBeans. Se considera apropiado el nivel de detalle especificado para la adaptación de modelos multidimensionales, sin embargo, no se generalizan los conceptos que se aplican, para que puedan extenderse a las diferentes plataformas de BI.

La escalabilidad, como una característica de un Datawarehouse, se menciona de manera genérica en diferentes publicaciones, haciendo referencia a los siguientes aspectos importantes y complementarios a lo presentado en este artículo: la arquitectura de procesamiento a nivel de hardware [6], el tamaño de lógico de los datos en los discos duros y el número de usuarios que acceden a las bases de datos a realizar consultas [7].

En las publicaciones, acerca de procesos de desarrollo de BI, las metodologías de desarrollo por ciclos iterativos son ampliamente aceptadas, calificando al proceso de desarrollo como incremental [1], [7] y evolutivo [2], [5]; lo cual justifica la necesidad de un diseño arquitectural escalable.

6 CONCLUSIONES Y TRABAJO FUTURO

El trabajo realizado permite concluir, en base a los resultados expuestos, que el desarrollo de un Datawarehouse con arquitectura escalable y, aplicando una metodología de desarrollo por ciclos iterativos, permite la integración de nuevos sistemas, fuentes de datos y, el desarrollo de aplicaciones de BI, de manera progresiva e independiente. Las aplicaciones de BI, sean éstas, informes automáticos y cubos de información, reducen la generación manual de información; ésta, puede ser generada, de manera automática, desde el primer ciclo de desarrollo, incrementando su alcance en los ciclos posteriores.

Como trabajo futuro, se evaluará el diseño arquitectural de la implementación del Datawarehouse de Mesa de Servicios de la empresa Tata Consultancy Services, con la integración de tres sistemas "fuente de datos". Adicionalmente, se realizará también el análisis de rendimiento de los procesos de un Datawarehouse relacionado al crecimiento de la solución, como una temática complementaria para ampliar el diseño de arquitectura propuesto, con una orientación a la optimización de rendimiento.

REFERENCIAS

[1]. Barreto Stein, Karla Vanessa. (Febrero 2006), "Implementación incremental para data warehouse". Revista electrónica GestioPolis. http://www.gestiopolis.com/canales6/ger/data-warehouse.htm

[2]. Dataprix.com, "Fases de implantación de un Data Warehouse". http://www.dataprix.com/fases-de-implantacion-de-un-data-warehouse

[3]. Herrera Cristhian, "Todo lo que querías saber sobre DatawareHouse (III) – Arquitectura del Datawarehouse". http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=datawarehouse3#2.9.2.6.6. Escalabilidad%20de%20usuarios%20en%20masa%7Coutline

[4]. López Requena, Martín Luis, "Microsoft Certified Trainer". (Málaga, Ago 2006). Microsoft Solutions Framework. http://www.malagadnug.org/ficheros/MSFMartinLuisReq.pdf

[5]. Benitez-Guerrero Edgard, Collet Christine, Adiba Michel, "El enfoque Whes en la evolución de los depósitos de datos". ISSN 1665-5745. México. http://redalyc.uaemex.mx/redalyc/src/inicio/ArtPdfRed.jsp?iCve=73000211

[6]. Dataprix.com, "Componentes a tener en cuenta a la hora de construir un DataWarehouse". http://www.dataprix.com/componentes-tener-en-cuenta-la-hora-de-construir-un-data-warehouse

[7]. Herrera Cristhian, "Todo lo que querías saber sobre DatawareHouse (III) – Repositorio del Datawarehouse". http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=datawarehouse3#2.9.2.6.6. Escalabilidad%20de%20usuarios%20en%20masa%7Coutline

[8]. Rainardi Vincent. (2008), "Building a Data Warehouse: With Examples in SQL Server. Apress".

[9]. Silvers Fon. (2008), "Building and Maintaining a Data Warehouse". Taylor & Francis Group.