

# **Guía práctica para Arquitecturas de Datos Empresariales**

**Alain Osorio Rodríguez**

Universidad de las Ciencias Informáticas, La Habana, Cuba, [aorodriguez@uci.cu](mailto:aorodriguez@uci.cu)

**Mairelys Martínez Lopez**

Universidad de las Ciencias Informáticas, La Habana, Cuba, [mmlopez@uci.cu](mailto:mmlopez@uci.cu)

## **ABSTRACT**

The data architecture is considered a key factor for the success of management information systems and business organizations. Integrates models, policies and rules that govern what data will be collected, how it will be stored, sorted and put into use by certain technological infrastructure.

There is abundant and varied literature on developing elements of data architecture and theories explaining why these elements should be developed in a certain way. Still incurred malpractices that lead to errors, sometimes for lack of experience by misinterpretation and other technologies for deployment in a given environment.

This paper defines a series of orderly steps in order to establish a guide for structuring enterprise data architectures, integration consecutive continuous and consistent, decisions to be taken when developing poses a software architecture. These steps have been defined by the work and experience in developing databases for medium and large institutions for more than six years, and the research and application of best practices in this field.

**Keywords:** architecture, database, development guide, quality assurance.

## **RESUMEN**

La Arquitectura de Datos se considera un factor clave para el éxito de los sistemas de gestión de información y organizaciones empresariales. Integra modelos, políticas y reglas que rigen qué datos serán recopilados, cómo serán almacenados, ordenados y puestos en uso mediante determinada infraestructura tecnológica.

Existe abundante y variada bibliografía sobre cómo desarrollar elementos que componen una arquitectura de software y teorías que explican el por qué se deben desarrollar estos elementos de cierta manera. Aun así se incurre en malas prácticas que conllevan a errores, en ocasiones por falta de experiencia y otras por mala interpretación de las tecnologías para su posterior implementación en determinado entorno.

El presente trabajo define una serie de pasos ordenados con el objetivo de establecer una guía para la estructuración de arquitecturas de datos empresariales, una integración consecutiva, continua y consecuente, de decisiones que se deben tomar cuando se plantea el desarrollo de una arquitectura de datos. Estos pasos se han definido mediante el trabajo y la experiencia en el desarrollo de bases de datos para medianas y grandes instituciones durante más de 6 años, así como la investigación y aplicación de buenas prácticas en este campo.

**Palabras claves:** arquitectura, base de datos, guía de desarrollo, aseguramiento de la calidad.

## 1. INTRODUCCIÓN

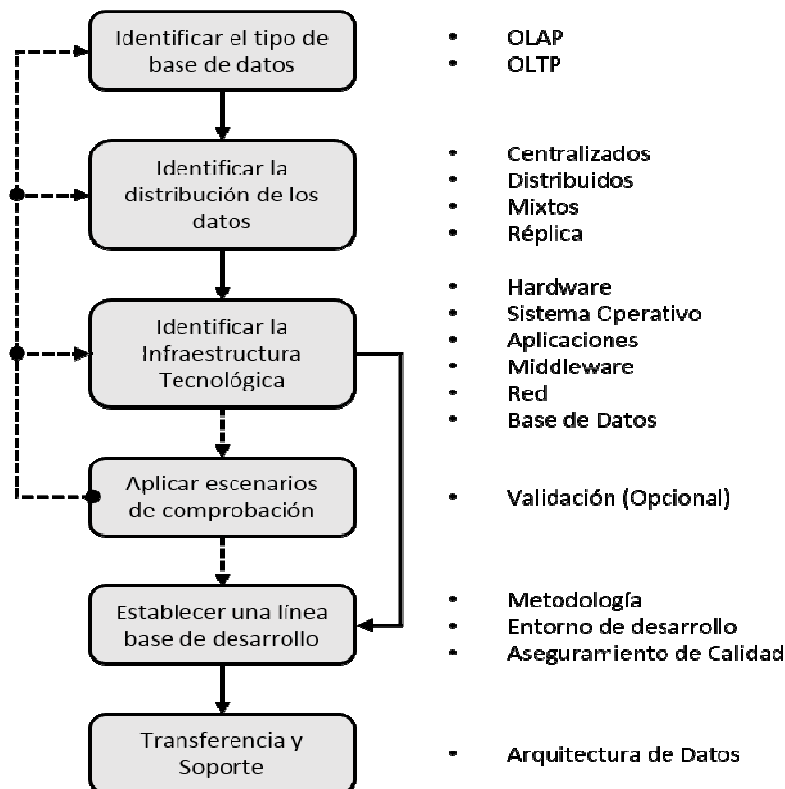
La Arquitectura de Datos en las Tecnologías de Información está compuesta por modelos, políticas, reglas o estándares que regulan qué datos son recopilados, cómo son almacenados, ordenados, integrados y puestos en uso en sistemas de datos y organizaciones. Es uno de varios dominios arquitectónicos que forman el pilar de una arquitectura empresarial o solución arquitectónica. (WebFinance, Inc, 2013)

Se pueden obtener "buenos" resultados desarrollando software organizada o desorganizadamente. La diferencia fundamental se evidencia en el esfuerzo de desarrollo, la calidad de la arquitectura de datos medida por el balance entre sus atributos de calidad y las facilidades de soporte y administración una vez que todo el sistema esté en funcionamiento en su entorno real de ejecución.

El presente trabajo define una serie de pasos ordenados con el fin de establecer una guía que pretende describir lo que se debería desarrollar en materia de Arquitectura de Datos Empresarial. Entiéndase como Empresarial a cualquier organismo, institución o empresa de mediano a gran tamaño. Para ello se describen un grupo de conceptos relacionados con la Arquitectura de Software y tecnología de base de datos, con el objetivo de establecer un camino guiado en la definición de una Arquitectura de Datos Empresarial.

## 2. DESARROLLO

A continuación se describen cada uno de los pasos propuestos para la estructuración de Arquitecturas de Datos Empresariales. La guía consta de 6 pasos:



**Figura 1: Guía práctica en 6 pasos**

## **2.1 IDENTIFICAR EL TIPO DE BASE DE DATOS A UTILIZAR**

Antes de todo se debe identificar el tipo de base de datos a utilizar según el propósito para el cual será concebida. Existen dos grandes grupos o tipos: bases de datos operacionales o procesamiento de transacciones -OLTP- y bases de datos para la toma de decisiones o procesamiento analítico -OLAP. Las bases de datos OLTP están concebidas para operar 24 horas los 7 días de la semana 365 días del año, mientras que las bases de datos OLAP se estructuran para realizar operaciones de Inteligencia de Negocio (Business Intelligence - BI) en determinado período. Las bases de datos OLTP se caracterizan por el diseño relacional y las bases de datos OLAP se caracterizan por el diseño multidimensional.

### **2.1.1 BASES DE DATOS OPERACIONALES**

Algunos tipos de base de datos que califican dentro del nivel operacional son las que se mencionan a continuación:

- Bases de datos para el procesamiento de transacciones en línea
- Bases de datos de propósito general
- Base de datos embebida
- Base de datos NoSQL

### **2.1.2 BASES DE DATOS PARA LA TOMA DE DECISIONES**

Los principales tipos de base de datos que califican dentro del nivel de toma de decisiones son las que se mencionan a continuación:

- Almacén de Datos (Data warehouse - DWH)
- Mercado de Datos (Data Mart - DM)
- Almacén de Datos Operacional (Operational Data Store - ODS)

## **2.2 IDENTIFICAR LA DISTRIBUCIÓN DE LOS DATOS Y COMUNICACIÓN**

Una vez seleccionado el tipo de base de datos que se debe utilizar, se debe identificar la distribución de los datos y su comunicación según el entorno funcional del sistema empresarial. Esta distribución está fuertemente vinculada a la distribución física de las fuentes de datos según las necesidades de la empresa y sus departamentos. Existen tres maneras de distribuir los datos:

- Centralizada
- Distribuida
- Mixta

El entorno de aplicaciones de procesamiento de grandes transacciones suele ser mayormente centralizado. El entorno de aplicaciones de procesamiento de transacciones de poca carga de operaciones suele ser distribuido. En teoría es posible tener un entorno distribuido para un ODS, pero en la práctica el entorno para un ODS es centralizado. Los Almacenes de Datos pueden ser centralizados o distribuidos pero en la mayoría de los casos son centralizados. Los Mercados de Datos son propiamente distribuidos ya que son personalizados para cada departamento y todos se nutren del Almacén de Datos, por lo que el entorno pudiera catalogarse como mixto cuando existan datos comunes e intercambio de datos entre estos.

### **2.2.1 RÉPLICA DE DATOS**

La réplica de datos no es más que el mecanismo automático de sincronización de datos distribuidos en diferentes fuentes de datos. En un ambiente de réplica, los servidores que pueden modificar los datos se denominan servidores de lectura/escritura, maestros o primarios. Los servidores que están al tanto de los cambios en los servidores maestros se denominan standby o esclavos. Un servidor esclavo que no pueda ser accedido hasta que

sea promovido a un servidor maestro es denominado servidor warm standby, y uno que pueda ser accedido mediante consultas de solo lectura es denominado servidor hot standby (The PostgreSQL Global Development Group, 2011). La comunicación puede ser mediante servidores maestros-esclavos o maestro-maestro.

Las soluciones de réplica pueden ser sincrónicas o asincrónicas. En las soluciones sincrónicas, una transacción no se considera terminada hasta que esta haya sido ejecutada satisfactoriamente en cada uno de los servidores. Esto garantiza que todos los servidores estén actualizados en tiempo real y que todos devolverán la misma información sin importar a que servidor se realizó la consulta. Las soluciones asincrónicas permiten alguna demora entre el tiempo en que se hizo efectiva una transacción y su propagación hacia otros servidores, abriendo la posibilidad que algunas transacciones puedan perderse en el cambio hacia un servidor de respaldo y que los servidores puedan no retornar los mismos resultados. La solución asincrónica es utilizada cuando la solución sincrónica pueda ser muy lenta. El rendimiento es un elemento muy importante a considerar en cualquier opción.

Las soluciones de réplica también se pueden categorizar por su granularidad. Algunas soluciones manejan todo el servidor de base de datos, mientras otras se manejan a nivel de base de datos o a nivel de tablas (The PostgreSQL Global Development Group, 2011).

Es muy importante seleccionar la distribución correcta de los datos según su propósito y manipulación, ya que cada entorno tiene especificaciones muy diferentes de diseño que condicionan todo el sistema.

### **2.3 IDENTIFICAR LA INFRAESTRUCTURA TECNOLÓGICA QUE SOPORTA LA ARQUITECTURA DE DATOS**

Toda infraestructura tecnológica empresarial requiere de especificaciones de hardware, sistema operativo, aplicaciones, middleware -programas intermedios-, base de datos y redes. Todos estos componentes se deben coordinar integralmente para posibilitar una infraestructura robusta, escalable, segura y consecuente con la metas de negocio de la empresa.

Se identifica como infraestructura tecnológica un grupo de recursos integrados de hardware y software para brindar servicios como por ejemplo correo electrónico, internet, directorios de personas, bases de datos, aplicaciones, entre otras.

En este punto hay dos variantes posibles: que los clientes para los cuales se desarrolla el sistema cuenten en su organización con una infraestructura tecnológica que satisfaga determinados servicios y/o necesidades; o que no tengan una infraestructura.

#### **2.3.1 INFRAESTRUCTURA EXISTENTE**

Si los clientes poseen una infraestructura tecnológica se debe valorar:

1. Utilizar la infraestructura tal como está concebida: si esta soporta la Arquitectura de Datos que se va a estructurar. Esta opción es muy viable ya que la Arquitectura de Datos se integraría a los servicios que esta brinda y se eliminan las dificultades de mantener diferentes infraestructuras independientes.
2. Plantearse una infraestructura paralela: si se va a definir una infraestructura independiente pero compartiendo algunos recursos y/o servicios con la infraestructura existente. Por ejemplo recursos de almacenamiento y aplicaciones.
3. Plantearse una infraestructura independiente: si se va a definir una infraestructura sin tener en cuenta los recursos que dispongan los clientes, ya sea porque pudiera ser perjudicial a la infraestructura existente o los recursos y servicios que esta brinde no sean de utilidad.
4. Redefinir la infraestructura para que soporte los nuevos requerimientos: solo cuando la Arquitectura de Datos se defina como el centro de esa infraestructura y se redefinan los recursos y servicios que brinda la organización a sus usuarios. Esta opción es la más radical y solo se debe implementar cuando sea realmente pertinente.

En ninguno de los casos la implementación de la Arquitectura de Datos en la infraestructura tecnológica debe interferir negativamente en los servicios que brinda esa infraestructura, ya que esta cumple un objetivo dentro de la organización.

### 2.3.2 INFRAESTRUCTURA NO EXISTENTE

Cuando la organización carece de una infraestructura tecnológica, esta se debe definir teniendo en cuenta requerimientos actuales y futuros. Sería un grave error estructurar una infraestructura tecnológica que tenga limitaciones en poco tiempo. La escalabilidad es un requisito importante a considerar cuando se despliegan nuevos sistemas o se actualizan. Cuando se definen nuevos requerimientos tecnológicos se debe estimar el tamaño inicial de los datos y su crecimiento, así como número de usuarios actuales y futuros y su demanda. También se debe estimar la capacidad de los recursos como procesadores, memoria y almacenamiento que serán necesarios en el tiempo de vida de operación de los sistemas en un lapso promedio de 5 años.

**Tabla 1: Elementos a considerar en la definición de la Infraestructura Tecnológica que soporte una Arquitectura de Datos.**

Elemento	Especificaciones a considerar
Hardware	<ul style="list-style-type: none"> <li>• Procesadores</li> <li>• Memoria</li> <li>• Sistemas de almacenamiento y salvados               <ul style="list-style-type: none"> <li>○ Arreglos de discos RAID</li> <li>○ Almacenamiento alternativo para los datos históricos</li> </ul> </li> <li>• Clusterización</li> <li>• Licencias y soporte</li> </ul>
Sistema Operativo	<ul style="list-style-type: none"> <li>• Virtualización</li> <li>• Compatibilidad con el resto de los componentes</li> <li>• Licencias y soporte</li> </ul>
Sistema Gestor de Base de Datos	<ul style="list-style-type: none"> <li>• Soporte a los principales requerimientos de datos</li> <li>• Mecanismo de réplica</li> <li>• Alta disponibilidad</li> <li>• Balanceo de Carga</li> <li>• Licencias y soporte</li> </ul>
Middleware	<ul style="list-style-type: none"> <li>• Identificar requerimientos e integración entre aplicaciones y la arquitectura de datos</li> <li>• Herramienta de réplica</li> </ul>
Aplicaciones	<ul style="list-style-type: none"> <li>• Identificar las aplicaciones que integran el ambiente empresarial</li> <li>• Optimizar operaciones de persistencia y recuperación de datos</li> </ul>
Redes	<ul style="list-style-type: none"> <li>• Identificar requerimientos según la distribución y accesibilidad de los datos</li> <li>• Seguridad y monitoreo de la actividad de los usuarios</li> </ul>

Los requerimientos tecnológicos para bases de datos operacionales y bases de datos para la toma de decisiones no se deben combinar. Existe un patrón muy diferente de utilización de recursos de hardware para el ambiente OLTP que el que existe para el ambiente OLAP. Los datos pueden ser optimizados físicamente para el acceso y procesamiento transaccional o acceso y procesamiento para sistemas para la toma de decisiones, pero no para ambos al mismo tiempo en el mismo equipamiento.

## 2.4 APLICAR ESCENARIOS DE COMPROBACIÓN

Es muy común tener diferentes variantes de selección de los puntos anteriormente descritos. Este paso se define como opcional, y tiene como objetivo definir la variante que sea más conveniente, consecuente y acorde con temas de financiación y presupuesto para lograr los objetivos de negocio de la empresa. También puede ser útil en la validación del escenario seleccionado en caso de que exista cierto grado de inseguridad o incertidumbre en el cumplimiento de los requerimientos identificados.

Quizás solo exista una variante correcta, pero si existe más de una variante, se deben implementar tantos escenarios según sea conveniente, a menor escala, como prototipo arquitectónico de comprobación. Esto no es más que definir un entorno de prueba para la certificación de todos los componentes que forman la Arquitectura de Datos Empresarial. Luego se escogería el más adecuado para continuar con su implementación.

## **2.5 ESTABLECER UNA LÍNEA BASE DE DESARROLLO**

Existe abundante teoría y bibliografía de base de datos. Teoría y bibliografía de diseño, programación en SQL y lenguajes de manipulación de datos, administración y configuración según el sistema gestor seleccionado. Pero no abunda teoría de cómo coordinar intencionada y justificadamente todos los elementos requeridos para el desarrollo de una base de datos en un proceso de desarrollo de software. Las metodologías y procesos de desarrollo de software no estructuran el desarrollo de bases de datos. Rational Unified Process (RUP) (IBM Corporation, 2013) aborda el desarrollo de base de datos como una tarea de diseño, careciendo de especificaciones necesarias en todo el esfuerzo de desarrollar y mantener una base de datos. Las metodologías ágiles no tienen distinción entre desarrollo de aplicaciones y desarrollo de base de datos dentro del desarrollo de aplicaciones, careciendo, como RUP, de detalles generales que son aplicables siempre. Las metodologías que se consideran parte ágil y parte tradicional como Agile Unified Process (AUP) (Ambler, 2013) heredan los mismos problemas. Agile Data Method (ADM) (Ambler, 2013) es un buen enfoque ya que define un grupo de buenas prácticas desde el punto de vista ágil, pero no define una continuidad durante todo el ciclo de desarrollo. DBPLANNING (Rodríguez, 2012) es un framework de desarrollo de base de datos que posibilita al equipo de desarrollo establecer una línea base de desarrollo de base de datos.

El framework DBPLANNING transita por las fases de Inicio, Desarrollo y Despliegue. Plantea 4 actividades: Modelado de Datos, Configuraciones, Implementación y ADTP (Acceso a Datos, Tuning y Prueba), las cuales se desarrollan durante las 3 fases definidas. Las actividades establecen relaciones las cuales no se deben violar.

El framework asume los roles de Arquitecto de Datos, Administrador de Base de Datos y Desarrollador de Base de Datos para la ejecución de las actividades y la elaboración de los 6 artefactos que define.

El Arquitecto de Datos es el encargado de la realización de la actividad de Modelado de Datos y la elaboración de los artefactos Arquitectura de Datos y Modelo de Datos. Tiene mayor importancia en la fase de Inicio, por lo que en la fase de Inicio de todo desarrollo de software es necesario una o varias personas que realicen una buena labor como Arquitecto de Datos para garantizar la Arquitectura de Datos del sistema.

El Desarrollador de Base de Datos es el encargado de la realización de las actividades de Implementación y ADTP, y la elaboración de los artefactos Código Fuente, Scripts de Escenarios y Acceso a Datos. Tiene mayor importancia en la fase de Desarrollo, por lo que en la fase de Desarrollo de todo desarrollo de software es necesario una o varias personas que realicen una buena labor como Desarrollador de Base de Datos para garantizar un correcto funcionamiento de las operaciones de almacenamiento y recuperación de la información de la Arquitectura de Datos del sistema.

El Administrador de Base de Datos es el encargado de la realización de la actividad de Configuraciones y la elaboración del artefacto Configuraciones. Tiene mayor importancia en la fase de Despliegue, por lo que en la fase de Despliegue de todo desarrollo de software es necesario una o varias personas que realicen una buena labor como Administrador de Base de Datos para garantizar la correcta implementación tecnológica de la Arquitectura de Datos del sistema, así como futuro soporte y administración.

Mediante la aplicación de este framework es posible una coordinación justificada e intencionada de los principales elementos de base de datos dentro del desarrollo de software.

### 2.5.1 AMBIENTES DE DESARROLLO

El entorno de desarrollo debe tener las mismas especificaciones de arquitectura de datos que el futuro entorno de producción para evitar problemas y ser consecuente con todos los requerimientos identificados.

Cuando se establezca un entorno de pruebas, es aconsejable (Bob Bryla, 2008):

- Definir un entorno mayor que el de producción para pronosticar requerimientos futuros.
- Debe contener datos conocidos, planes de ejecución, resultados de rendimiento, y resultados de datos.
- Debe ser utilizado para cada liberación de la base de datos y herramientas, así como para nuevas características.
- Debe ser utilizado activamente como parte de la implementación tecnológica.
- Probar toda la gama de operaciones de persistencia y recuperación de datos, todo tipo de sentencias DML y la mayoría de sentencias DDL.
- Probar con grandes volúmenes de datos.
- Probar con gran cantidad de usuarios concurrentes.

Las pruebas pre-producción se deben desarrollar en el hardware de producción. Mover la arquitectura a una infraestructura diferente requiere probar nuevamente todo el entorno.

### 2.5.2 ASEGURAMIENTO DE CALIDAD

El aseguramiento de la calidad en cada uno de los elementos que componen un sistema es de suma importancia. La calidad de cada una de las estructuras que componen una arquitectura garantiza los principales requisitos no funcionales. El aseguramiento de la calidad de cada uno de los elementos que conforman una arquitectura de datos está condicionado por los propios elementos que la conforman dentro de un entorno específico, no se puede especificar genéricamente.

El aseguramiento de la calidad para arquitecturas de datos empresariales está muy vinculada a las decisiones arquitecturales que se tomen tal y como se definen en los pasos de esta guía. Primero se debe seleccionar correctamente el tipo de bases de datos a utilizar según su propósito. Tomar la decisión correcta a la hora de seleccionar el tipo de bases de datos es el primer paso para asegurar la calidad de la arquitectura de datos.

Luego se debe seleccionar la distribución correcta de los datos según el entorno funcional de la empresa. La distribución de los datos no debe crear un caos. La comunicación entre las fuentes de datos debe ser fluida, acorde a las necesidades de distribución de los datos, y simple en la mayoría de los casos. Si la comunicación entre las fuentes de datos distribuidas no es fluida, es compleja o casi imposible de administrar en determinado entorno, se debe considerarse una solución centralizada. Tomar la decisión correcta en este punto es vital sobre todo para la optimización y seguridad de los datos.

Una vez seleccionado el tipo de base de datos y su distribución, se debe seleccionar los requerimientos tecnológicos que los soporten. La decisión correcta va desde especificar requerimientos actuales hasta definir requerimientos futuros para evitar un entorno tecnológico con limitaciones a corto plazo.

Después de haber definido la arquitectura de datos, que no es más que seleccionar el tipo de base de datos, su distribución y tecnología que los soporta, se deben desarrollar los artefactos de base de datos (modelo de datos, configuraciones, código fuente, scripts de datos, vista de acceso a datos). El aseguramiento de la calidad de estos artefactos se garantiza siendo consecuente con la arquitectura de datos definida y están caracterizados por:

- Teorías de desarrollo de bases de datos
- Estándares de codificación
- Patrones y buenas prácticas en el diseño de las Bases de Datos

- Buenas prácticas de implementación en SQL y lenguajes de manipulación de datos
- Administración según el Sistema Gestor de Base de Datos seleccionado
- Optimización según el Sistema Gestor de Base de Datos seleccionado
- Gestión de configuraciones según el Sistema Gestor de Base de Datos seleccionado

DBPLANNING framework es muy útil en el aspecto del aseguramiento de la calidad de los artefactos ya que define cuales artefactos debemos desarrollar para asegurar una correcta implementación de determinada arquitectura de datos. Si no conocemos de antemano cuales artefactos debemos desarrollar, cuesta un poco más establecer una base para el aseguramiento de la calidad. El framework no define cómo se deben desarrollar los artefactos. Estos se deben desarrollar según las especificaciones de la arquitectura de datos, teoría y buenas prácticas.

## **2.6 TRANSFERENCIA TECNOLÓGICA, ADMINISTRACIÓN Y SOPORTE**

La etapa final de todo desarrollo de software es la migración hacia el entorno de producción, que no es más que el entorno funcional del sistema. Con esta migración, comúnmente conocida como despliegue, comienza un proceso de transferencia tecnológica, administración y soporte.

La transferencia tecnológica es el traspaso a los clientes de todas las descripciones arquitectónicas.

La administración, en un ambiente de producción, es el monitoreo de toda la infraestructura tecnológica durante su funcionamiento.

El soporte es servicio de ayuda técnica relacionada con la resolución de problemas detectados durante la administración del sistema, los cuales no han podido ser solucionados por los administradores de la parte cliente.

Antes de migrar al entorno de producción se deben especificar entregables como (Bob Bryla, 2008):

- Diagrama Entidad Relación
- Diagrama Físico de la base de datos
- Requerimientos de espacio
- Metas de optimización para consultas y procesamiento de transacciones
- Requerimientos de seguridad
- Requerimientos de datos
- Planes de ejecución de consultas
- Procedimientos de pruebas de aceptación

El artefacto que describe la arquitectura de datos es el principal actor de la arquitectura de datos en un ambiente de producción. En este deben quedar bien reflejados todos los elementos que la componen. El artefacto Arquitectura de Datos que plantea DBPLANNING está concebido genéricamente para que describa todos los elementos que componen una arquitectura de datos y sea consecuente con esta última.

## **3. CONCLUSIONES**

Este trabajo plantea una serie de pasos ordenados cuyo objetivo es establecer una ruta en la estructuración de arquitecturas de datos empresariales, una integración consecutiva, continua y consecuente, de decisiones que debemos tomar cuando se plantea el desarrollo de una arquitectura de datos.

Para definir una arquitectura de datos debemos seleccionar el tipo de base de datos, la distribución de los datos y la infraestructura que la soporta. Luego de comprobar que los elementos que conforman la arquitectura de datos son correctos, se debe establecer una línea base de desarrollo, muy útil para el aseguramiento de la calidad, acorde a las decisiones arquitecturales. Por último se establece como actor fundamental del entorno de producción al artefacto que describe la arquitectura de datos desarrollada.



Mediante la aplicación práctica de la teoría expuesta en este trabajo, es posible una adecuada estructuración de arquitecturas de datos para sistemas empresariales, ya que es posible identificar temprana y pertinentemente todos los elementos necesarios para esta, en determinado escenario. A su vez, la arquitectura que se desarrolle contará con características relevantes tales como escalabilidad, mejora en la administración de los cambios, guiada por un proceso que podrá ser auditado en cualquier fase del desarrollo y enfocado en el aseguramiento de la calidad.

## REFERENCIAS

Ambler, Scott W. 2013. Agile Data Home Page. [En línea] 2013. [Citado el: 9 de Enero de 2013.] <http://agiledata.org/>.

—. 2013. The Agile Unified Process (AUP). [En línea] 2013. [Citado el: 9 de Enero de 2013.] <http://www.ambysoft.com/unifiedprocess/agileUP.html>.

Bob Bryla, Kevin Loney. 2008. Oracle Database 11g DBA Handbook. s.l.: McGraw-Hill, 2008. 0-07-159579-1.

Claudia Imhoff, Nicholas Gallempo, Jonathan G. Geiger. 2003. Mastering Data Warehouse Design. Relational and Dimensional Techniques. Indianapolis: Wiley Publishing, Inc, 2003. ISBN 0-471-32421-3.

IBM Corporation. 2013. Rational Unified Process. 2013.

Inmon, W. H. 2005. Building the Data Warehouse, Fourth Edition. Indianapolis: Wiley Publishing, Inc., 2005. ISBN: 978-0-7645-9944-6.

Oracle Corporation. 2012. Oracle: Big Data for the Enterprise. [An Oracle White Paper] 2012.

Rodríguez, Alain Osorio. 2012. Modelo de Desarrollo de Base de Datos v2.0. VI Taller de Calidad, Ingeniería, Arquitectura y Gestión de Software. Habana: UCIENCIA, 2012. ISBN: 978-959-286-019-3.

The PostgreSQL Global Development Group. 2011. PostgreSQL 9.1.0 Documentation. 2011.

### ***Authorization and Disclaimer***

*Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.*