Diseño de un Almacén de datos basado en Data Warehouse Engineering Process (DWEP) y HEFESTO

Castelán García Leopoldo, Ocharán Hernández Jorge Octavio
Maestría en Ingeniería de Software, Facultad de Estadística e Informática
Universidad Veracruzana
lcastelan@uv.mx, jocharan@uv.mx

Resumen. El desarrollo de un almacén de datos se basa en el diseño de un modelo conceptual que incluye tanto los requisitos de información de los usuarios así como las fuentes de datos operacionales. A partir de éste se obtiene un modelo lógico basado en una tecnología de base de datos específica que guía la implementación. Actualmente muchas de las propuestas no definen mecanismos para estructurar de manera sistemática el desarrollo de un almacén de datos, convirtiéndolo en una tarea compleja y artesanal. Para dar solución a este problema, el trabajo expuesto propone el uso de la metodología HEFESTO para definir la arquitectura de datos y DWEP para facilitar el diseño y construcción de un almacén de datos. El propósito es combinar estas metodologías ayudando a reducir los problemas que resultan de seguir un proceso sin comprenderlo.

Palabras clave-Almacén de Datos, DWEP, PU, HEFESTO, UML.

1. Introducción

1.1. Almacén de Datos

Los Almacenes de Datos (AD) o *Data Warehouse* son una colección de datos históricos que sirven de apoyo a la toma de decisiones para mejorar un proceso de negocio. Bill Inmon lo define como [1]:

> "…un almacén de datos es una colección de datos orientados a temas, integrados, no-volátiles y variables en el tiempo, organizados para soportar necesidades empresariales…".

Un AD se dice que es orientado a temas porque la información se clasifica en base los intereses de la organización. Es integrado cuando sus datos provienen de diversas fuentes, es decir, que son producidos por distintos departamentos y aplicaciones, tanto internas como externas. Estos datos deben ser consolidados en una instancia antes de ser agregados al AD. Este proceso se conoce como Extracción, Transformación y Carga de Datos (*Extraction, Transformation and Load* - ETL). La integración de datos resuelve diferentes tipos de problemas relacionados con las convenciones de nombres, unidades de medidas, codificaciones, fuentes de datos múltiples, entre otros. Son no-volátiles porque la información es útil para el análisis y la toma de decisiones solo cuando es estable. Los datos operacionales varían constantemente, en cambio, los datos una vez que entran en el AD no cambian, y por último se dice que son variables en el tiempo debido al gran volumen de información que se maneja cuando se realiza una consulta, los resultados deseados tardarán en originarse, este espacio de tiempo entre la búsqueda de datos y el resultado es del todo normal en este ambiente y es precisamente por ello, que la información que se encuentra dentro del AD se denomina de tiempo variable [2].

Un AD permite tener el manejo y control de la información, así se tiene asegurada una vista única de los datos de fuentes funcionalmente distintas (bases corporativas, bases propias, sistemas externos, etc.). Los usuarios finales no tienen la necesidad de aprender a utilizar diferentes sistemas de acceso y manipulación de los datos. Facilita la

comprensión de los datos al transformarlos en información útil y ayuda a la organización a responder preguntas esenciales para la toma de decisiones que le permitan obtener ventajas competitivas y mejorar su posición en el mercado.

1.2. Importancia del almacén de datos

Para Inmon la importancia de un AD se define en tres dimensiones [1]:

1. Mejora la entrega de información: información que sea completa, correcta, consistente, oportuna y accesible.
2. Facilita el proceso de toma de decisiones: con un gran respaldo de información se puede tomar decisiones rápidamente; así también, la gente de negocios adquiere confianza en sus propias decisiones y logra un mayor entendimiento de los impactos de éstas.
3. Impacto positivo sobre los procesos empresariales: cuando la gente tiene acceso a una mejor calidad de información, la empresa mejora eliminando los retardos en los procesos que resultan de información incorrecta, inconsistente; logrando así una integración y optimización de procesos empresariales a través del uso compartido e integrado de las fuentes de información.

1.3. Problemática de los sistemas basados en almacén de datos

En distintas fuentes [5] [6] se observa que entre el 40% y 50% de los sistemas basados en AD fallan o son abandonados por aspectos que no se tienen considerados desde un principio. Según Larry Poole [7] las razones de por qué estos sistemas fallan son:

1. No cuentan con un líder que entienda el valor del proyecto y esté dispuesto a asignar los recursos apropiados.
2. Los requisitos son pobres, ya que no se involucran a los usuarios en las discusiones.
3. Los diseños son pobres debido a que los requisitos son deficientes y el tiempo de modelado es limitado.
4. No se tiene entrenamiento a usuarios finales para el uso adecuado de la solución, para llevar a buen término la implantación del proyecto.
5. Se cree que con la solución inicial se termina el proyecto descuidando su mantenimiento o crecimiento.
6. Se escogen inadecuadamente las herramientas a utilizar.
7. Muchos proyectos arrancan pensando en una solución final pero sin saber el tiempo y no se estima el trabajo que requiere, o si la solución es compleja.
8. La solución no cumple con los objetivos.

1.4. Arquitectura de un almacén de datos

La Arquitectura de un AD está formada por diversos elementos que interactúan entre sí y que cumplen una función específica dentro del sistema, como se muestra en la Figura 1, Los componentes de una AD según Bernabeu son [2]:

- OLTP (On Line Transaction Processing), representa toda aquella información transaccional que genera la organización diariamente y las fuentes externas.
- LOAD MANAGER. Los ETL se encargan de extraer los datos desde los OLTP para manipularlos, integrarlos, transformarlos y posteriormente cargar los resultados obtenidos en el AD, es necesario contar con un sistema que se encargue de ello.

- DW MANAGER. Su finalidad es transformar e integrar los datos fuentes y de almacenamiento intermedio en un modelo adecuado para la toma de decisiones. Permitiendo realizar todas las funciones de definición y manipulación del depósito de datos, para poder soportar todos los procesos de gestión del mismo.
- QUERY MANAGER. Este componente realiza las operaciones necesarias para soportar los procesos de gestión y ejecución de consultas relacionales, propias del análisis de datos, recibe las consultas del usuario, las aplica a la estructura de datos correspondiente y devuelve los resultados obtenidos.
- HERRAMIENTAS Y CONSULTAS DE DATOS. Son los sistemas que permiten al usuario realizar la exploración de datos del AD. Básicamente constituyen el nexo entre el depósito de datos y los usuarios.
- USUARIOS. Son aquellos que se encargan de tomar decisiones y de planificar las actividades del negocio.

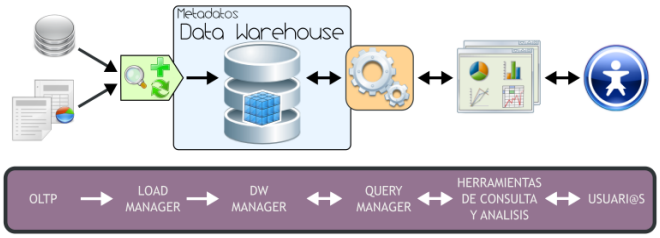

Figura 1. Arquitectura de un Almacén de Datos tomada de la metodología HEFESTO [2].

Esta arquitectura de AD opera de la siguiente manera:

Los datos se extraen de aplicaciones, bases de datos, archivos, entre otros. Esta información generalmente reside en diferentes tipos de sistemas, orígenes y arquitecturas con diferente formatos, los datos son integrados, transformados y limpiados, para luego ser cargados en el AD, la información se estructura en cubos multidimensionales para responder a consultas dinámicas con una buena presentación. Los usuarios acceden a los cubos, utilizando diversas herramientas de consulta, exploración, análisis, reportes, etc.

Un cubo multidimensional, representa o convierte los datos planos que se encuentran en filas y columnas, en una matriz de N dimensiones. Los objetos más importantes que se pueden incluir en un cubo multidimensional son los indicadores, atributos y jerarquías, de esta manera los atributos existen a lo largo de varios ejes o dimensiones, y la intersección representa el valor que tomará el indicador que se está evaluando.

2. Propuesta

2.1. Motivación

Los problemas más frecuentes de los sistemas basados en AD se encuentran en la recolección de requerimientos, el análisis y el diseño [3], debido a que no sigue una metodología estándar para su desarrollo.

La metodología denominada proceso de ingeniería para el desarrollo de almacenes de datos (DWEP) [4] la cual está basada en el proceso unificado (PU), abarca los flujos de trabajo de requerimientos, análisis, diseño, pruebas, mantenimiento y revisiones posteriores al desarrollo; sus principales características son que es iterativa, dirigida

por casos de uso y se basa en las etapas de desarrollo de PU, utilizando UML como lenguaje para modelado gráfico.

Por otro lado, en el componente del proceso de arquitectura de datos se tiene a HEFESTO [2] una metodología cuya propuesta se fundamenta en una amplia investigación, comparación de metodologías existentes y la experiencia en la elaboración de almacenes de datos. La ventaja principal de esta metodología es que específica puntualmente los pasos a seguir en cada fase a diferencia de otras metodologías que mencionan los procesos, más no explican cómo realizarlos.

2.2. Objetivo

Crear un proceso de desarrollo para el diseño de un AD basado en la integración de la metodología proceso de ingeniería para el desarrollo de almacenes de datos (DWEP) y la metodología HEFESTO, partirá de la recolección de requerimientos y necesidades de información del usuario, y concluirá en la elaboración de un modelo conceptual, lógico y físico con la finalidad de poder facilitar el trabajo que implica la elaboración de un AD desde su inicio.

2.3. Fases del DWEP y proceso unificado

El PU como se muestra en la Figura 2, es un marco de desarrollo compuesto de cuatro fases, cada una de ellas a su vez dividida en una serie de iteraciones que ofrecen como resultado un incremento del producto desarrollado, que añade o mejora las funcionalidades del sistema en desarrollo [8].

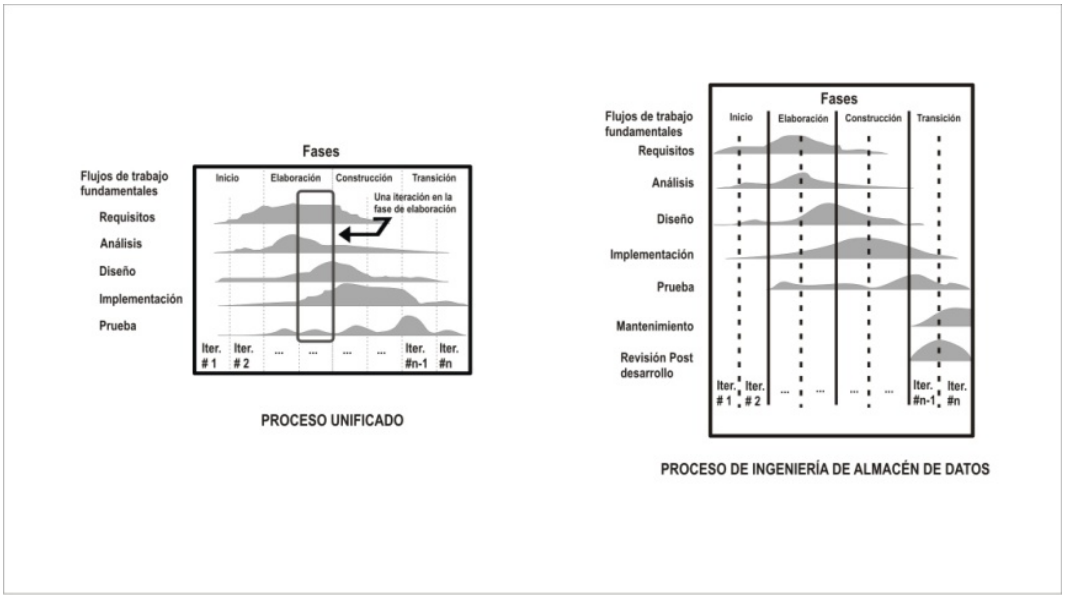

Figura 2. El proceso unificado [8] y proceso de ingeniería de almacenes de datos [4]

2.3.1. Fase de inicio: El objetivo de esta fase es analizar el proyecto para justificar su puesta en marcha, para lograrlo se realiza una descripción general del proyecto, se detectan los riesgos críticos y se establecen la funcionalidad básica del software con una descripción de la arquitectura candidata.

2.3.2. Fase de elaboración: Una vez finalizada la fase de inicio, se pretende formar una arquitectura sólida para la construcción del software. En esta fase se busca establecer la base lógica de la aplicación con los casos de uso definitivos y los artefactos del sistema que lo componen.

2.3.3. Fase de construcción: Se inicia a partir de la línea base de arquitectura que se especificó en la fase de elaboración y su finalidad es desarrollar un producto listo para la operación inicial en el entorno del usuario final.

2.3.4. Fase de transición: Una vez que el proyecto entra en la fase de transición, el sistema ha alcanzado la capacidad operativa inicial. Esta fase busca implantar el producto en su entorno de operación.

2.4. Flujos de trabajo aplicados al proceso DWEP

En términos generales para el PU y el DWEP un flujo de trabajo es un conjunto de actividades realizadas en un área determinada cuyo resultado es la construcción de artefactos (un texto, un diagrama, una página Web, código en lenguaje de programación, etc.).

2.4.1. Requerimientos. Durante este flujo de trabajo, los usuarios especifican las medidas y agregaciones más interesantes, el análisis dimensional, consultas usadas para la generación de reportes periódicos y frecuencia de la actualización de los datos. El PU sugiere el uso de casos de uso [9] [4]. Esto ayuda a comprender el sistema y obtener los requisitos y funciones para la solución. Además establece como deben ser las interacciones del sistema.

2.4.2. Análisis. Tiene como objetivo mejorar la estructura y los requisitos obtenidos en la etapa de requerimientos. En esta etapa se documentan los sistemas operacionales preexistentes que alimentaran el AD. El PU propone el uso del diagrama de clase [4] [9].

2.4.3. Diseño. Al final de este flujo de trabajo, está definida la estructura del AD. El principal resultado de este flujo de trabajo es el modelo conceptual del AD.

2.4.4. Implementación. Durante este flujo de trabajo, el AD es construido y se empiezan a recibir datos de los sistemas operaciones, se afina para un funcionamiento optimizado, entre otras tareas.

2.4.5. Pruebas. El objetivo de este flujo de trabajo es verificar que la aplicación funcione correctamente, realizar las pruebas y analizando los resultados de cada prueba.

2.4.6. Mantenimiento. Un AD es un sistema que se retroalimenta constantemente. El objetivo de este flujo de trabajo es definir la actualización y carga de los procesos necesarios para mantener el AD.

2.4.7. Revisiones post desarrollo. Esto no es un flujo de trabajo de las actividades de desarrollo, sino un proceso de revisión para la mejora de proyectos a futuro. Si hacemos un seguimiento del tiempo y esfuerzo invertido en cada fase es útil en la estimación de tiempo y de las necesidades para generar los requisitos para desarrollos futuros.

2.5. Etapas de la metodología HEFESTO

El objetivo de la metodología HEFESTO [2] es que de manera metódica y sencilla podamos comprender cada paso que se ejecuta para entender el por qué del proceso y no caer en el error de seguir un método sin saber que se está haciendo, guiándose por pasos lógicos relacionados durante todas las etapas del proceso. La metodología HEFESTO, puede ser utilizada en cualquier ciclo de vida que no requiera fases extensas

de requerimientos y análisis, con el fin de entregar una implementación que cumpla con una parte de las necesidades proporcionadas por el usuario.

2.5.1. Descripción

La metodología HEFESTO inicia con la recolección de las necesidades de información de los usuarios obteniendo así las preguntas claves del negocio. Luego, se deben identificar los indicadores resultantes y sus perspectivas de análisis, mediante las cuales se construirá el modelo conceptual de datos del AD, se analizarán las base de datos de los OLTP o fuentes externas para señalar las correspondencias con los datos fuentes y seleccionar los campos de estudio de cada perspectiva. Una vez hecho esto, se pasará a la construcción del modelo lógico, tomando en cuenta las jerarquías que intervendrán. Por último, se definirán los procesos de carga, transformación, extracción y limpieza de los datos fuente.

A continuación podemos ver cada uno de los pasos correspondientes a la metodología HEFESTO y que resumen la descripción antes mencionada, sobre los procesos que se siguen.

1. Análisis de Requerimientos
    a. Identificar preguntas
    b. Identificar indicadores y perspectivas de análisis
    c. Modelo conceptual
2. Análisis de los OLTP
    a. Determinación de Indicadores
    b. Establecer correspondencia
    c. Nivel de granularidad
    d. Modelo conceptual ampliado
3. Modelo lógico del Almacén de Datos
    a. Tipo del modelo lógico del Almacén de Datos
    b. Tabla de dimensiones
    c. Tabla de hechos
    d. Uniones
4. Procesos ETL

2.5.2. Pasos y aplicación metodológica

2.5.2.1. Análisis de Requerimientos. Se identifican los requerimientos del usuario con el fin de entender los objetivos de la organización, haciendo uso de técnicas y herramientas, como la entrevista, la encuesta, el cuestionario, la observación, el diagrama de flujo y el diccionario de datos, obteniendo como resultado una serie de preguntas que se deberán analizar con el fin de establecer cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del AD. Finalmente se realizará un modelo conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.

2.5.2.2. Análisis de los OLTP. Tomando en cuenta el resultado obtenido en el paso anterior se analizarán las fuentes OLTP para determinar cómo serán calculados los indicadores con el objetivo de establecer las respectivas correspondencias entre el modelo conceptual y las fuentes de datos. Luego, se definirán qué campos

se incluirán en cada perspectiva y finalmente, se ampliará el modelo conceptual con la información obtenida en este paso.

2.5.2.3. Modelo lógico del Almacén de Datos. Como tercer paso, se realizará el modelo lógico de la estructura del AD, teniendo como base el modelo conceptual. Para esto, debemos definir el tipo de representación de un AD que será utilizado, posteriormente se llevarán a cabo las acciones propias al proceso, para diseñar las tablas de dimensiones y de hechos. Por último, se realizarán las uniones pertinentes entre estas tablas.

2.5.2.4. Procesos ETL. El último paso de la metodología HEFESTO es probar los datos, a través de procesos ETL. Para realizar la compleja actividad de extraer datos de diferentes fuentes, para luego integrarlos, filtrarlos y depurarlos, se podrá hacer uso de software que facilita dichas tareas, por lo cual este paso se centrará solo en la generación de las sentencias SQL que contendrán los datos que serán de interés.

Antes de realizar la carga de datos, es conveniente efectuar una limpieza de los mismos, para evitar valores faltantes y anómalos. Al generar los ETL, se debe tener en cuenta cuál es la información que se desea almacenar en el AD, para ello se pueden establecer condiciones adicionales y restricciones. Estas condiciones deben ser analizadas y realizadas con mucha prudencia para evitar pérdidas de datos importantes.

2.6. Modelado de proceso

Para aplicar el proceso de desarrollo propuesto en este trabajo se definen tres niveles de abstracción: conceptual, lógico y físico. El nivel conceptual representa las interacciones entre las entidades y las relaciones, el nivel lógico describe con tanto detalle como sea posible los datos, sin tener en cuenta cómo estén físicamente en la base de datos y el nivel físico incluye la especificación técnica, después de la reglas del negocio para determinar el diseño del AD.

Esta propuesta se encuentra en la definición del nivel conceptual y en una etapa posterior la definición de los demás niveles. El objetivo más importante de este nivel es la representación de las principales propiedades sin tener en cuenta detalles específicos de tecnología alguna de bases de datos, posibilitando así la independencia del modelo respecto de la plataforma en la cuál sea implementado. Una vez que los requisitos de información se han especificado, debe derivarse un modelo conceptual inicial.

En esta etapa inicial de desarrollo (conceptual) se crean las bases del proyecto, definiendo el alcance, plan inicial, la visión del negocio junto con las metas y la justificación del proyecto. Los requerimientos iniciales son capturados a través de casos de uso.

Se define un equipo de trabajo para el plan de proyecto que maneja las dependencias principales y la estrategia general, mientras que los planes más refinados manejan las tácticas propias de las situaciones en cuestión de cada iteración.

Este nivel finaliza con la existencia de un alcance definido, plan de desarrollo, análisis de riesgos donde los clientes concuerden con lo anterior.

En la Figura 3 se representa de forma general el nivel conceptual.

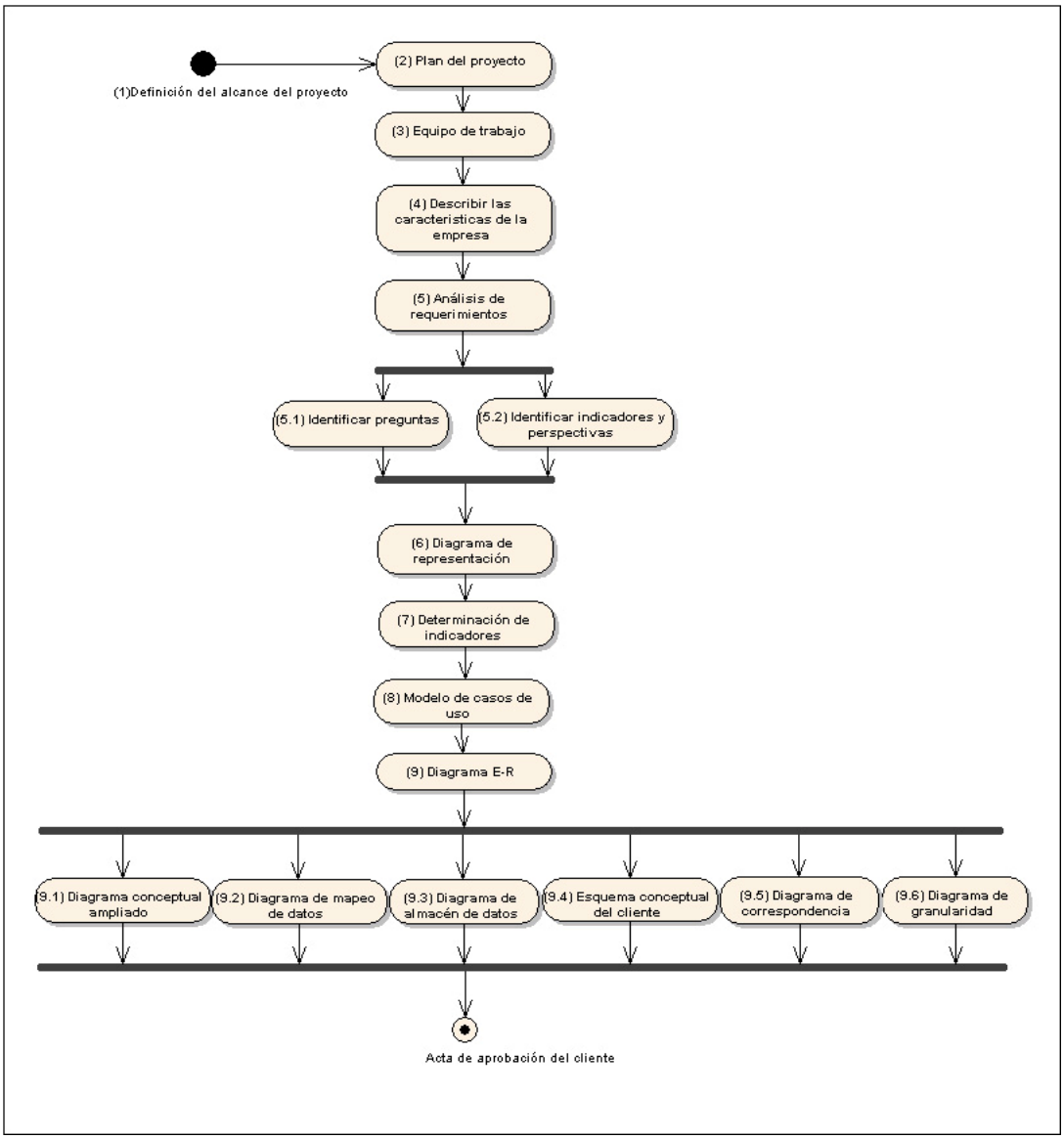

Figura 3. Esquema propuesto de nivel conceptual

Los pasos para el desarrollo de un AD son los siguientes:

1. Se debe iniciar con la definición del alcance, para evitar grandes problemas en sus fases, se recomienda usar la técnica de descomposición de tareas, y para su representación normalmente se usa la forma tipo organigrama.
2. Se debe redactar un plan inicial que contempla la visión del negocio junto con las metas y la justificación del proyecto.
3. Se deberán definir roles dependiendo de las actividades de cada usuario.
4. Se debe realizar un documento que describa las características de la empresa que contemple datos que la identifiquen, objetivos, políticas, estrategias, definición de los procesos y la relación de las metas con el AD.
5. Como siguiente paso de realiza la identificación de preguntas mediante cuestionarios, entrevistas u observaciones sobre los objetivos del proyecto, con la finalidad de identificar los indicadores y perspectivas de las cuales partirá el análisis de diseño.

6. Una vez que se han establecido las preguntas claves, se debe proceder a su descomposición para descubrir los indicadores que representan lo que se desea analizar concretamente y las perspectivas que se refieren a los objetos mediante los cuales se quiere examinar los indicadores, realizando un detalle de estos.
7. A partir de los indicadores y perspectivas obtenidas en el paso anterior se debe construir un modelo conceptual.
8. Se realiza la representación de los casos de uso que representan los requerimientos del AD.
9. La fuente de datos del AD es el modelo entidad-relación representado en diagramas UML. Al transformar el modelo ER a un diagrama de clases se transforma en el esquema conceptual de la fuente de datos (SCS). En el diseño conceptual del AD según DWEP se realiza los diagramas: conceptual de datos (SCS), mapeo de datos (DMS), almacén de datos (DWCS) y el esquema conceptual del cliente (CCS. El esquema conceptual de la base de datos se realiza en tres niveles que consisten en la realizar un detalle de cómo se encuentra integrada.
10. Este nivel finaliza con la redacción de un acta de aprobación del cliente respecto a la definición de indicadores, fuente de datos y el modelo conceptual.

3. Conclusiones

Este trabajo presenta una aproximación de modelo de desarrollo que se realiza, mediante la integración de HEFESTO Y DWEP. Este modelo de desarrollo consta de tres niveles en los cuales se integrada cada una de las actividades a realizar y los artefactos por cada nivel. La propuesta se centra en los pasos del nivel conceptual debido al tiempo que requiere para el análisis de integración de los artefactos generados por ambas metodologías para la definición del modelo lógico y físico que no ha sido posible definir de una manera clara.

El uso de este modelo posibilita que el desarrollo de un AD esté bien estructurado, permitiendo considerar varios aspectos muy importantes en el desarrollo de este tipo de sistemas, como son la recopilación de requisitos de información y las OLTP, los procesos ETL, el AD como estructura física y la explotación de la información.

La necesidad que existe en las empresas por compartir y obtener un conocimiento sobre el mundo de información que guardan para poder realizar acciones para la toma de decisiones, hace posible el desarrollo de AD que sean guiados por procesos bien definidos y sustentados en metodologías que hacen uso de estándares como UML para el modelado, que garantiza que el diseño de cada elemento sea definido de forma única y poder ofrecer sistemas de calidad basados en una ingeniería de software que esté sustentada en la elaboración de artefactos y seguimiento de procesos que sean claros que permiten entender el por qué realizar dicha actividad al momento de su diseño.

Este modelo de desarrollo será propuesto para la elaboración del sistema de indicadores de la Universidad Veracruzana, con la idea de evaluar cada uno de los niveles considerando los artefactos que son necesarios; conocer los resultados obtenidos para validar si es un proceso de ayuda para el cumplimiento de los objetivos o si es necesario un nuevo planteamiento en su desarrollo.

Referencias

1. Inmon, W. Building the data warehouse. s.l. : Wiley & Sons, 2002.
2. Dario, Bernabeu Ricardo. DATA WAREHOUSING:Investigación y Sistematización de Conceptos. Córdoba, Argentina : Licencia de Documentación Libre de GNU, 2009.
3. B. Husemann, J. Lechtenborger, G. Vossen. Conceptual Data Warehouse Desing, Proceeding of the International Workshop on Design and Management of Data Warehouses. Sweden : StockHolm, 2000.
4. Lujan, S. Data WareHouse Desig with UML, PHD. Thesis Universidad de Alicante. Alicante, España : s.n., 2005.
5. consortiwn, Custer. 41% HAVE EXPERIENCED DATA WAREHOUSE PROJECT FAILURES. [En línea] http://www.cutter.com/research/2003/edge030218.html.
6. Madsen, Mark. A 50% Data Warehouse Failure Rate is Nothing New. [En línea] Mark Madsen. http://it.toolbox.com/blogs/bounded-rationality/a-50-data-warehouse-failure-rate-is-nothing-new-4669.
7. Poole, Larry. 8 Reasons Why Business Intelligence Initiatives Fail. [En línea] www.xyber.net/8Reasons.doc.
8. (OMG)., Object Management Group. Unifie Modeling Language (UML),version 2.0. [En línea] http://www.uml.org/.
9. Mora, J. Trujillo and L. Data WareHouse Desig with UML, PHD. Thesis Universidad de Alicante. Alicante, España : s.n., 2005.
10 . Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El proceso unificado de desarrollo de software. Madrid : Addison Wesley, 2000.