



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

PROCESO DE TITULACIÓN

NOVIEMBRE 2019 – FEBRERO 2020

INGENIERÍA EN SISTEMAS

TÍTULO DE INGENIERO EN SISTEMAS

PROPUESTA TECNOLÓGICA

TEMA

Sistema de toma de decisiones nivel administrativo para la distribuidora de insumos agrícolas
“AGROLIVE”

EGRESADO

Bernal Villamar Héctor Omar

TUTOR

ING. España Lón Angel Rafael

AÑO

2020

INTRODUCCIÓN

El entorno actual como se desempeñan las empresas ha provocado un cambio en la manera de gerenciar y aprovechar al máximo el recurso de información para la producción de servicios que se ofrecen. La generación y almacenamiento de información de los sistemas tradicionales no pueden procesar tal cantidad y emitir los análisis pertinentes, producto de la demasía de información se vuelve más complejo obtener lo que realmente es útil y en muchas ocasiones no se consideran datos indispensables para su exploración y posterior estudio.

Es un problema administrar grandes cantidades de información, por este motivo se ha venido desarrollando nuevos sistemas que mitiguen esta falencia, permitiendo tener conocimiento dónde y cómo se organiza los datos con el propósito de dar soporte a la toma de decisiones de los ejecutivos de una organización. Entre los sistemas desarrollados surge el termino Data warehouse. Basado en la inteligencia empresarial (BI-Inteligencia de negocios) es una amplia categoría de programas de aplicaciones y tecnologías para recopilar, almacenar, analizar y facilitar el acceso a los datos de diversas fuentes, lo que proporciona a los usuarios de la empresa información confiable y oportuna para su posterior análisis, mejorando así la toma de decisiones.

El presente proyecto tiene la finalidad proporcionar a los directivos una herramienta para el análisis, basándose en información íntegra y global del negocio permitiendo así analizar y responder a necesidades de planeación, predicción y toma de decisiones.

CAPÍTULO 1

DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. **Ámbito de Aplicación: descripción del contexto y hecho de interés**

La distribuidora “Agrolive” con el afán de mejorar y competir en el mercado desea actualizar sus procesos tecnológicos para ofrecer un servicio de calidad a su distinguida clientela.

El problema principal se presenta en la manera en que se obtienen los reportes de transacciones no se tiene un panorama claro de ingresos y egresos generando problemas para la gerencia que carece de información para toma de decisiones.

Debido al crecimiento continuo y a la evolución de normativas financieras en las distribuidoras de productos químicos han ido ajustando sus procesos, estructura organizacional, forma de cobertura para brindar servicio a sus clientes. Esto ha dado como resultado a que las plataformas tecnológicas igualmente evolucionen para poder soportar diversas operaciones que una institución realice, pasando de sistemas distribuidos a sistemas centralizados de diversas interfaces y bases de datos.

En el caso de Agrolive, el aplicativo fue desarrollado en una plataforma propietaria con el uso básico de estándares de programación, con poca documentación y una metodología tradicional de desarrollo. Esto dio origen a que la aplicación fuera creciendo desordenadamente, con duplicidad de información, sin un modelo relacional completo, motivo por el cual hasta la actualidad influye en la prestación de servicios, generando incidentes permanentes en el soporte del aplicativo transaccional.

Todo lo expuesto anteriormente es causa para que la producción de la información de toma de decisiones requiera un gran esfuerzo y sea en muchos casos ineficiente e insegura, teniendo que pasar por varios departamentos quienes manipulan la información de diversas índoles con gran riesgo de cometer errores para poder presentar informes ejecutivos a los diferentes niveles de mando de la institución. Además los reportes que se receptan en la gerencia son muy básicos y no se puede realizar un análisis de valores y perspectivas de las estrategias institucionales.

Por todo lo descrito, la distribuidora desea incorporar un sistema de información gerencial para la toma de decisiones a nivel administrativo, liberando así recursos al departamento de gerencia, con los cuales responder con celeridad a los cambios en el entorno y adecuar sus tácticas y estrategias para ser eficientes y poder servir a sus clientes de la mejor manera.

Metodología de investigación

El tipo de investigación para este caso está compuesto por una parte exploratoria, para toma de decisiones y de proyecto factible.

➤ **Exploratoria**

El proyecto posee propiedades de ser un prototipo o piloto además el BI no ha sido estudiado anteriormente en “Agrolive”.

➤ **Para toma de decisiones**

Gracias a las características que posee un Data warehouse que permite dar apoyo a la toma de decisiones.

1.1.1 Porque es importante

La importancia se basa en la manera que se obtendrán los resultados de las transacciones y que se tendrá un panorama claro para que el gerente pueda tomar decisiones productivas y también podrá hacer una proyección según los resultados obtenidos para definir nuevas estrategias de negocio.

(Duque Méndez & Tamayo Alzate, 2001)

1.1.2 Que lo hace diferente

Muchos tipos de datos de negocio se analizan a través de un Data warehouse. La necesidad de contar con este sistema se hace evidente cuando los requerimientos analíticos de la organización entran en conflicto con el rendimiento de las bases de datos operacionales o transaccionales. Sobre todo, debido a la ejecución de consultas complejas que muchas veces son imposibles para esas bases de datos. Por lo tanto: (Flores & Martínez Chirichian, 2017)

- ✓ Un Data warehouse (almacén de datos) se emplea para hacer el trabajo analítico, dejando las bases de datos transaccionales libres para centrarse en las transacciones.
- ✓ Tiene la capacidad de analizar datos de múltiples fuentes y puede negociar las diferencias en cuanto a esquemas de almacenamiento utilizando procesos de ETL (extracción, transformación y lectura).
- ✓ Al integrar datos de múltiples sistemas de origen, permite una visión central en toda la empresa.

- ✓ Mantiene el historial de datos incluso si los sistemas transaccionales de origen no lo hacen.
- ✓ Mejora los datos, proporcionando códigos y descripciones coherentes, flags o incluso arreglando datos erróneos.
- ✓ Presenta la información de la organización de forma coherente.
- ✓ Proporciona un único modelo de datos común para todos los datos de interés independientemente de la fuente de los datos.
- ✓ Reestructura los datos de manera que tienen sentido para los usuarios de negocios.
- ✓ Reestructura los datos de modo que ofrece un excelente rendimiento para consultas analíticas complejas, sin afectar a los sistemas operativos.
- ✓ Añade valor a las aplicaciones de negocio operativas, en especial a las de gestión de relaciones con clientes (CRM).

Un Data warehouse puede ayudar a las organizaciones a extraer el máximo valor de los datos que se generan en el día a día, permitiendo además analizarlos y compararlos con sus valores históricos y no solo mediante datos actuales.(Duque Méndez & Tamayo Alzate, 2001)

1.1.3 Contribución de la propuesta

La propuesta influirá en el potencial para el procesamiento de datos históricos brindando mayor flexibilidad y rapidez para la obtención de resultados claves para obtención de datos estadísticos en el departamento administrativo.

1.1.4 Análisis de requerimientos

El Data warehouse debe ser un gran repositorio de datos históricos y actuales que responda a dudas sobre información de ventas, créditos, productos, clientes etc.

De acuerdo a las necesidades previstas y al medio en que se desarrolla en problema los principales requerimientos que debe cumplir este proyecto son:

1.2. Establecimientos de requerimientos

Para la realización del presente sistema de toma de decisiones en la distribuidora de productos químicos Agrolive, se definen de manera general para el departamento de gerencia los siguientes requerimientos:

1.2.1 Requerimientos funcionales

Requerimiento	Data Warehouse
Código:	SIAGRO-01
Nombre/Req:	Atributo Año
Característica:	El sistema deberá permitir analizar la información por año
Req.NoFun:	SGBD
Prioridad:	Alta

Requerimiento	Data Warehouse
Código:	SIAGRO-02
Nombre/Req:	Informes -Total de facturación
Característica:	El sistema deberá poder permitir analizar el total de la facturación por artículo, año, mes, trimestre y día
Req.NoFun:	SGBD

Prioridad:	Alta
-------------------	------

Requerimiento	Data Warehouse
Código:	SIAGRO-03
Nombre/Req:	ETL-Auditoria
Característica:	Todas las tablas del sistema deberán disponer de auditoría que indique cuando se cargó o modifico cada registro.
Req.NoFun:	SGBD
Prioridad:	Alta

1.2.2 Requerimientos no funcionales

Para establecer los requerimientos no funciones se realizó un análisis del hardware y software cliente que disponía la entidad. Se realizaron las pruebas correspondientes, los resultados obtenidos y la recomendación fue, que el equipo informático actual no cumplía con las exigencias mínimas para un funcionamiento óptimo del aplicativo concluyendo así que el reemplazo de los mismos era inminente, las características de los nuevos equipos sugeridos se describen a continuación.

ID	TIPO	DESCRIPCION
1	Disponibilidad	Es necesario tener disponibilidad inmediata de la información, se debe contar siempre con la información relacionada a la última transacción finalizada a nivel de base de datos. Los tiempos de disponibilidad son 8*5 el primer año con posibilidad de escalar a 24/7 el segundo año de operación.
2	Desempeño	Se espera que el análisis de la información sea lo más eficaz posible, con tiempos de respuesta menores a 10 segundos e interoperabilidad de al menos 30 usuarios en la fase piloto, máximo de 50 usuarios el primer año y expectativa de 70 usuarios el segundo año.
3	Despliegue	El despliegue se realizará inicialmente sobre la instalación donde se encuentra el sistema SIAG de la distribuidora “Agrolive”. Las características de Hardware(Cliente) son: <ul style="list-style-type: none">✓ Motherboard H310✓ Tamaño disco duro: SATA [1 TB] 7200 RPM✓ Procesador: Intel® Core i5-8400 8va Generación✓ Capacidad de memoria RAM: 8 GB✓ S.O: Windows 10 ver. 1903 minima

--	--	--

1.2.3 Requerimientos del hardware (Servidor)

MARCA-MODELO	HARDWARE	DESCRIPCIÓN
Servidor HPE ProLiant DL380 Gen10 5220	Disco duro (HDD)	(Preferible 8 de bahías) SATA, SAS, SSD [1 TB]
	CPU	Intel Xeon-Gold 5220 (2,2 GHz/18 núcleos/125 W)
	MEMORIA RAM	HPE de rango dual x4 DDR4-2933 de 32 GB (1 x 32 GB) CAS-21-21-21
	FUENTE	800 W
	Adaptador ETHERNET	1 GB

1.3 Justificación de requerimientos

En procura de brindar un aporte a este punto vital que es la toma de decisiones, se plantea desarrollar un prototipo de Data warehouse que demuestre los beneficios que se obtienen al contar con la información de primera mano en el momento que el usuario lo requiera y sin la intervención de terceros, con características gráficas y estadísticas ahorrando tiempo para facilitar el análisis de información permitiendo así tomar decisiones oportunas. (Valencia Arcos & Guevara Lenis, 2007)

En toda organización o empresa, se toman de decisiones diariamente, desde aplicar descuentos, o comenzar a producir o no una pieza, lanzar una campaña por (correo, redes sociales, mensajes de texto etc.), ordenar o no materiales adicionales, etc. En Algunas

ocasiones, estas decisiones están basadas en hechos, pero con frecuencia están basadas en la experiencia, el conocimiento acumulado o las reglas básicas del negocio. Un ejecutivo de cuentas, de camino a una cita con un cliente, revisa antiguas propuestas y el historial de órdenes, pagos, entregas a su cliente. Con tan solo realizar una pequeña revisión, observa que últimamente ha disminuido el volumen de órdenes de su cliente, a fin de transformar los datos en información útil, para incrementar la rentabilidad de la compañía y tener éxito ayudando a que se cumplan con tareas y objetivos específicos. (KEMP DE LA HOZ, 2005)

Porque es importante para la empresa

En numerosas ocasiones se plantean las empresas que se necesita para continuar con su día a día de forma adecuada. Es importante primero que sepamos, aunque sea a grandes rasgos, qué es un Data warehouse (almacén de datos). Utilizamos este término para definir los sistemas que se usan en las empresas con la intención de llevar a cabo un análisis de datos y un trabajo de reportes. Gracias a estos sistemas todos los datos se integran de forma combinada desde distintas fuentes y se concentran en una localización aquella información que podemos necesitar. Una gran cantidad de empresas, sobre todo las que han visto cómo sus datos se multiplicaban en los últimos años, han encontrado en este tipo de sistema la solución a sus problemas y en relación a eso se mostrarán los motivos por los que se podría necesitar en los negocios.(Juan Grabiél Arqueros Lavado., 2014)

- **Demasiadas hojas de calculo**

Eliminación de este tipo de procesos

- **Tiempos de espera**

El tiempo de acceso a se va a reducir y veremos cómo podremos mejorar mucho los resultados.

➤ **Datos erróneos**

Análisis y cifras en los que no haya ningún tipo de error dado que se trabajaría sobre archivos únicos.

CAPITULO 2

DESARROLLO DEL PROTOTIPO

2.1 Definición del prototipo tecnológico

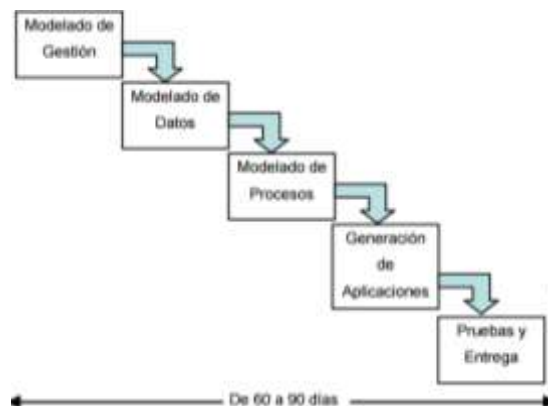
El sistema para la toma de decisiones nivel administrativa en la distribuidora Agrolive, es un software que proporcionara información clave para mejorar la calidad de toma de decisiones analizando datos actuales e históricos.

2.2 Fundamentación teórica del prototipo

Las referencias teóricas incluidas para el desarrollo de la aplicación, se detallan los, programas que serán utilizados para comprender el funcionamiento de todo el sistema.

2.2.1 Metodología de desarrollo del prototipo

La metodología para desarrollar la interfaz de prototipo fue diseñada por la metodología de desarrollo RAD (Rapid Application Development). Una de las más usadas en la plataforma C# de Visual Studio por sus ciclos de desarrollo más pequeños, interfaces graficas estándar y tiempo de prueba y entrega son más rápidos.(Vardaro et al., 2016)



(Cabrera Torres, Moran Cabrera, & Rodriguez Anormaliza, 2015)

Figura # 1 MODELO RAD

2.2.2 Herramientas utilizadas para la construcción de la aplicación C#

Vinculado a la plataforma .NET lenguaje de propósito general orientado a objetos creado por Microsoft. Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando estos últimos años con el objetivo de mejorar tanto su sistema operativo como su modelo de componentes (COM) para obtener una plataforma con la que sea sencillo el desarrollo de software en forma de servicios web.(Saib, 2012)

Los servicios web son un novedoso tipo de componentes software que se caracterizan a la hora de trabajar por su total independencia respecto a su ubicación física real, la plataforma sobre la que corre, el lenguaje de programación con el que hayan sido desarrollados o el modelo de componentes utilizado para ello.(Saib, 2012)

El acceso a estos servicios se realiza en base a estándares de Internet, como son diferentes mecanismos del protocolo HTTP (GET y PUT) o el novedoso protocolo RPC conocido como SOAP (Simple Access Object Protocol), que no es más que una combinación de estándares como HTTP y XML para realizar llamadas a los miembros de estos servicios web. La idea detrás de SOAP consiste sencillamente en utilizar HTTP como medio de transporte para el envío de los mensajes de solicitud

de ejecución de los miembros de servicios web remotos (lo que permite atravesar barreras tales como firewalls) y utilizar XML como lenguaje para escribir los cuerpos de estos mensajes.(Saib, 2012)

Pero la plataforma .NET no son sólo los servicios web, sino que también ofrece numerosos servicios a las aplicaciones que para ella se escriban, la total integración entre lenguajes (por ejemplo, es posible escribir una clase en C# que derive de otra escrita en Visual Basic.NET que a su vez derive de otra escrita en Cobol). (Vardaro et al., 2016)

Es posible programar la plataforma .NET en prácticamente cualquier lenguaje. .NET, un lenguaje que no cuente con elementos heredados de versiones anteriores e innecesarios en esta plataforma y que por tanto sea lo más sencillo posible para programarla aprovechando toda su potencia y versatilidad.

C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi. De hecho, su creador Anders Heljsberg fue también el creador de muchos otros lenguajes y entornos como:

Turbo Pascal, Delphi o Visual J++. La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic, y que además la migración a este lenguaje por los programadores de C/C++/Java sea lo más inmediata posible. Además de C#, Microsoft proporciona Visual Studio.NET, la nueva versión de su entorno de desarrollo adaptada a la

plataforma .NET y que ofrece una interfaz común para trabajar de manera cómoda y visual con cualquiera de los lenguajes de la plataforma .NET (por defecto, C++, C#, Visual Basic.NET y JScript.NET, aunque pueden añadirse nuevos lenguajes mediante los plugins que proporcionen sus fabricantes).(Saib, 2012)

SQL Server

Microsoft SQL Server es un sistema de gestión de base de datos relacional desarrollado por Microsoft. Como un servidor de base de datos, es un producto de software con la función principal de almacenar y recuperar datos según lo solicitado por otras aplicaciones de software que pueden funcionar ya sea en el mismo ordenador o en otro ordenador a través de una red (incluyendo el Internet).

Existen al menos una docena de ediciones diferentes de Microsoft SQL Server, dirigidos a diferentes públicos y para cargas de trabajo que van desde pequeñas aplicaciones de una sola máquina hasta grandes aplicaciones orientados a Internet con muchos usuarios concurrentes. (Sql, Transact, & Básico, 2019)

Historia de SQL Server

En 1988 Microsoft se unió a Ashton-Tate y Sybase para crear una variante de Sybase SQL Server para IBM OS / 2 (entonces desarrollado conjuntamente con Microsoft), que fue lanzado el año siguiente. Esta fue la primera versión de Microsoft SQL Server, y sirvió como la entrada de Microsoft en el mercado de bases de datos a nivel de empresa, compitiendo contra Oracle, IBM, y más tarde, Sybase. SQL Server 4.2 fue enviado en 1992, incluido con OS / 2 versión 1.3, seguido de la versión 4.21 para Windows NT, lanzado junto con Windows NT 3.1.

SQL Server 6.0 fue la primera versión diseñada para NT, y no incluyó ninguna dirección de Sybase. (Sql et al., 2019)

A partir de ahí van apareciendo diversas versiones, siendo los lanzamientos más importantes:

- SQL Server 2000
- SQL Server 2005
- SQL Server 2008
- SQL Server 2012
- SQL Server 2014
- SQL Server 2016

Para cada una de las versiones, Microsoft ofrece también diversas Ediciones, con diferentes conjuntos de características y dirigidos a diferentes usuarios. Estas ediciones son:

- Data Center
- Enterprise
- Standard
- Web
- Business Intelligence
- Workgroups

Y la más importante ya que va dirigida al gran público.

Express: SQL Server Express Edition es una versión totalmente gratuita, reducida, de libre distribución, que incluye el motor de base de datos central. Si bien no hay limitaciones en el número de bases de datos o usuarios soportados, se limita a la utilización de un procesador, 1 GB de memoria y 10 archivos de datos (base de datos de 4 GB de archivos anteriores a SQL Server Express 2008 R2).

Microsoft SQL Analysis Services

Analysis Services es un motor de datos analíticos (Vertipaq) utilizado en el soporte de decisiones y análisis de negocios. Proporciona modelos de datos semánticos de nivel empresarial para informes comerciales y aplicaciones de clientes como Power BI, Excel, informes de Reporting Services y otras herramientas de visualización de datos. (Sql et al., 2019)

Instalado como una instancia de servidor local, SQL Server Analysis Services admite modelos tabulares en todos los niveles de compatibilidad (según la versión), modelos multidimensionales, minería de datos y Power Pivot para SharePoint.

Ofrece los datos analíticos para:

- ✓ Informes empresariales
- ✓ Aplicaciones cliente como Excel
- ✓ Informes de Reporting Services
- ✓ Otras herramientas de BI

Con Microsoft SQL Analysis Services es posible crear cubos de análisis multidimensionales que permiten analizar cada área de la compañía de acuerdo con diferentes factores. Por ejemplo, el análisis de las ventas basado en criterios como:

- ✓ Países
- ✓ Grupos de productos
- ✓ Equipo comercial
- ✓ Marcas
- ✓ Márgenes
- ✓ Unidades de negocio Etc.

Los datos procedentes de diferentes fuentes de su compañía pueden ser transformados y modelados, preparándolos para su visualización y análisis en herramientas sencillas como Excel, Power BI, Reporting services u otras aplicaciones. (Eguila, 2007)

Microsoft SQL Server Analysis Services ofrecen potentes funcionalidades para el Data mining y el procesamiento analítico de datos en línea, dentro de un entorno de fácil uso y rico en funcionalidades. (Bernabeu, 2010)

A los fines de este proyecto, con el objeto de generar información objetiva, oportuna, de calidad y confiable y ponerla a disposición de la gerencia y que los datos obtenidos sean útiles para el proceso de toma de decisiones, el uso de un Data warehouse es propicio para el fin último de los datos. (Eguila, 2007)

A través del Data warehouse es posible:

Visualizar información en diferentes periodos de tiempo.

- Hacer filtros dinámicos.
- Manejar escenarios (filtros por dimensión, etc.) y moverse entre ellos, así como imprimir los o exportarlos en diferentes formatos JPEG, PNG, XLS, HTML, PDF.

- Crear consultas personalizadas, mediante la agregación de columnas calculadas, cortes por dimensión, etc.
- Definir operaciones en una columna de indicador.
- Crear plantillas de análisis (tendencias, máximos y mínimos, tendencia incremental, etc.).
- Realizar proyecciones.

¿Qué es Data warehouse?

Un Data warehouse o almacén de datos es el centro de la arquitectura para los sistemas de información desde la década de los '90. Soporta el procesamiento informático al proveer una plataforma sólida y robusta, a partir de los datos históricos para realizar análisis. Facilita la integración de sistemas de aplicación no integrados. Organiza y almacena los datos que se necesitan para el procesamiento analítico-informático sobre una amplia perspectiva de tiempo. Un Data warehouse definido es una colección de datos orientados, integrado, no volátil, de tiempo variante, usado principalmente para el soporte en el proceso de toma de decisiones gerenciales. Cabrera Torres, Moran Cabrera, & Rodríguez Anormaliza, 2015).

Un Data warehouse se puede caracterizar haciendo un contraste de cómo los datos de un establecimiento son almacenados en un Data warehouse, difieren de los datos operacionales usados por las aplicaciones de producción.

La primera fase consiste en la selección de la fuente de los datos, seguida por el análisis y adecuación de los metadatos al dominio de aplicación específico del proyecto. Finalizadas las fases iniciales, da lugar al comienzo del proyecto Data

warehouse, que consiste en la selección de los procesos de negocio, definición de la granularidad de los datos, el modelado de las dimensiones y la identificación de la tabla de hechos, concluido esto, inicia la fase de implementación y la carga de los datos en el Data warehouse desde los sensores ubicados en las centrales meteorológicas u otras fuentes de datos, como los registros históricos. (Aquiicultura et al., 2018)

2.3 METODOLOGIAS DE DESARROLLO DE UN DATA WAREHOUSE

Para la obtención de resultados esperados en un Data warehouse (almacén de datos), los procesos de negocio se seleccionan con el objetivo de modelarlos, para así establecer una granularidad de cada uno de ellos. Por tal motivo es necesario comprender correctamente los datos de los diferentes sistemas dentro de un negocio y sus respectivas relaciones. La gestión de estas relaciones durante la carga de almacenamiento de datos es esencial. Es por esto que para el desarrollo de un Data Warehouse no existe una única metodología en la que se basará el diseño, dependiendo exclusivamente del contexto en el que se encuentra la empresa y los objetivos que persiga se definirá el alcance.(Bernabeu, 2010)

Estas diferentes metodologías se pueden englobar dentro de dos grandes bloques:

- Top-down
- Bottom-up

Modelo top-down

Modelo propuesto por Bill Inmon, se utiliza cuando la tecnología y los problemas o carencias del establecimiento se conocen con anterioridad. Este enfoque logra la sinergia

entre los problemas de negocio alcanzando objetivos perseguidos. Se trata de un método sistémico, que minimiza los problemas de integración, pero costoso, debido a la gran cantidad de datos y su poca flexibilidad. Además, en este método se formula un resumen del sistema, sin especificar detalles. Cada parte del sistema se mejora diseñándola con mayor detalle. Después, cada parte nueva se vuelve a mejorar, cada vez con mayor detalle, hasta que la especificación completa es lo suficientemente detallada para la validación del modelo. Este modelo se diseña con frecuencia con la ayuda de pruebas de "caja negra" que hacen más fácil cumplir requerimientos, aunque estas cajas negras no expliquen en detalle los componentes individuales. (Duque Méndez & Tamayo Alzate, 2001)

El enfoque top-down considera que el almacén de datos debe responder a las necesidades de cada uno de los usuarios del establecimiento, y no sólo de un determinado grupo o departamento.

Metodologías contenidas dentro del modelo Top-down

Metodología propuesta por (Hefesto)

La metodología HEFESTO, permite la construcción de Data Warehouse de forma sencilla, ordenada e intuitiva.

HEFESTO es una metodología, cuya propuesta esta fundamentada en una amplia investigación, comparación de metodologías existentes y experiencias propias en procesos de confección de almacenes de datos.

La idea principal, es comprender cada paso que se realizará, para no caer en el tedio de tener que seguir un método al pie de la letra sin saber literalmente qué se está haciendo. (Bernabeu, 2010)

La construcción e implementación de un Data warehouse puede adaptarse a cualquier ciclo de vida de desarrollo de software, con la limitación de que, para algunas fases en particular las acciones que se han de realizar serán muy diferentes. Lo que se debe tener muy en cuenta, es no entrar en la utilización de metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que conlleve demasiado tiempo y fases muy largas. Lo que se busca, es entregar una primera implementación que satisfaga una parte de las necesidades, para demostrar las ventajas del Data warehouse. (Castelán & Ocharán, 2012)

La metodología HEFESTO, puede ser integrado en cualquier ciclo de vida que cumpla con la condición antes declarada.

Para que se llegue a una total comprensión de cada uno de los pasos o etapas, se describe continuación la metodología con el fin ejemplificar cada concepto.



Figura # 2 Metodología (Bernabeu, 2010)

Como se puede apreciar en la figura, se comienza recolectando las necesidades de información de los usuarios y se obtienen las preguntas claves del negocio. Luego, se deben identificar los indicadores resultantes de los interrogantes y sus respectivas dimensiones de análisis, mediante las cuales se construirá el modelo conceptual de datos del Data warehouse.

Después, se analizarán los OLTP (Proceso de transacciones en línea) para determinar cómo se construirán los indicadores como por ejemplo el total de las unidades vendidas o de ganancias de uno o varios productos, señalar las correspondencias con los datos fuentes y para seleccionar los campos de estudio de cada perspectiva. (Eguila, 2007)

Una vez hecho esto, se pasará a la construcción del modelo lógico del almacén, en donde se definirá cuál será el tipo de esquema que se implementará. Seguidamente, se confeccionarán las tablas de dimensiones que son elementos que contienen atributos (campos) que se utilizan para restringir y agrupar los datos almacenados en una tabla de hechos (proceso principal de entidad) cuando se realizan consultas sobre dichos datos en un entorno Data warehouse o Datamart. Por último, se definirá la tabla de hechos la cual es la tabla principal de un esquema dimensional

(en estrella o en copo de nieve) que contiene los valores de las medidas de negocio, para luego efectuar sus respectivas uniones.

Finalmente, se definirán los procesos de extracción, transformación y carga (ETL) de los datos fuente, que poblarán y actualizarán el almacén de datos. (Eguila, 2007)

Características

Las características que dispone esta metodología son las siguientes:

- Los objetivos y resultados deseados en cada fase se distinguen fácilmente y son sencillos de entender.
- Se funda en los requerimientos del usuario, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el establecimiento.
- Disminuye la resistencia al cambio, ya que involucra al usuario final en cada etapa para que tome decisiones respecto al comportamiento y funciones del Data warehouse.
- Utiliza modelos conceptuales y lógicos, sencillos de analizar e interpretar.
- Es independiente del tipo de ciclo de vida que se apliquen para contener la metodología.

- Es autónomo de las herramientas que se usen para su implementación.
- Es autónomo de las estructuras físicas que contengan el Data warehouse y de su respectiva distribución.
- Cuando finaliza una fase, los resultados obtenidos se convierten en el punto de inicio para llevar a cabo el paso siguiente.
- Aplicado tanto para Data warehouse como para Data mart.

Metodología propuesta por (Bill Inmon)

Esta metodología propone los mecanismos necesarios para llevar a cabo la correcta realización de un Data warehouse. Para Bill Inmon, el diseño de un Data warehouse comienza con los procesos que definirán el almacenamiento de los datos en el sistema, debido a que se almacenarán grandes volúmenes de datos, por lo que depende de este proceso la eficiencia en el acceso a los datos. (Adriana Patricia Muñoz Zapata, 2011)

Además, la definición de Inmon sostiene uno de los principios fundamentales del desarrollo de un Data warehouse, principio en el que el ambiente de origen de los datos y el ambiente de acceso de los datos, deben estar físicamente en diferentes DB (bases de datos) y en equipos (hardware) separados.

Por último, los actuales sistemas tienen gran cantidad de datos, lo que hace poco realista el intentar hacer cargas cada poco tiempo. Si el volumen de datos no es cuidadosamente gestionado y condensado, dicho volumen de datos impide que los objetivos del Data warehouse se alcancen. (KEMP DE LA HOZ, 2005)

A la metodología Inmon frecuentemente se le asocia con los Data warehouse a nivel empresarial, que involucran desde un inicio todo el ámbito corporativo, sin centrarse en un incremento específico hasta después de haber culminado completamente el diseño del Data warehouse. En su doctrina, un Datamart es sólo una de las capas del Data warehouse y los Datamart son dependientes del depósito central de datos o Data warehouse Corporativo y, por lo tanto, se construyen después de él. (Kimball and Inmon, 2012)

Inmon considera que, es necesario desarrollar una estrategia de Data warehouse e identificar las principales áreas desde el inicio del proyecto, conociendo de esta forma con antelación y extrema exactitud la estructura que presentarán los principales núcleos de desarrollo en el negocio, para asegurar una solución integral, evitando la aparición de situaciones inesperadas que puedan poner en peligro el proyecto.

Inmon es defensor de utilizar el modelo relacional para el ambiente en el que se implementará el Data warehouse corporativo, ya que como él mismo asegura, la creación de una base de datos relacional con una leve normalización, son la base de los Data mart. (Kimball and Inmon, 2012)

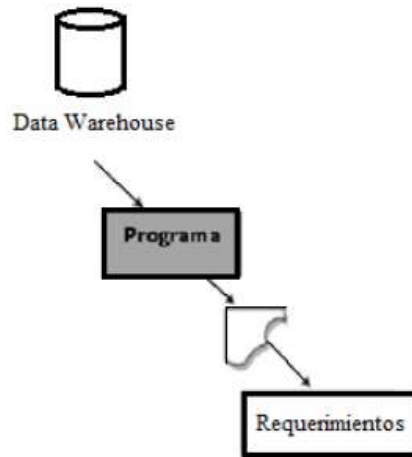


Figura #3: Metodología propuesta por Bill Inmon

La metodología de Inmon tiene un enfoque a modo de explosión en el sentido de que en cierto modo no viene acompañada del ciclo de vida normal de las aplicaciones, sino que los requisitos irán acompañando al proyecto según vaya comprobándose su necesidad. Esta visión de Inmon puede traer consigo mucho riesgo a la empresa, ya que invierte grandes esfuerzos en el desarrollo del Data warehouse y no es hasta la aparición de los Datamart cuando se inicia a explotar la inversión y obtener beneficios. Esta estrategia se contempla en el marco de que es imposible conocer cuáles son las necesidades concretas de información de una compañía, el ambiente dinámico en que se mueve la organización, el cambio de estructura que conlleva el desarrollo de la nueva plataforma y los consiguientes cambios al software transaccional que implica su introducción. Con esto posiblemente después de la gran inversión en tiempo y recursos en el desarrollo del

Data warehouse, se haga evidente la necesidad de cambios esenciales que traen consigo altos costos de desarrollo para la organización, poniendo en evidente peligro el éxito de todo el proyecto en sí y que podían ser evitados con una rápida detección en una temprana puesta en explotación de un primer avance del Data warehouse. (Kimball and Inmon, 2012)

Esta metodología para la construcción de un sistema de este tipo, es frecuente a la hora de diseñar un sistema de información, utilizando las herramientas habituales como el esquema Entidad/Relación, pero al tener un enfoque global, es más difícil de desarrollar en un proyecto sencillo, pues se está intentando abordar “todo”, a partir del cual luego se irá al “detalle”. Esta es otra de las delimitaciones que trabajan en contra de la metodología de Inmon ya que implica un mayor tiempo de consumo, teniendo como consecuencia que muchas empresas se inclinen por usar metodologías con las que obtengan resultados palpables en un espacio menor de tiempo. (Kimball and Inmon, 2012).

En resumen, la metodología de Inmon es distinta a las otras, ya que en estas metodologías convencionales generalmente se habla de conformar un Data Warehouse a partir de distintas fuentes, generando de esta forma los Datamarts, que, en este caso, estarían todos en un mismo repositorio.(Duque Méndez & Tamayo Alzate, 2001)

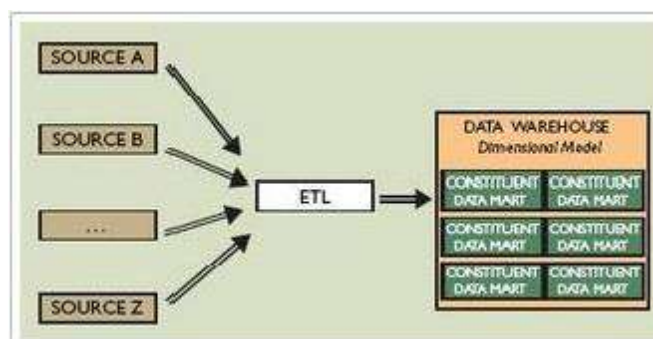


Figura # 4: Ejemplo de metodología convencionales (Kimball and Inmon, 2012)

A diferencia de la anterior, en esta metodología propuesta por Inmon, el Data warehouse no está modelado dimensionalmente, sino que está en tercera forma normal (3NF). Así, el creador de este modelo entiende que esta forma es mucho más rica y adaptable que el modelo anterior. Una vez que se tiene el Data Warehouse generado, se pueden crear los Datamarts para las áreas de negocio que se necesiten, y además se puede utilizar para cualquier otro tipo de sistema decisional como por ejemplo sistemas expertos, o minería de datos. A continuación, la Figura 3, ilustra el modelo propuesto por Inmon.

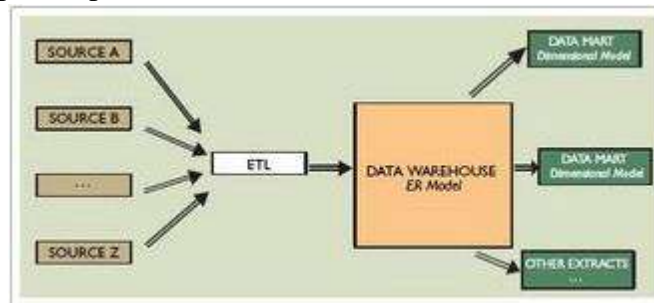


Figura # 5: Ejemplo de metodologías Inmon

Modelo Bottom up

Es un modelo que se caracteriza por ser rápido ya que se basa en experimentos y prototipos propuesta por Ralph Kimball. Siendo este un método flexible que permite a la organización

ir más lejos con menores costos. La idea es construir Datamart independientes para evaluar las ventajas del nuevo sistema a medida que avanzamos. En él, las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo. (Kimball and Inmon, 2012)

Las estrategias basadas en el flujo de información Bottom-up pueden ser potencialmente necesarias y suficientes, porque se basan en el conocimiento de todas las variables que pueden afectar a los elementos del sistema.(Kimball and Inmon, 2012)

Metodologías incluidas dentro del modelo Bottom-up

Rapid Warehousing Methodology (RWM)

Dicha metodología es iterativa y está basada en el desarrollo incremental del proyecto de Data Warehouse dividido en seis fases como se aprecia en la figura.

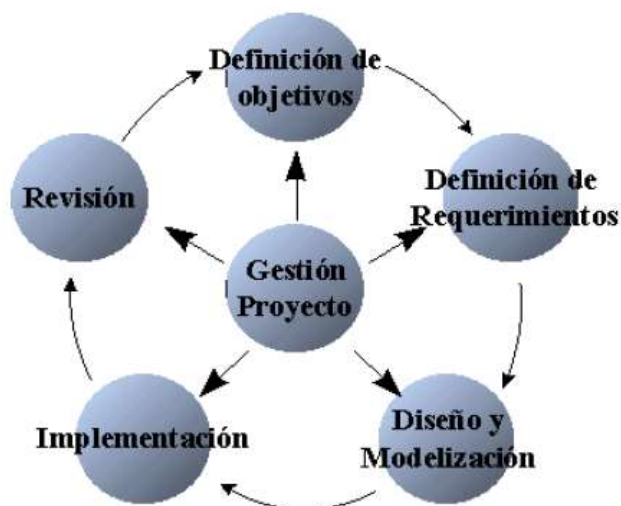


Figura #6 Rapid Warehousing Methodology (Bruno, 2019)

Descripción de fases de Rapid Warehousing Methodology

✓ Definición de los objetivos

En esta fase se definirá el equipo de proyecto que debe estar compuesto por representantes del departamento informático y de los departamentos usuarios del Data warehouse, además de la figura de jefe de proyecto. Se definirá el alcance del sistema y cuáles son las funciones que el Data warehouse realizará como suministrador de información de negocio estratégica para la empresa. Se definirán así mismo, los parámetros que permitan evaluar el éxito del proyecto. (Bruno, 2019)

✓ Definición de los requerimientos de información

Durante esta fase se mantendrán sucesivas entrevistas entre los representantes del departamento usuario final y los representantes del departamento de informática. Se realizará el estudio de los sistemas de información existentes, que ayudarán a comprender las carencias actuales y futuras que deben ser resueltas en el diseño del Data warehouse. Asimismo, en esta fase el equipo de proyecto debe ser capaz de validar el proceso de entrevistas y reforzar la orientación de negocio del proyecto. Al finalizar esta fase se obtendrá el documento de definición de requerimientos en el que

se reflejarán no sólo las necesidades de información de los usuarios, sino cuál será la estrategia y arquitectura de implantación del Data warehouse.(Vardaro et al., 2016)

✓ **Diseño y modelización**

Los requerimientos de información identificados durante la anterior fase proporcionarán las bases para realizar el diseño y la modelización del Data Warehouse. En esta fase se identificarán las fuentes de los datos (sistema operacional, fuentes externas, etc.) y las transformaciones necesarias para, a partir de dichas fuentes, obtener el modelo lógico de datos del Data Warehouse. Este modelo estará formado por entidades y relaciones que permitirán resolver las necesidades de negocio de la organización.(Eguila, 2007)

✓ **Implementación**

La implantación de un Data Warehouse lleva implícitos los siguientes pasos:

- ✓ Extracción de los datos del sistema operacional y transformación de los mismos.
- ✓ Carga de los datos validados en el Data Warehouse. Esta carga deberá ser planificada con una periodicidad que se adaptará a las necesidades de refresco detectadas durante la fase de diseño del nuevo sistema.
- ✓ Explotación del Data Warehouse mediante diversas técnicas dependiendo del tipo de aplicación que se dé a los datos. Entre las técnicas más habituales podemos encontrar las siguientes:

- Query & Reporting
- On-line analytical processing (OLAP)
- Executive Information System (EIS) o Información de gestión
- Decision Support Systems (DSS)
- Visualización de la información

✓ **Revisión**

La construcción del Data Warehouse no finaliza con la implantación del mismo, sino que es una tarea iterativa en la que se trata de incrementar su alcance aprendiendo de las experiencias anteriores. Después de implantarse, se debe realizar una revisión del Data warehouse planteando preguntas que permitan, después de los seis o nueve meses posteriores a su puesta en marcha, definir cuáles serían los aspectos a mejorar o potenciar en función de la utilización que se haga de la nueva aplicación. (Aqüicultura et al., 2018)

✓ **Gestión del Proyecto**

La gestión del proyecto debe encargarse de la coordinación y ejecución de las distintas fases que conforman la construcción e implantación de un Data warehouse. Este proceso se tiene que apoyar en una metodología específica para este tipo de trabajos, si bien es más importante que la elección de la mejor de las metodologías, el realizar un control para asegurar el seguimiento de la misma. En las fases que se establezcan es fundamental incluir una fase de formación en la herramienta utilizada, para un máximo

aprovechamiento de la aplicación. Seguir los pasos de la metodología y comenzar el Data warehouse por un área específica de la organización permitirá la obtención resultados tangibles en un corto espacio de tiempo. (Bruno, 2019)

✓ Metodología Kimball – Ciclo de Vida

La siguiente figura muestra el esquema de las fases que componen la metodología propuesta por Kimball y las siguientes secciones resumen el contenido de cada una de las fases.(Kimball and Inmon, 2012)

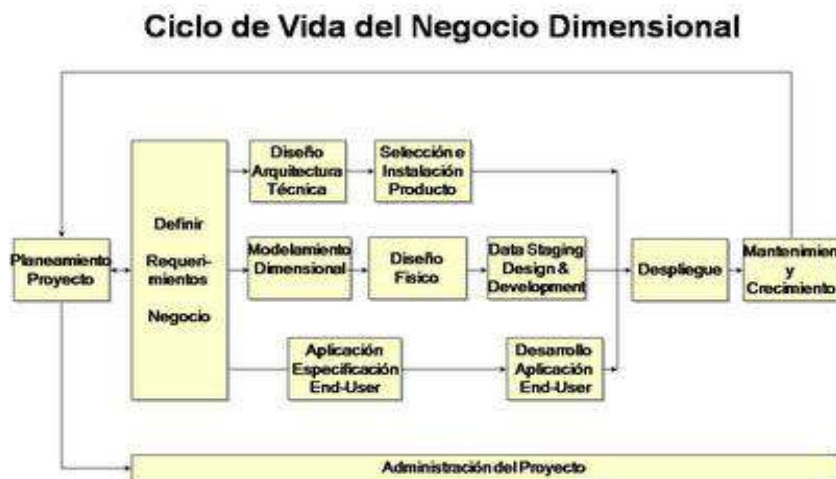


Figura #7 Metodología Kimball - Ciclo de vida(Kimball and Inmon, 2012)

La arquitectura de un Data warehouse puede tener diferentes estructuras en diferentes implementaciones. Algunas pueden tener un ODS (operational data store), mientras que otras pueden tener múltiples Data marts. Algunas pueden tener un pequeño número de fuentes de datos, mientras que otras pueden tener docenas de fuentes de datos, es mucho más razonable de presentar las diferentes capas de la arquitectura de un Data warehouse. (AGUILAR, n.d.2014)

En general, toda arquitectura de un Data warehouse tiene las siguientes capas:

- Capa fuente de datos
- Capa de extracción de datos
- Área organizacional
- Capa ETL (Extracción, transformación y lectura)
- Capa de almacenamiento de datos
- Capa logia de datos
- Capa de metadatos
- Capa de operaciones del sistema

Descripción de capas

- ✓ **Capa de fuentes de datos:** Esta representa las diferentes fuentes de datos que alimentan los datos del data warehouse. La fuente de datos puede estar en cualquier formato: archivo de texto plano, base de datos relacional, otros tipos de base de datos, archivo Excel, etc. Todos estos pueden actuar como fuente de datos. Además, los tipos de datos pueden ser muy variados:
 - Datos de operaciones, como datos de ventas, datos de recursos humanos, datos de productos, datos de inventario, datos de marketing y datos de sistemas.
 - Logs (registros) de un servidor web, con datos de navegación de los usuarios.
 - Datos internos de investigación de mercado.
 - Datos de terceros, como datos del censo, datos demográficos o datos de encuestas.

- ✓ **Capa de extracción de datos:** Los datos se extraen de las fuentes de datos y se llevan al sistema Data warehouse. Es probable que en esta capa se limpien algunos datos mínimos, pero no es previsible que haya una transformación de datos importante.
- ✓ **Área de pruebas:** aquí es donde los datos son depurados y transformados en un Datamart y Data warehouse. Tener un área común facilita el proceso y la integración posterior de los datos.
- ✓ **Capa ETL (extracción, transformación y lectura):** Es aquí donde los datos obtienen su inteligencia ya que se aplica la lógica para transformar los datos de carácter transaccional a una naturaleza analítica. En esta capa es también donde se limpian los datos. La fase de diseño ETL es frecuentemente la fase que más se demora en un proyecto de Data warehouse y habitualmente se utiliza una herramienta ETL en esta capa.
- ✓ **Capa de almacenamiento de datos:** Aquí es dónde se colocan los datos transformados y limpios. Basándose en el alcance y la funcionalidad se pueden encontrar tres tipos de entidades: Data warehouse, data mart y almacén de datos operacional (ODS). En cualquier sistema puedes encontrar sólo uno de los 3, 2 de los 3, o los tres tipos juntos.
- ✓ **Capa lógica de datos:** Aquí es donde se almacenan las reglas de negocio. Estas reglas de negocio no afectan a las reglas de transformación de datos, pero afectan a lo que luego puedes ver en los informes.
- ✓ **Capa de presentación de datos:** Se refiere a la información que llega a los usuarios. Esto puede ser en forma de un informe tabular o gráfico a través de un navegador, un informe enviado por email que se genera automáticamente

y se envía a diario, una alerta que advierte a los usuarios acerca de excepciones, etc. Usualmente en esta capa se utiliza una herramienta OLAP y una herramienta de generación de informes.

- ✓ **Capa de metadatos:** Aquí es donde la información sobre los datos almacenados en el Data warehouse es almacenada. Un modelo de datos lógico sería un ejemplo de algo que está en esta capa de metadatos. Frecuentemente se utiliza una herramienta de metadatos para administrar los metadatos.
- ✓ **Capa de operaciones del sistema:** Esta capa incluye información sobre cómo está funcionando el sistema de Data warehouse, cuál es el estado de trabajo ETL, cuál es el rendimiento del sistema y el historial de acceso de los usuarios.

Propiedades de la arquitectura de un Data warehouse

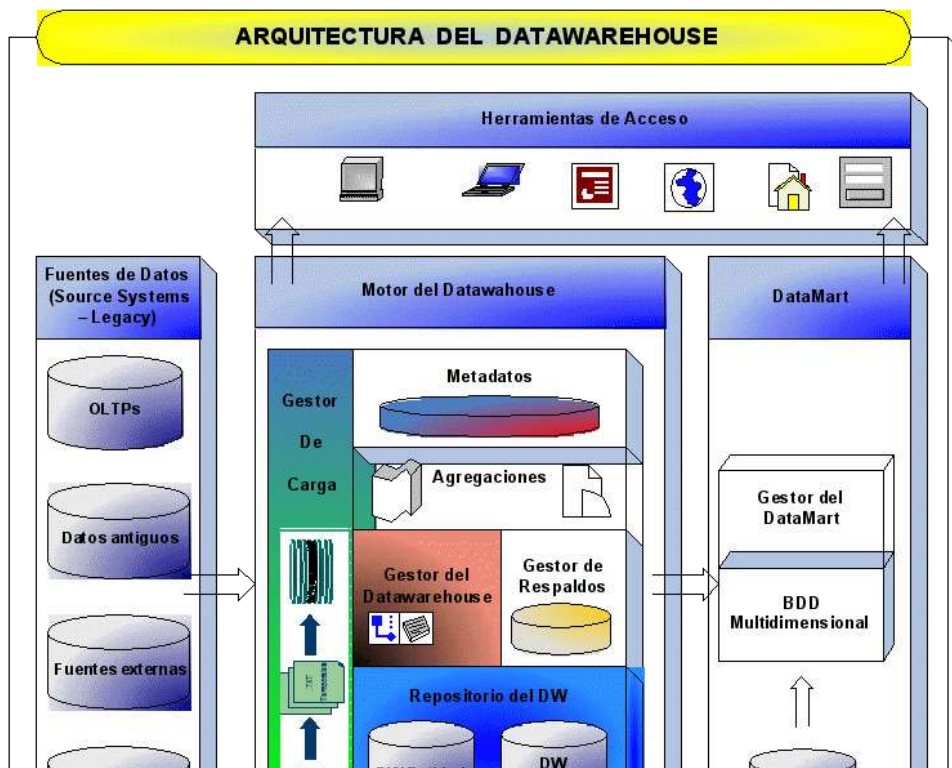
Las siguientes propiedades son esenciales para la arquitectura de un Data warehouse:

Separación: el proceso analítico y transaccional deben mantenerse separados tanto como sea posible.

- **Escalabilidad:** la arquitectura de un Data warehouse, tanto de hardware como de software, debe ser fácil de actualizar a medida que crece el volumen de datos que

debe ser gestionado y procesado, así como el número de requisitos de los usuarios que tienen que ser satisfechos.

- **Extensibilidad:** la arquitectura debería ser capaz de alojar nuevas aplicaciones y tecnologías sin necesidad de revisar todo el sistema.
- **Seguridad:** monitorizar los accesos es esencial debido a los datos estratégicos que hay almacenados en el Data warehouse.
- **Administrabilidad:** la gestión del Data warehouse no debería ser excesivamente difícil.



figura# 8 arquitectura Data warehouse (Flores & Martínez Chirichian, 2017)

OLTP Y OLAP

Para el proceso ETL son las bases para la obtención de información definidas como una tabla de hechos y tablas de dimensiones. Consideradas tablas de hechos a la principal de ventas y las dimensiones son las existen para formarlas.

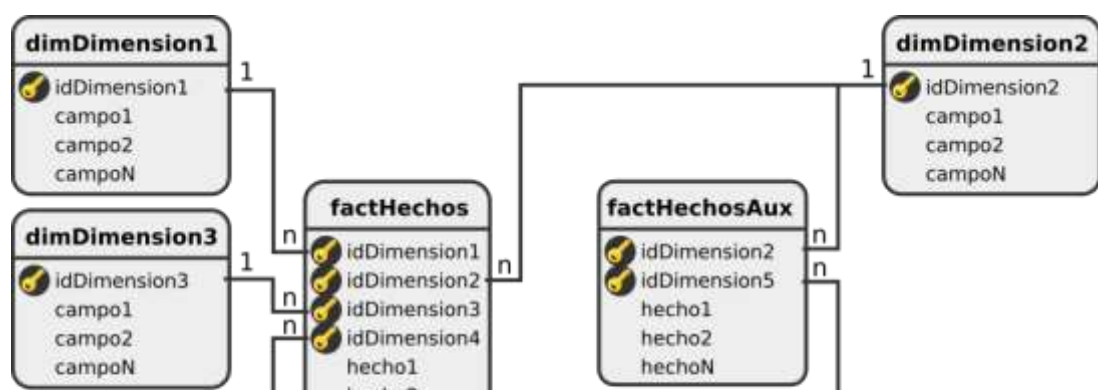


Imagen # 9 [tablas de hechos y dimensiones]

Procesamiento de transiciones en línea (OLTP)

Es una clase de sistemas que facilitan y administran aplicaciones orientadas a transacciones, lo general para la entrada de datos. Los Data warehouse son compatibles con el sistema OLTP, proporcionando no solo la descarga de datos OLTP a medida que se acumulan, sino también los servicios de que otro modo se degradarían si se llevaran a cabo en la base de datos OLTP. (Juan Grabiél Arqueros Lavado., 2014)

Procesamiento analítico en línea (OLAP)

Es una estructura de datos utilizada para analizar la información de la base de datos de los sistemas de datos múltiples a la vez. Data warehouse y OLAP son utilizados indistintamente aplicados los diferentes de sistema de Business Intelligence (BI). Los componentes de estos sistemas incluyen base de datos y aplicaciones que proporcionan las herramientas analíticas que sirven para apoyar en la toma de decisiones en las organizaciones. La tecnología OLAP permite que los Data warehouse sean efectivos a la hora de hacer análisis en línea, proporcionando

rápidas repuestas a consultas analíticas complejas.(Juan Grabiél Arqueros Lavado., 2014)

Diferencias de los sistemas OLTP y OLAP

Propósito:

- OLTP es un Data Warehouse que provee una aplicación relacionada a transacciones.
- Los sistemas OLAP son diseñados para el análisis y la toma de decisiones.

Fuentes de Datos:

- En los sistemas OLTP, muchos usuarios pueden insertar datos simultáneamente. Las actualizaciones en línea están constantemente en movimiento manteniendo registros de operaciones.
- En OLAP, la mayoría de las Fuentes de datos tiene una planificación para su actualización. Los datos de OLAP provienen de varias bases de datos OLTP.

Índices:

- Los índices OLTP pueden ser lentos cada vez que se actualicen tablas e índices. Por esta razón, OLTP tiene menos índices por tablas que pueden mostrar las consultas más comunes asociados con los informes de flujo de trabajo.

- En contraste, los índices en OLAP causan una mejora tremenda en la velocidad de búsqueda.

Espacio en Disco:

- OLTP no consume mucho espacio en disco, más que Gigabytes.
- En la otra mano tenemos a OLAP que ocupa grandes cantidades de datos, alcanzando los Terabytes.

Tabla de comparación

	OLAP	OLTP
Usuario	Gerente, Ejecutivo	Empleado, Profesional de Sistemas
Función	Soporte a las Decisiones	Operación diaria
Diseño de la Base de Datos	Orientado a una materia	Orientado a una aplicación
Datos	Históricos, resumidos, multidimensionales, integrados, consolidados, opcionalmente detallados	Actuales, detallados, relacionales, aislados
Uso	Ad hoc, requerimientos del momento	Repetitivo
Acceso	Exploración	Lectura/Escritura, Índices/Algoritmos de dispersión
Unidad de Trabajo	Consultas complejas	Transacciones simples

Imagen # 10 diferencias entre OLAP y OLTP. (Flores & Martínez Chirichian, 2017)

Fuente de Datos Disparos (heterogéneos):

Como resultado de este formato de datos pobre, puede haber múltiples apariciones de los mismos datos, lo que conduce a la redundancia de datos. Para superar este problema, el ETL es muy útil, ya que refina los datos heterogéneos mediante la

integración de los datos en un lugar de trabajo antes de cargarlos en el Data warehouse.

Una fuente de datos interna: Es una colección de datos de usuarios individuales que se encuentran dentro de la organización. Los datos internos aumentan la complejidad de la transformación e integración de datos, ya que se obtiene de diversas fuentes, tales como hojas de cálculo y documentos de texto.

Una fuente de datos externa: es una fuente de datos utilizando los datos recogidos de fuentes externas, como las estadísticas actuales de la industria y cuotas de mercado de los competidores. Los datos externos ayudan a una organización a detectar las tendencias actuales de la industria y comparar el rendimiento con respecto a sus competidores. Estos no se ajustan necesariamente a los formatos de la organización, por lo que tiene que ser convertido a formatos apropiados.

Los metadatos son datos que definen el Data warehouse. Incluyen información sobre el contenido del Data warehouse, los procesos que tienen lugar en la parte de atrás, fuentes de datos, el almacenamiento y la seguridad y la autenticación. (Bernabeu, 2010).

Puntos a tener en cuenta:

- Heterogeneidad de las fuentes de datos
- Tendencia a crecer
- Orientada al negocio

El Modelado Dimensional es una técnica de diseño de base de datos donde los datos son representados en 2 tipos de tablas:

- La Tabla de Hechos usada para almacenar hechos y medidas actuales en la empresa.

- La Tabla de Dimensiones almacena campos que describen los hechos.

Un Data mart es un Data Warehouse en miniatura que se personaliza para adaptarse a las exigencias de un departamento. Se trata de un repositorio de datos de una organización para proporcionar las diferentes funciones de la organización con información significativa.

Diseño: Las actividades típicas en esta etapa incluyen entrevista a los usuarios finales, definición de keys(llaves), indicadores de performance, mapeo de los procesos de toma de decisión y diseño de esquemas lógicos y físicos.

Prototipo: en esta etapa, el equipo de diseño crea un prototipo de Data Warehouse, basado en las entradas y salidas. Los usuarios finales sugieren cambios hasta que los requerimientos son completados.

Despacho: El prototipo es aprobado por los usuarios finales y es despachado en dos ambientes distintos: el environment(ambiente) de producción-prueba y el de producción actual.

Operación: Esto involucra al mantenimiento de día a día del Data Warehouse.

Mejora: Las modificaciones y las mejoras son hechas al Data Warehouse por los cambios de negocios en cuanto a los procesos y necesidades.

La presentación de datos describe todos los métodos y sistemas por los cuales se pone disponible la información a los usuarios

Reportes Ad hoc: son informes predefinidos que se adaptan a las necesidades de los usuarios inexpertos. Se producen por el uso de consultas simples en lugar de

consultas complejas. Son esencialmente informes que no están formateados y no están destinados a usuarios profesionales.(Juan Grabiél Arqueros Lavado., 2014)

Los indicadores clave de rendimiento (KPI) son un conjunto de indicadores de desempeño definidos para un determinado proceso en la empresa.

Un Dashboard es una herramienta de Business Intelligence que se utiliza para describir el desempeño de los procesos de una empresa.

2.3 Objetivos del prototipo

2.3.1 Objetivo General

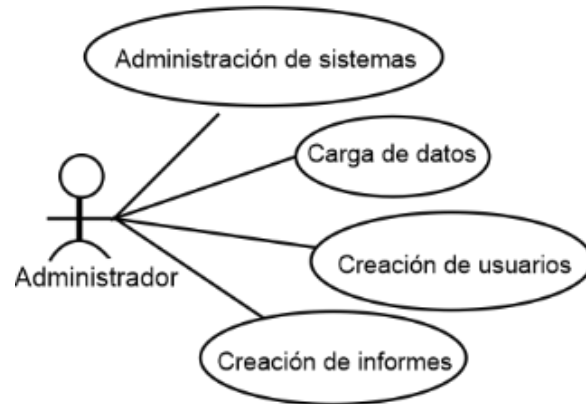
- ✓ Desarrollar un sistema de toma de decisiones nivel administrativo para la distribuidora de insumos agrícolas “Agrolive”.

2.3.1 Objetivos Específicos

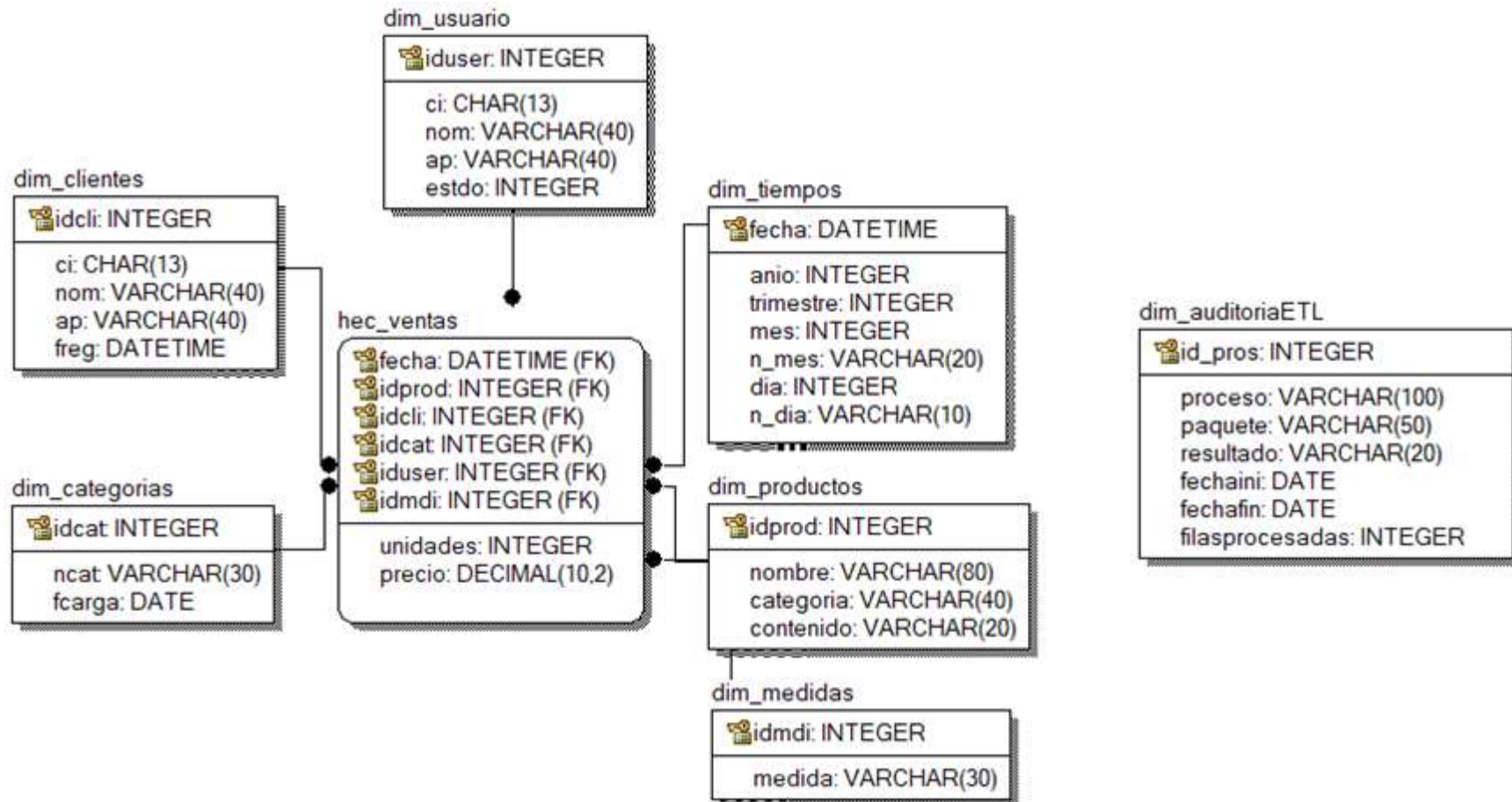
- ✓ Analizar y depurar los datos disponibles para el diseño del aplicativo
- ✓ Diseñar una interfaz gráfica amigable, fácil, ligera y adaptable a cambios
- ✓ Hacer del sistema la base principal de información para la toma de decisiones.

2.4 Diseño del prototipo

2.4.1 Casos de uso



2.4.2 Modelo Lógico (Esquema estrella)



2.4.3 Diccionario de datos

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
	idprod	int	NOT NULL	NO	SI
Hec_ventas	idcli	int	NOT NULL	NO	SI
	iduser	int	NOT NULL	NO	SI
	idcat	int	NOT NULL	NO	SI
	idmdi	int	NOT NULL	NO	SI
	unidades	int	NOT NULL	NO	NO

Tabla de hechos ventas: relación (productos, categorías, clientes, Medidas, usuarios, Tiempos)

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
	fecha	datetime	NOT NULL	SI	NO
Dim_tiempos	anio	int	NOT NULL	NO	NO
	trimestre	int	NOT NULL	NO	NO
	mes	datetime	NOT NULL	NO	NO
	n_mes	Varchar(20)	NOT NULL	NO	NO
	dia	int	NOT NULL	NO	NO
	n_dia	Varchar(20)	NOT NULL	NO	NO

Tabla de dimensión # 1 tiempo

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
	idcli	int	NOT NULL	SI	NO
dim_cliente	ci	Char(13)	NOT NULL	NO	NO
	nom	Varchar(40)	NOT NULL	NO	NO
	ap	Varchar(40)	NOT NULL	NO	NO
	freg	date	NOT NULL	NO	NO

Tabla de dimensión # 2 Clientes

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
dim_productos	idprod	int	NOT NULL	SI	NO
	nprod	Varchar(40)	NOT NULL	NO	NO
	cat	Varchar(20)	NOT NULL	NO	NO
	contenido	date	NOT NULL	NO	NO

Tabla de dimensión # 4 Productos

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
dim_categorias	idprod	int	NOT NULL	SI	NO
	ncat	Varchar(40)	NOT NULL	NO	NO

Tabla de dimensión # 5 categorías

N.Tabla	N.Campo	T.Datos	Op.Null	¿Clave Primaria?	¿Clave Foránea?
dim_medidas	idmdi	int	NOT NULL	SI	NO
	medida	Varchar(40)	NOT NULL	NO	NO

Tabla de dimensión # 6 Medidas

2.4.4 Script de la base datos

Hec_ventas		
NOMBRE	TIPO	CLAVE
idprod	INT NOT NULL	REFERENCES dim_prod
idcat	INT NOT NULL	REFERENCES dim_categorias
idcli	INT NOT NULL	REFERENCES dim_clientes
fecha	INT NOT NULL	REFERENCES dim_tiempos
unidades	INT NOT NULL	
precio	DECIMAL(10,2)	

Dim_tiempos		
NOMBRE	TIPO	CLAVE
fecha	DATETIME	PRIMARY KEY
anio	INT NOT NULL	
trimestre	INT NOT NULL	
mes	INT NOT NULL	
n_mes	VARCHAR(20) NOT NULL	
dia	INT NOT NULL	
n_dia	VARCHAR(20)	

dim_clientes		
NOMBRE	TIPO	CLAVE
idcli	INT NOT NULL	PRIMARY KEY
ci	CHAR(13) NOT NULL	
nom	VARCHAR(40) NOT NULL	
ap	VARCHAR(40) NOT NULL	
freg	DATETIME	

dim_categorias		
NOMBRE	TIPO	CLAVE
idcat	INT NOT NULL	PRIMARY KEY
ncat	CHAR(40) NOT NULL	

Hec_ventas		
NOMBRE	TIPO	CLAVE
idprod	INT NOT NULL	PRIMARY KEY
nombre	VARCHAR(50) NOT NULL	
cate	VARCHAR(40) NOT NULL	
contenido	INT NOT NULL	

dim_usuarios		
NOMBRE	TIPO	CLAVE
iduser	INT NOT NULL	PRIMARY KEY
ci	CHAR(13) NOT NULL	
nomb	VARCHAR(40) NOT NULL	
ap	VARCHAR(40) NOT NULL	

auditoria_ETL		
NOMBRE	TIPO	CLAVE
Id_pros	INT NOT NULL	PRIMARY KEY
proceso	VARCHAR(100) NOT NULL	
paquete	VARCHAR(50) NOT NULL	
resultado	VARCHAR(20) NOT NULL	
fechaini	DATETIME	
fechafin	DATETIME	
filasprocesadas	INT NOT NULL	

2.5 Ejecución y/o ensamblaje del prototipo



SIAGRO 1.0 [GERENCIA]

2455976

1112771.18

467

viernes, 13 de diciembre de 2019
13:50:56

SIAGR[®]
Sistema de Información para Agricultores Ver 1.0

USUARIO SISTEMA

SIAGRO 1.0 [GERENCIA]

DEVALLE DE VENTAS Y DIA

Fecha: 13/12/2019 13:55:50
Reporte: 62013

180000
160000
140000
120000
100000
80000
60000
40000
20000
0

AÑO	MES	DIA	UNIDADES	TOTAL
2019	Noviembre	Jueves 14	3,100	7,490.00
		TOTAL X MES	3,100	7,490.00
2017	AGILL	Miércoles 5	465	1,596.00
		Jueves 6	1,505	803.07
		Viernes 7	2,679	1,305.04
		Sábado 8	610	141.40
		Lunes 10	11,425	1,121.00
		Martes 11	3,104	507.00

viernes, 13 de diciembre de 2019
13:59:19

USUARIO SISTEMA

SIAGRO 1.0 [GERENCIA]

DEVALLE DE ETL

INFORME	USUARIO	REALIZADO	PRIMA-BOCETO	PRIMA-FIN	ETLAS-PROCESADAS	TIEMPO
fac_venta	ETL_venta	2010300	10/12/2019 20:49	10/12/2019 20:51	1746	79
dat_venta	ETL_venta	2010300	10/12/2019 20:49	10/12/2019 20:50	0	21
dat_producto	ETL_producto	2010300	10/12/2019 20:49	10/12/2019 20:50	0	26
dat_usuario	ETL_usuario	2010300	10/12/2019 20:49	10/12/2019 20:50	0	16
dat_registro	ETL_registro	2010300	10/12/2019 20:49	10/12/2019 20:50	0	21

viernes, 13 de diciembre de 2019
14:00:21

USUARIO SISTEMA

CAPITULO 3

EVALUCACIÓN DEL PROTOTIPO

3.1 Plan de evaluación

CRONOGRAMA DE ACTIVIDADES DEL DESARROLLO DEL APLICATIVO		
FECHA	ACTIVIDAD	RESPONSABLE
16/05/2019	Entrevista con el gerente.	Omar Bernal Villamar
20/05/2019	Recopilación y levantamiento de requerimientos del sistema.	Omar Bernal Villamar
21/05/2019	Instalación de herramientas necesarias para el desarrollo de la aplicación	Omar Bernal Villamar
25/05/2019	Migración de DB antigua al gestor SQL Server	Omar Bernal Villamar
28/05/2019	Realización de la DB. Utilizando los requerimientos del prototipo	Omar Bernal Villamar
04/06/2019	Análisis de requerimiento mediante la metodología de construcción Hefesto.	Omar Bernal Villamar
08/06/2019	Creación del modelo físico de la DB en el gestor SQL Server	Omar Bernal Villamar
11/06/2019	Creación de relaciones entre tablas ETL en tablas estáticas	Omar Bernal Villamar
15/06/2019	Creación de procesos, trabajos e inicios de sesión en SQL Server	Omar Bernal Villamar
05/06/2019	Comprobación de la BD realizando pruebas básicas para descartar errores	Omar Bernal Villamar
29/06/2019	Implementación de paquetes SSIS en SQL Server	Omar Bernal Villamar
06/07/2019	Configuración de catálogos, asignación de inicio de sesión, creación de servidor proxy en el agente, creación de tareas y pasos de integration services para automatizar ETL en SQL Server	Omar Bernal Villamar
09/07/2019	Creación de ETL(Extracción, transformación y lectura) principal en Integration Services	Omar Bernal Villamar
11/07/2019	Creación del cubo OLAP y las dimensiones en Analysis Services	Omar Bernal Villamar
13/07/2019	Procesar cubo con sus respectiva granularidad	Omar Bernal Villamar
16/07/2019	Creación del servidor de informes (Reporting Services) en C#.	Omar Bernal Villamar

19/07/2019	Configuración de puerto, inicio de sesión y ejecución del servidor de informes (Reporting services).	Omar Bernal Villamar
23/07/2019	Creación de reportes	Omar Bernal Villamar
26/07/2019	Desarrollo de interfaz. Pruebas de conexión entre C#, Analysis Services, Integration Services y Reporting Services con SQL Server.	Omar Bernal Villamar
03/08/2019	integración e implementación de proyectos y módulos.	Omar Bernal Villamar
06/08/2019	Elaboración del prototipo. Desarrollo del formulario de inicio de sesión.	Omar Bernal Villamar
07/08/2019	Desarrollo de interfaz. Elaboración de menú principal del sistema.	Omar Bernal Villamar
10/08/2019	Desarrollo de interfaz. Conexión al servidor de informes, Integration Services y Analysis Services de SQL Server desde la interfaz C#.	Omar Bernal Villamar
13/08/2019	Desarrollo de interfaz. Elaboración de validaciones y tiempos de respuesta.	Omar Bernal Villamar
14/08/2019	Pruebas y testeo.	Omar Bernal Villamar

3.2 Resultados de la evaluación

El aplicativo informático desarrollado e implementado en la distribuidora de productos agrícolas “Agrolive”. Se realizaron las pruebas correspondientes en tiempo real en el departamento de gerencia, consecuente a los resultados obtenidos el gerente se mostró muy satisfecho al constatar que el sistema cumple con los requerimientos establecidos y reconoce que la aplicación supera las expectativas planteadas para cubrir las falencias de toma de decisiones de dicho establecimiento.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Con base en lo antes presentado, se puede concluir que se lograron los principales objetivos del proyecto, ya que se desarrolló una plataforma de escritorio la cual logró aplicar las técnicas de ETL para el procesamiento de los datos originados por las transacciones de la distribuidora. La implementación de nuevos procesos de tecnologías como lo es el proyecto SIAG, ayudará a optimizar el proceso de toma de decisiones fortaleciendo la operatividad, con el fin de generar operaciones competitivas y exitosas.

La eficiencia en la obtención de informes detallados con gráficos, el gerente podrá analizar los resultados para hacer más eficiente la toma de decisiones. Así como también generar nuevas estrategias administrativas.

El uso de esta herramienta generará cambios positivos dentro de la distribuidora, siendo de gran ayuda en el proceso para alcanzar el éxito y lograr una rentabilidad. Gracias a esto, el gerente contará con mayor tiempo para el análisis de resultados lo cual favorecerá a la creación de nuevas estrategias de mercado, mejorando así el proceso de toma de decisiones. Los beneficios que se obtendrán al utilizar esta arquitectura de inteligencia de negocios se podrán percibir dentro de un mediano o largo plazo de tiempo, reflejando las ventajas competitivas derivadas del uso óptimo de la arquitectura y del conocimiento generado.

Recomendaciones

Como recomendación final se sugiere mantener el S.O con las últimas actualizaciones disponibles (Programar en horas no laborables). A futuro se deberá realizar un plan de mantenimiento y actualización del hardware según los avances tecnológicos como el que se dispone actualmente y así mantenerlo a la vanguardia tecnológica; ya que es de gran importancia para mantener el correcto funcionamiento del Data warehouse debido a su demandante crecimiento.

Bibliografía

- Duque Méndez, N., & Tamayo Alzate, A. (2001). Data Warehouse (bodega de datos). Herramienta para la toma de decisiones (parte I). *Noos : Revista Del Departamento de Ciencias*, (12), 118–126.
- Hadley L., "Developing a Data Warehouse Architecture", extraído el 28 de febrero de 2003 de <http://www.users.qwest.net/~lauramh/resume>.
- Kimball R., " Build your DataWarehouse one piece at a time", *Datawarehouse Designer*, Octubre 2002. Extraído el 4 de marzo del 2003 de <http://www.iemagazine.com/021030>
- Kimball R., "Is Data Staging Relational?" *DBMS*, April 1998. Extraído el 26 de febrero de 2003 de <http://www.dbmsmag.com>
- Kimball R., Laura Reeves, Margy Ross & Warren Thornthwaite, *The Data Warehouse Life Cycle Toolkit*, New York : John Wiley & Sons, 1998.
- Flores, D. A., & Martínez Chirichian, R. (2017). Desarrollo de Data Warehouse Comercial con Software de Control de Stock y Reportes. Retrieved from <https://rdu-dev.iua.edu.ar/handle/123456789/656>
- Adriana Patricia Muñoz Zapata, L. C. C. (2011). No Title p . *Phys. Rev. E*, 24. Retrieved from http://ridum.umanizales.edu.co:8080/jspui/bitstream/6789/377/4/Muñoz_Zapata_Adriana_Patricia_Articulo_2011.pdf
- AGUILAR, J. (n.d.). Datawarehousing : Modelado de Datos y OLP Modelado de Datos y de Información Usuarios tienen diferentes vistas de los datos. *Universidad de Los Andes*, 84.
- Aqüicultura, P. D. E. P. E. M., Donalek, J. G., Soldwisch, S., Coesão, E. D. E., Moreira, M. A., Fernandes, R. F., ... Jose Perona, J. (2018). No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title. □□□□ □□□□ □□ □□□□ □□ □□□□, □□□□ □□□□(1), 43. <https://doi.org/10.1017/CBO9781107415324.004>
- Bernabeu, R. D. (2010). *Hefesto Data Warehouseingvkcvandv*. 146. Retrieved from <http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/hefesto-metodologia-propia-para-la-construccion-un-data-warehttp://www.dataprix.com/data-warehousing-y-metodologia-hefesto/ii-hefesto-metodologia-propia-para-la-construccion-un-data-wa>
- Bruno, L. (2019). 済無No Title No Title. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Cabrera Torres, A. A., Moran Cabrera, E., & Rodriguez Anormaliza, R. I. (2015). Uso de la Tecnología Data Warehouse en Unidades Educativas de Nivel Medio: Consideraciones Teóricas. *Ciencia Unemi*, 7(11),

51. <https://doi.org/10.29076/issn.2528-7737vol7iss11.2014pp51-57p>

Castelán, L., & Ocharán, J. (2012). Diseño de un Almacén de datos basado en Data Warehouse Engineering Process (DWEPE) y HEFESTO. *Universidad Veracruzana*, 10. Retrieved from <http://www.uv.mx/mis/files/2012/11/Diseno-de-un-Almacen-de-datos.pdf>

Duque Méndez, N., & Tamayo Alzate, A. (2001). Data Warehouse (bodega de datos). Herramienta para la toma de decisiones (parte I). *Noos : Revista Del Departamento de Ciencias*, (12), 118–126.

Eguila, A. (2007). Implementación De Una Herramienta De Inteligencia De Negocios Para La Administración De Justicia Sobre Una Metodología Ad-Hoc. *Tesis*, 145.

Flores, D. A., & Martínez Chirichian, R. (2017). *Desarrollo de Data Warehouse Comercial con Software de Control de Stock y Reportes*. Retrieved from <https://rdu-dev.iaa.edu.ar/handle/123456789/656>

Juan Grabiél Arqueros Lavado. (2014). *UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA "JUSTIFICACIÓN DEL USO DE UN SISTEMA DE INTELIGENCIA DE NEGOCIOS" INFORME PRÁCTICO DE SUFICIENCIA*. 27. Retrieved from http://repositorio.unapikitos.edu.pe/bitstream/handle/UNAP/4517/Juan_Tesis_Titulo_2014.pdf?sequence=1&isAllowed=y

KEMP DE LA HOZ, E. A. K. (2005). *Desarrollo de una metodología que permita a empresas el desarrollo de un Data Warehouse y su integración con Sistemas Workflow utilizando herramientas de libre distribución y/o bajo costo*. 151.

Kimball and Inmon. (2012). Enfoques de desarrollo DW Antecedentes. *2012*, 33;Creación:5 mayo 2012;Recuperado:29 marzo 2015. Retrieved from http://www.interaktiv.cl/blog/wp-content/uploads/2012/04/4.-Metodologia_disegno_DW1.pdf

Saib, W. (2012). *Diseño Y desarrollo de una aplicación en .net para el Control De Empleados Y Gestión De Tareas*. Retrieved from <http://repositorio.upct.es/xmlui/bitstream/handle/10317/2717/pfc4259.pdf?sequence=1&isAllowed=y>

Sql, M., Transact, S., & Básico, S. Q. L. (2019). *Manual SQL Server – Transact SQL Básico /Avanzado*.

Valencia Arcos, J. del C., & Guevara Lenis, J. E. (2007). *Data warehouse para el análisis académico de la Escuela Politécnica Nacional*. 1–135. Retrieved from <http://bibdigital.epn.edu.ec/handle/15000/445>

Vardaro, M. J., Systems, H. I. T., AG, H. T., Jari, A., Pentti, M., Information, B. G., ... Measurements, C. (2016). No Title. *2002*(1), 35–40. <https://doi.org/10.1109/ciced.2018.8592188>