



Universidad de las Ciencias Informáticas

Facultad 3

Centro de Informatización de la Gestión de Entidades

---

## **“Definición de Arquitectura de Datos para Sistemas de Gestión Empresarial.”**

Trabajo final presentado en opción al título de Máster en  
Informática Aplicada

---

**Autor: Ing. Ariel Torres Gálvez**

**Tutor: Dr. C Pedro Piñero Pérez**

**Ciudad de la Habana, diciembre 2012**

### **Declaración Jurada de Autoría**

Declaro por este medio que yo Ariel Torres Gálvez, con carné de identidad 84101605147, soy el autor principal del trabajo final de maestría "Definición de Arquitectura de Datos para Sistemas de Gestión Empresarial", desarrollada como parte de la Maestría en Informática Aplicada y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los            días del mes de            del año            .

## **Resumen**

La Arquitectura de Software es uno de los elementos más importantes dentro del desarrollo de cualquier sistema de software y dentro de ella la Arquitectura de datos. La misma permite definir todos los cómo acerca del tratamiento que se le realizará a los datos durante todo el proceso. Si no se realizara una correcta definición, podría obtenerse un producto defectuoso, con dificultades para el soporte y mantenimiento del mismo tirando por la borda el esfuerzo de todo el proceso de desarrollo. Dentro del mundo empresarial, uno de los sistemas más complejos de desarrollar son los Sistemas de Gestión Empresarial o ERP como también se les conoce. Estos manejan un gran volumen de datos los cuales se encuentran divididos en un grupo considerable de subsistemas o módulos. Cada uno de estos módulos puede considerarse un proyecto por su complejidad. Para el desarrollo de estos sistemas las definiciones de la arquitectura de datos son fundamentales, ya que el intercambio de datos entre estos módulos en la mayoría de los casos es complejo. La información manejada es muy sensible, por lo que la seguridad de estos datos es un eje fundamental. El presente trabajo define las pautas necesarias en la disciplina con las cuales se pretende enfrentar el desarrollo de todos y cada uno de estos módulos.

**Palabras claves:** Arquitectura de datos, Sistemas de Gestión Empresarial, ERP

## **Abstract**

Software Architecture is one of the most important elements in the development of any software system and within the Data Architecture. It allows to define all the how's about the treatment that the data will be done throughout the process. If there is a correct definition, a defective product could be obtained with difficulty support and maintenance of the throwing away the effort of the whole process of development. Within the business world, one of the most complex systems are developing Enterprise Resource Planning or ERP as they are also known. They handle a large volume of data which are divided into a large group of subsystems or modules. Each of these modules can be considered a project of such complexity. For the development of these systems the definitions of the data architecture are essential, since the exchange of information between modules in most cases is very complex. Another element of weight to bear in mind is that this information is very sensitive, so that the security of this data is a cornerstone to consider. This paper we define the necessary guidelines in the discipline with which it is intended to face the development of all and each of these modules.

**Keywords:** Data Architecture, Enterprise Resource Planning, ERP

# INDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>1 CAPÍTULO 1 MARCO TEÓRICO</b> .....	<b>9</b>
1.1 INTRODUCCIÓN .....	9
1.2 ARQUITECTURA DE SOFTWARE .....	9
1.3 ARQUITECTURA DE DATOS .....	9
1.4 SISTEMAS DE GESTIÓN EMPRESARIAL .....	10
1.4.1 <i>Sistemas de Gestión Empresarial Propietarios</i> .....	10
1.4.2 <i>Sistemas de Gestión Empresarial Libre</i> .....	16
1.4.3 <i>Sistemas de Gestión Empresarial Cubanos</i> .....	23
1.5 FACTIBILIDAD DE SOLUCIÓN .....	27
1.6 CONCLUSIONES .....	28
<b>2 CAPÍTULO 2. PROPUESTA DE SOLUCIÓN</b> .....	<b>30</b>
2.1 INTRODUCCIÓN .....	30
2.2 RESTRICCIONES DE DISEÑO .....	30
2.3 ELEMENTOS DE LA VISTA DE DATOS .....	30
2.4 ESTRUCTURA DE LA CAPA DE DATOS .....	31
2.5 NORMAS DEL DISEÑO .....	31
2.5.1 <i>Patrones de solución</i> .....	31
2.5.2 <i>Comentarios</i> .....	35
2.5.3 <i>Estándares de nomenclatura</i> .....	35
2.5.4 <i>Soluciones genéricas</i> .....	38
2.6 DEFINICIÓN DE LAS BASES DE DATOS .....	39
2.6.1 <i>Estructura de la base de datos</i> .....	39
2.7 SEGURIDAD DE DATOS .....	43
2.7.1 <i>Acceso a los datos</i> .....	43
2.8 RÉPLICA .....	45
2.8.1 <i>Soluciones de réplica</i> .....	45
2.9 SCRIPT DE INSTALACIÓN .....	47
2.9.1 <i>Clasificación de los script</i> .....	47
2.9.2 <i>Plantillas de script</i> .....	48
2.10 TECNOLOGÍA .....	48
2.11 FORMALIZACIÓN .....	48
2.11.1 <i>Diccionario de datos</i> .....	49
2.11.2 <i>Estructura del Expediente</i> .....	49
2.12 CONCLUSIONES PARCIALES DEL CAPÍTULO .....	50
<b>3 CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN</b> .....	<b>52</b>
3.1 INTRODUCCIÓN .....	52
3.2 MÉTODO DE EVALUACIÓN DE ARQUITECTURA. QUASAR .....	52
3.2.1 <i>Evaluación de los factores de calidad</i> .....	53
3.3 EXPERIMENTO .....	58
3.4 CERTIFICACIÓN DEL SISTEMA .....	64
3.5 IMPACTO .....	65
3.6 CONCLUSIONES .....	65
<b>CONCLUSIONES GENERALES</b> .....	<b>67</b>
<b>RECOMENDACIONES</b> .....	<b>68</b>
<b>BIBLIOGRAFÍA</b> .....	<b>69</b>
<b>ANEXOS</b> .....	<b>72</b>

## INDICE DE FIGURAS

FIGURA 1 CONCURRENCIA EN EL ACCESO DE DATOS.....	32
FIGURA 3 ÁRBOL SIMPLE. (BLAHA, 2010).....	34
FIGURA 4 ÁRBOL SIMPLE APLICADO EN CEDRUX.....	34
FIGURA 5 ÁRBOL CON HIJOS.....	34
FIGURA 6 DENOMINACIÓN DE LAS BASE DE DATOS. ....	36
FIGURA 7 CREACIÓN DE ESQUEMAS EN POSTGRESQL. ....	36
FIGURA 8 CREACIÓN DE TABLA EN POSTGRESQL. ....	36
FIGURA 9 DENOMINACIÓN DE TABLAS.....	37
FIGURA 10 APARIENCIA DE LOS CAMPOS.....	37
FIGURA 11 NOMBRE DE LLAVES PRIMARIAS SEGÚN NOMBRE DE LA TABLA. ....	37
FIGURA 12 NOMBRE DE LLAVE FORÁNEA. ....	37
FIGURA 13 NOMBRE DE LOS TABLESPACE. ....	38
FIGURA 14 NOMBRE DE LOS DOMINIOS.....	38
FIGURA 15 NOMBRE DE LAS SECUENCIAS.....	38
FIGURA 16 TABLAS QUE COMPRENDE LA MULTIENTIDAD. ....	38
FIGURA 17 MULTIMONEDA.....	39
FIGURA 18 ESTRUCTURA ORGANIZACIONAL DE LA BASE DE DATOS EN ESQUEMAS. ....	40
FIGURA 19 ESTRUCTURA CON ROBUSTEZ.....	40
FIGURA 20 DISEÑO ESTRUCTURAL DE LA BASE DE DATOS. ....	41
FIGURA 21 FLUJO DE ACCESO EN CEDRUX. ....	44
FIGURA 22 MODELO DE DATOS DEL MÓDULO SEGURIDAD.....	55
FIGURA 23 DISEÑO DE BASE DE DATOS DEL SISTEMA CEDRUX. ....	57
FIGURA 24 TEST DE WILCOXON SOBRE VARIABLES DEPENDIENTES ....	62
FIGURA 25 ANÁLISIS ESTADÍSTICOS DE LAS VARIABLES DEPENDIENTES.....	62
FIGURA 26 RANGO Y DIFERENCIAS ENTRE LAS MUESTRAS DE LAS VARIABLES INDEPENDIENTES .....	63
FIGURA 27 ESTADÍSTICAS DE LOS INDICADORES DE LAS VARIABLES INDEPENDIENTES.....	64

## ÍNDICE DE TABLAS

TABLA 2.1 TABLA CON CONTROL DE CONCURRENCIA. ....	32
TABLA 2.2 TABLA CONTENEDORA DE LAS SECUENCIAS DE TABLAS A REPLICAR. ....	47
TABLA 2.3 REQUERIMIENTOS TECNOLÓGICOS. ....	48
TABLA 3.1 APLICACIÓN DE LAS NORMAS DE DISEÑO. ....	55
TABLA 3.2 REUTILIZACIÓN DEL SGE-UCID. ....	58
TABLA 3.3 VARIABLE INTEROPERABILIDAD ENTRE LOS MÓDULOS Y LOS DATOS. ....	58
TABLA 3.4 CUBRIMIENTO DE NECESIDADES DE GESTIÓN DE DATOS. ....	59
TABLA 3.5 ESTRUCTURA ORGANIZACIONAL POR ESQUEMAS. ....	59
TABLA 3.6 NIVEL DE SEGURIDAD. ....	60
TABLA 3.7 APLICACIÓN DE LAS NORMAS DE DISEÑO. ....	60
TABLA 3.8 FORMALIZACIÓN DE LA SOLUCIÓN. ....	60
TABLA 3.9 CAPACIDAD DE INSTALACIÓN MODULAR. ....	61
TABLA 3.10 SIGLAS DE LAS VARIABLES DEPENDIENTES: ....	61
TABLA 3.11 SIGLAS DE LAS VARIABLES INDEPENDIENTES: ....	63

## INTRODUCCIÓN

Un sistema de Gestión de Recursos Empresariales o ERP<sup>1</sup>, por sus siglas en inglés, es una solución informática integral que está formada por unidades interdependientes llamadas Módulos. Los fundamentales son los denominados Módulos Básicos, de adquisición obligatoria, y sobre los cuales se agregan los Módulos Opcionales, los cuales no son obligatorios y se agregan para incorporar nuevas funcionalidades al ERP. También existen los Módulos Verticales, que no son más que módulos opcionales para resolver específicamente las funciones y necesidades de un sector de negocio determinado.

Aunque existe una gran diversidad y variedad de sistemas ERP, se puede decir que todos tienen puntos en comunes como son (Vera, 2006):

- Elevado número de funcionalidades: Los sistemas ERP poseen un gran número de funcionalidades lo que le permite abarcar la casi totalidad de los procesos de negocio de la mayoría de las empresas.
- Grado de abstracción: Los ERP tienen la capacidad de manejar cualquier tipo de circunstancia que pueda tener en la empresa y soporta diversos grupos empresariales sin conexión entre ellos.
- Adaptabilidad: Son sistemas capaces de adaptarse a cualquier empresa, independiente al sector al que pertenezcan y de las particularidades de los procesos de negocio.
- Modularidad: Los ERP están conformados por un número específico de módulos, independientes entre sí, pero que a su vez están relacionados, lo que permite una gran adaptabilidad de las empresas de acuerdo a su tamaño y disponibilidad de recursos.
- Universalidad: Los ERP pueden ser utilizados por cualquier organización, sin embargo muchos de los proveedores explican que existen ERP para algunas industrias específicas.
- Integrales: Los ERP permiten controlar los diferentes procesos de la empresa, entendiendo que todos los departamentos se relacionan entre sí y que el resultado de un proceso es punto de inicio del siguiente. Si la empresa no usa un ERP, necesita tener varios programas que controlen todos los procesos mencionados, con la desventaja de que al no estar integrados, la información

---

<sup>1</sup> Enterprise Resource Planning (Planificación de Recursos Empresariales).

se duplica y crece el margen de contaminación en la información. (Codorniú, 2010)

Entre las ventajas que presentan estos sistemas se encuentran (González, 2010):

- Permiten trabajar con la información en tiempo real y de manera fiable.
- Se pueden personalizar y por lo tanto optimizar según las necesidades del destinatario. Son sistemas que garantizan la integridad de los datos que se ingresan, ya que estos sólo se introducen una vez.
- Son sistemas seguros lo que conlleva que se puede proteger la información tanto contra ataques internos y externos, como por ejemplo delitos de malversación.
- Permiten poder acceder al mismo desde cualquier lugar o cualquier dispositivo.
- Se consigue automatizar los procesos que se manejan bajo políticas preestablecidas, minimizando la posibilidad de errores humanos o información incorrecta e inconsistente.
- Se pueden unificar los recursos TI <sup>2</sup>de la empresa. Utilizando un solo programa para la gestión integral unificada de la misma y obteniendo una reducción de gastos en TI debido a la mayor facilidad del mantenimiento de las aplicaciones y sistemas.

También presentan un conjunto de desventajas como son:

- Se requiere una fuerte inversión para la educación continuada del personal de la empresa dado que estos sistemas tienen un manejo particular y es necesario una formación previa antes de usarse.
- La instalación de un ERP es muy costosa, dado que muy a menudo estos sistemas requieren el uso de licencias y su posterior renovación.
- Los ERP sufren el problema del “eslabón más débil” que viene a significar que un fallo en un departamento o en uno de los empleados de la empresa puede afectar al funcionamiento del resto del sistema.
- Las diferentes modificaciones en el sistema una vez que este esté establecido suelen ser muy costosas y elevadas.
- Muy a menudo se pueden encontrar problemas de compatibilidad con otros sistemas que usa la empresa.

---

<sup>2</sup> Tecnología de la Información.

## **Entorno nacional**

Debido a la gran utilidad e importancia de los Sistemas de Gestión Empresarial o ERP, nuestro país no se encuentra ajeno al uso de los mismos. La dirección del país, como parte del fortalecimiento de la gestión de las entidades y la informatización de la sociedad cubana, ha planteado la necesidad de informatizar los procesos de gestión de las entidades presupuestadas y empresariales a escala nacional utilizando plataformas confiables y eficientes. Muchas de las entidades o empresas cubanas utilizan algunos de estos sistemas para llevar a cabo, de manera automatizada, los procesos que son transversales para cualquier empresa o entidad. Los sistemas de este tipo que hoy se utilizan en nuestro país por parte de estas empresas son, en su gran mayoría propietarios, como es el caso de SAP ERP, lo cual implica el pago de grandes sumas de dinero por concepto de licencias. Otros de los sistemas, aunque libres, no es posible implantarlos en todas las empresas, debido a que los mismos se encuentran desarrollados para actividades específicas de la economía y enfocadas para las pequeñas y medianas empresas. Otro punto que lleva al traste de la implantación de alguno de estos sistemas es que nuestra economía socialista tiene características muy particulares y peculiares como es el caso de la dualidad monetaria, por lo que la mayoría de estos sistemas no se ajustan a las características de la misma. Como parte de lo anteriormente expuesto se puede concluir que en la actualidad no existe en Cuba un sistema informático integral de gestión que cumpla con la totalidad de los requerimientos de funcionalidad, interoperabilidad y seguridad que espera el gobierno cubano de una solución de este tipo, de manera que pueda ser utilizada como herramienta para potenciar el cumplimiento de las funciones de las entidades a todos los niveles con un máximo de racionalidad y control de los recursos financieros, materiales y humanos. (Ríos, 2008)

Para anular dicho inconveniente se ha decidido, por parte de la alta dirección de nuestro país, el desarrollo de un sistema de este tipo de producción nacional, que se ajuste totalmente a las necesidades de nuestra economía. Muchos han sido los intentos por lograr alcanzar esta meta pero aún no se ha podido concretar ninguno de ellos. Es el caso del inicio del ERP en el año 2005, donde, la Universidad de las Ciencias Informáticas (UCI) de conjunto con DESOFT pretendieron realizar en serio un sistema de gestión, pero la falta de compromiso de las entidades encargadas de brindar la información técnica y teórica que se requiere para el desarrollo de estos sistemas, así como la no incorporación de los funcionales necesarios dentro del equipo de trabajo y la no concreción de las propuestas DESOFT-UCI para la creación de un grupo de trabajo que aporte definiciones concretas dieron al traste con el avance y desarrollo del mismo. También existen otros casos como es el de ERP Cubalse,

donde se trató de aunar esfuerzos entre la UCI y Cubase, pero el mismo fracasó debido a que dicha entidad no tenía como prioridad primordial en ese momento en desarrollo de un ERP si no de un sistema de gestión de inventario aunque bien se puede destacar que en materia de arquitectura ya tenían un adelanto de lo que podría ser luego la base para el desarrollo del ERP. (Brito, 2011)

La mayoría de estos sistemas que se han desarrollado en nuestro país no cubren con las funcionalidades y características que se han definido lo formen, además de que se encuentran destinados a sectores específicos de nuestra sociedad. Un ejemplo de estos sistemas es el Versat-Sarasola, el cual ha sido desarrollado por un grupo de trabajo de la Provincia de Villa Clara perteneciente al Ministerio del Azúcar para este sector particular de nuestra economía. El desarrollo de este sistema ha sido sobre plataformas propietarias, lo cual no se encuentra a tono con la política de nuestro país de uso de software libre que han sido adoptadas en nuestro país. Además dicho sistema no cubre el total de funcionalidades básicas que han sido identificadas por los funcionales como es el caso de la multientidad<sup>3</sup>. La solución propuesta por este sistema se ha tratado de acomodar a otras empresas o entidades que pertenecen a otros sectores y en mucho de los casos se han obtenido resultados muy discretos de satisfacción o no ha podido ser posible (Arza, 2006). Otro de los casos ha sido el desarrollo llevado a cabo por los profesionales de la informática de la Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa (UCID), los cuales solamente han podido obtener resultados muy discretos e insatisfactorios que han sido implantados en algunas de las entidades pertenecientes a este ministerio. (Brito, 2011)

Por la complejidad y tamaño de estas soluciones se requiere la disponibilidad de un grupo importante de recursos materiales, infraestructura, capital humano y organización para su desarrollo. Debido a estas particularidades la dirección del país ha dado a la UCI<sup>4</sup> la tarea de llevar a cabo semejante desafío, ya que la misma cuenta con la infraestructura tecnológica y el capital humano necesario para llevar a cabo el desarrollo de este sistema de gestión empresarial. El desarrollo del mismo se realizará de conjunto con profesionales del UCID de la rama de la informática. También se contará con el asesoramiento de las entidades rectoras de cada una de las áreas de las actividades económicas.

---

<sup>3</sup> Convivencia, en una misma base de datos, de la información de varias entidades, de manera tal que cada una de ellas solo tenga acceso a la información que le corresponde sin violar ni alterar la información restante.

<sup>4</sup> UCI: Universidad de las Ciencias Informáticas.

Para la realización de dicha solución informática es necesario realizar, como parte del proceso de desarrollo de software, las definiciones arquitectónicas que soportarán toda la implementación de las funcionalidades del mismo. Uno de los elementos que componen estas definiciones se encuentra la Arquitectura de Datos, la cual recoge los conceptos y estándares que se necesitarán para llevar a cabo del desarrollo de cualquier sistema. Hasta el momento en el país no se encuentra definido una infraestructura de manipulación de datos para Sistemas de Gestión Empresarial lo suficientemente genérico y estándar que permita su uso en el desarrollo de este tipo de sistemas, la cual guíe el desarrollo y definición de la vista de datos, provocando problemas en la definición, duplicidad e incongruencia de la información que se almacena.

## Diseño teórico

### Problema científico

El estado actual de la infraestructura para la manipulación de datos en los proyectos del programa ERP está afectando el cubrimiento de las necesidades gestión de datos y la interoperabilidad entre los módulos y datos.

### Objeto de estudio

Arquitectura de Datos.

### Objetivo general

Definir la Arquitectura de Datos del sistema de Gestión Empresarial que permita cubrir las necesidades gestión de datos y la interoperabilidad entre los módulos y datos.

### Objetivos específicos

- Realizar un estudio detallado sobre los principales conceptos de Arquitectura de Datos existentes.
- Realizar un análisis de los Sistemas de Gestión Empresarial más representativos existentes así como los desarrollados en el país.
- Definir una propuesta metodológica de Arquitectura de Datos para el ERP cubano.
- Validar la propuesta de Arquitectura de Datos definida a partir de su implantación en el sistema CEDRUX.

### Campo de acción

La Arquitectura de Datos de sistemas de Gestión Empresarial.

### Hipótesis

Si se realiza una definición de Arquitectura de Datos para los proyectos del programa ERP se logrará el cubrimiento de la necesidad de gestión de datos, así como se garantizará en dichos proyectos la interoperabilidad entre los módulos y los datos.

### Variables independientes

Variable	Dimensión	Indicador
Arquitectura de Datos	Diseño Estructural	Estructura organizacional por esquemas.
	Seguridad	Nivel de seguridad.
	Normas de diseño	Grado de aplicación de las normas de diseño.
	Formalización	Grado de formalización de la solución.
	Portabilidad	Capacidad de Instalación de forma modular.

### Variables dependientes

Variable	Dimensión	Indicador
Interoperabilidad entre los módulos y los datos	Coexistencia de datos	Aplicación de la Multientidad.
	Integridad	Intercambio de datos correctamente.
Cubrimiento de necesidades de gestión de datos	Complejidad	Grado de completamiento de los datos.

## **Muestreo**

Para este trabajo se tomó una población a todos los proyectos que actualmente se desarrollan en el Centro de Informatización para la Gestión de Entidades (CEIGE), el cual suma un número de 17 proyectos, de estos solamente se probará en un total de 13, los cuales representan un 76,46 % y en estos momentos, se encuentran en condiciones de realizar una evaluación de las variables tanto dependientes como independientes definidas en este trabajo.

## **Diseño de investigación**

El diseño de la investigación será preexperimental donde se analizarán las muestras antes de aplicar la Arquitectura de Datos y finalmente se desarrollarán mediciones de los proyectos después de aplicar la arquitectura.

La propuesta será validada mediante:

- Método de Evaluación de Arquitectura QUASAR.
- Test Wilcoxon (Comparación de dos muestras relacionadas).

## **Aporte práctico de la investigación**

Como resultado de esta investigación se obtuvo la definición de Arquitectura de datos para Sistemas de Gestión empresarial.

## **Métodos y técnicas a utilizar**

Los métodos teóricos a utilizar para estudiar las características del objeto de investigación son:

**Histórico lógico:** En la investigación se realiza un estudio del estado del arte con el fin de analizar la evolución que han tenido los diferentes temas asociados a la problemática (modelos, metodologías y otros elementos relacionados con la arquitectura y componentes de sistemas) y lograr un mayor entendimiento de los mismos, establecer las conexiones y dependencias fundamentales entre ellos.

**Hipotético deductivo:** Se sigue este método para dar solución al problema planteado, definiéndose una hipótesis a partir de la cual, siguiendo reglas lógicas de deducción, se llega a nuevos conocimientos y predicciones que posteriormente son sometidos a verificaciones empíricas.

**Sistémico:** El uso de este método se evidencia al tratar el problema como un todo único, estudiando la dinámica de todos los componentes que se determinan en cada uno de los elementos arquitectónicos en los proyectos, así como la relación que existe entre ellos para determinar sus dependencias e integración como sistema. Ello jugará un papel fundamental para garantizar la calidad del trabajo.

Los métodos empíricos utilizados para estudiar las características de los fenómenos originados o que involucran al objeto son:

**Observación:** Se hará uso de la misma durante varios momentos de la investigación, pues permite conocer de manera directa el comportamiento del objeto de estudio y como esta permite investigar el fenómeno en su manifestación externa, sin llegar a la esencia del mismo.

**Medición:** Se sigue este método con el objetivo de obtener información numérica acerca de una propiedad o cualidad del objeto, donde se comparan magnitudes medibles y conocidas. Se hará uso de este método en los resultados que brinden las distintas validaciones a la que se sometan los resultados de la investigación.

### **Estructura de la tesis**

Este documento quedará estructurado como sigue:

**Capítulo 1:** En este capítulo se expondrán los principales conceptos relativos a la arquitectura de datos. Se expondrá un estudio del estado del arte de los Sistemas de Gestión Empresarial más representativos en el mercado, así como su tratamiento a la arquitectura de datos de los mismos. También se brindará un estudio de los sistemas de este tipo que hayan sido desarrollados en nuestro país y el tratamiento que realizan los mismos a la arquitectura de datos.

**Capítulo 2:** Se expresará de manera detallada una propuesta de definición de arquitectura de datos teniendo en cuenta todos los conceptos y propuestas de soluciones necesarias para lograr que dicha definición sea todo lo genérico que se necesita para dar solución al problema planteado.

**Capítulo 3:** En este capítulo se realizará la validación de la propuesta realizada en el capítulo 2 utilizando los métodos seleccionados, los cuales permitirán arribar a conclusiones sobre la efectividad de esta definición que se propone utilizando las variables definidas para dicha validación.

# **1 CAPÍTULO 1 MARCO TEÓRICO**

## **1.1 Introducción**

En este capítulo se abordarán los principales conceptos asociados a temas de arquitectura haciendo énfasis en los referentes a la Arquitectura de datos: definiciones, principales corrientes, etc. También se realizará un análisis de los principales sistemas de gestión empresarial así como los ERP más importantes tanto nacionales como internacionales, lo cual contribuirá a la adquisición de elementos importantes para las definiciones de la propuesta metodológica que se pretende realizar.

## **1.2 Arquitectura de Software**

Actualmente en la literatura es posible encontrar numerosas definiciones del término Arquitectura de Software, cada una con planteamientos diversos.

Uno de los conceptos que se maneja es el de Kazman, el cual plantea que “la arquitectura de software de un programa o sistema de computación es la(s) estructura(s) del sistema que comprende los componentes del software, las propiedades visibles de esos componentes y las relaciones entre ellos.” (Kazman, y otros, 2001)

Otra definición es la que hace el Instituto de Ingenieros Electricistas y Electrónicos (del inglés, IEEE) la cual refiera la arquitectura de software como: “la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”. (Fernández, 2011)

Teniendo en cuenta los objetivos que persigue este trabajo, se decide asumir la siguiente definición de arquitectura de software: “La arquitectura de software es la representación de alto nivel de abstracción de un sistema, está constituida por las partes del mismo, el mecanismo de integración e interrelación, restricciones y configuraciones a entornos de solución, los principios del diseño y evolución así como el ambiente tecnológico de implantación.” (Ochoa, 2011)

## **1.3 Arquitectura de datos**

La arquitectura de datos dentro de cualquier proyecto de desarrollo de software es una de las áreas claves del mismo. La misma define como se almacenarán, manejarán y usarán los datos en un sistema. También establece pautas para las operaciones con los datos para poder hacer posible predecir, modelar, guiar y controlar el flujo de datos en un sistema. Esto es sobre todo muy importante cuando en los sistemas, el desarrollo de los componentes es llevado a cabo por diferentes desarrolladores. (Gálvez, y otros, 2010) (Jorrín, 2011)

Una de las definiciones más importantes sobre este tema es la realizada por el SEI en la norma CMU/SEI-2001-TR-018 en la cual se refiere a la Arquitectura de Datos de la siguiente manera: (SEI, y otros, 2001)

La Arquitectura de Datos define cómo serán almacenados, administrados y utilizados los datos en un sistema. En particular describe:

- cómo los datos se almacenan de manera persistente.
- cómo los componentes y los procesos referencian y manipulan los datos.
- cómo sistemas externos o legados van a acceder a los datos.
- interfaces de administración de datos para los sistemas externos o legados.
- implementación de las operaciones de datos comunes.

También explica que la arquitectura de datos se describe en términos de una colección de patrones arquitectónicos genéricos que definen y limitan el manejo de datos. Cada uno de estos patrones arquitectónicos muestra las operaciones de datos comunes y cómo estas se realizan en el sistema de suministro de destino. Sin esa orientación, las operaciones de datos comunes pueden ser implementadas de manera diferente, por lo que es imposible de predecir, modelo, calibre, o controlar el flujo de datos en el sistema.

#### **1.4 Sistemas de Gestión Empresarial**

La creación de aplicaciones empresariales y productos de software en general, es una propuesta difícil, incluso para el más simple de los productos. Eso es porque con el fin de tener éxito, las aplicaciones empresariales tienen que anticipar las diferentes necesidades del cliente y permitir la configuración de resolver sus problemas específicos.

##### **1.4.1 Sistemas de Gestión Empresarial Propietarios**

###### **1.4.1.1 SAP-ERP**

SAP-ERP es un sistema modular en el cual los mismos se encuentran estrechamente relacionados a través de paquetes de interoperabilidad. Esta aplicación se encuentra desarrollada sobre arquitectura orientada a servicios (SOA) que le permite crear nuevos servicios, procesos y productos con más facilidad y un coste inferior que las arquitecturas tradicionales. (SAP-España, 2011)

Este ERP utiliza como plataforma tecnológica a SAP NetWeaver, la cual no es más que la base tecnológica que les permite a los usuarios administrar, integrar y también añadir procesos y aplicaciones empresariales. Con los componentes, herramientas y aplicaciones de SAP NetWeaver puede coordinar procesos empresariales, crear aplicaciones y desplegar soluciones. (SAP-España, 2011)

Los elementos por los que está compuesta esta plataforma son SAP NetWeaver Master Data Management, SAP NetWeaver Application Server, SAP NetWeaver Business Intelligence, SAP NetWeaver Composite Environment, SAP NetWeaver Enterprise Portal, SAP NetWeaver Mobile y SAP NetWeaver Process Integration. (SAP-España, 2011)

NetWeaver Master Data Management (SAP MDM) es el encargado de centralizar y coleccionar todos los datos que sean manejados por las aplicaciones o módulos del sistema. Para ello tiene como funcionalidades la eliminación de datos duplicados, no permitir la inconsistencia entre datos, la consolidación, el sincronizar y distribuir los datos para todos los sistemas que estén integrados en SAP. (SAP-España, 2011)

Para el almacenamiento de los datos, esta plataforma, una vez que los datos pasaron por el proceso de depuración, los envía al servidor de datos maestros. El mismo utiliza tres capas para el almacenamiento y manipulación: (SAP-España, 2011)

- La capa de objeto es un depósito flexible de los datos maestros. Por ejemplo, si diez registros de clientes existen en todos los sistemas distribuidos, la capa de objeto tiene que tener un lugar para almacenar los campos de todos ellos.
- El nivel de servicio es como una caja de herramientas para la manipulación y gestión de datos maestros. Herramientas para crear, modificar y supervisar el estado de los objetos, ejecutar consultas, definición y ejecución de flujos de trabajo y la colaboración sobre la limpieza de datos, todos los existentes para ayudar a SAP MDM hacer al mundo seguro para la limpieza de datos.
- La capa de aprovisionamiento gestiona cómo se distribuyen los datos mediante el uso de SAP Exchange Infrastructure. Esta capa realiza funciones tales como permitir que los sistemas distribuidos se suscriban a ciertos datos en el repositorio central de manera que se pueda manipular cualquier cambio que fluyen a ellos.

SAP MDM está compuesto por: (SAP-España, 2011)

- Consolidación de contenido: Luego de realizada la limpieza y correspondencia de los datos por todo el sistema los mismos son cargados en SAP MDM. Los datos son sometidos a una variedad de algoritmos de correspondencia para la detección de los datos duplicados. Luego de que los datos son detectados es creada una tabla de mapeo para que los duplicados pueden ser identificados más tarde durante reportes, análisis u otro procesamiento.
- Armonización de los datos maestros: Una vez los datos son depurados se realiza los datos son creados centralmente y distribuidos a los distintos

sistemas. Los datos pueden variarse localmente en un sistema distribuido y luego el cambio puede ser automáticamente distribuido para un repositorio central o para otros sistemas. La calidad de datos puede ser comprobada a través de los flujos de datos del sistema. Los datos maestros recibidos de un repositorio central pueden estar extendidos en el repositorio local. Un conjunto de pasos de flujo de trabajo pueden aplicar a cambios y actualizaciones que requieren aprobación para distintos tipos de cambios.

- Centralización de los datos maestros: Una vez depurados todos los datos y distribuidos el sistema almacena los mismos en un repositorio central inhabilitando así la modificación de los mismos, es decir, si se quiere insertar la misma información el sistema no lo permite.

SAP es un sistema propietario, que gestiona y almacena datos de:

**Finanzas:** maneja todas las transacciones contables del sistema, permitiendo a los usuarios enfocarse en los análisis contables y financieros. Dentro de los datos que maneja se encuentran los de contabilidad multimonedada, contabilidad de inventarios, presupuesto, catálogo de cuentas personalizado y estructurado con un máximo de 5 niveles, contabilidad por centros de costo, entre otros. (Tavira, 2010)

**Compras y proveedores:** este módulo manipula los datos necesarios para permitir el control de las operaciones de compra que son realizadas así como ayuda a la gestión de la información de la relación de los proveedores. Entre los datos que maneja se encuentran los de orden de compra, entrada de mercancía, factura de proveedor, nota de crédito de proveedor, las fechas de entrega por partidas de compras, contabilización de facturas y notas de crédito, abastecimiento de proveedores teniendo en cuenta la multimonedada, etc. (Tavira, 2010)

**Control de Gestión (Controlling):** partiendo de los datos de la contabilidad financiera, la contabilidad analítica muestra los ingresos, gastos e inversiones. (Carnicero, y otros, 2004) Para ello es necesario conocer la contabilidad por centros de coste, la contabilidad presupuestaria, el control de costes del producto, las órdenes internas y la relación de costes basados en actividades. (Juaristi, 2007)

**Logística:** engloba la gestión de datos de todo el ciclo de vida de los productos de una empresa, desde la compra y almacenaje de materia prima, pasando por la fabricación del producto hasta su venta y distribución. También contiene toda la gestión de datos de cadena de suministro y bienes, desde su compra hasta su distribución y facturación. Dicho módulo tiene integración con Capital Humano, Finanzas y Controlling. (Juaristi, 2007)

**Ventas-Clientes:** este módulo tiene la responsabilidad de la facturación dentro del sistema. Para ello necesita manejar datos como la cotización, las órdenes de venta, la remisión, las facturas de clientes, las notas de crédito, la solicitud de anticipo y la factura de anticipo. (Tavira, 2010)

**Inventario:** gestiona todos los productos y servicios desde el dato maestro, precios, costos, stock, reportes de inventario, artículos alternativos, gestión de almacenes, etc. (Tavira, 2010)

**Gestión de Bancos:** controla la relación de los datos de los pagos que se procesan por cuentas por cobrar y cuentas por pagar. También manipula la gestión de datos de los medios de pago hasta los reportes de antigüedad de saldos y conciliaciones bancarias. Para ello es necesario conocer los pagos recibidos y efectuados, conocer y controlar las cuentas por cobrar y las cuentas por pagar, gestionar los datos de los cheques, etc. (Tavira, 2010)

**Producción:** su responsabilidad es permitir gestionar las listas de materiales para así gestionar las materias primas y componentes involucrados en la manufactura de productos terminados y semiterminados. Por eso este módulo entre los datos que maneja se encuentra el listado de materiales, variaciones en los cierres de las órdenes de producción, asignación a las órdenes de producción los materiales e insumos y la mano de obra, transacciones por órdenes de producción, los datos de los planes de producción teniendo en cuenta las necesidades. (Tavira, 2010)

**Recursos humanos:** con este módulo el sistema permite gestionar todos los datos de los trabajadores de las entidades. Para ello es necesaria la administración de los datos que permitirán realizar la nómina de cada uno de los trabajadores que se almacenarán en el sistema. Este módulo en particular necesita gestionar los datos maestros de personal con los cuales se conocerán toda la información referente a los mismos, entre los datos que maneja se encuentran datos personales, medidas de los empleados, acuerdos contractuales y empresariales y cualificaciones de empleado. También almacena los gastos de viaje, desarrollo de personal, gestión de la formación, selección de personal y gestión de tiempos. (Juaristi, 2007)

Si bien SAP es uno de los líderes mundiales en este tipo de sistemas, también es importante mencionar que el mismo no cuenta entre sus funcionalidades la gestión de datos de la multientidad, es decir, que este tipo de sistemas está diseñado para su uso en una empresa en específico, por lo que no permite, desde una empresa padre, conocer los datos de sus sucursales o empresas hijas, requisito este primordial hoy en el sistema de gestión empresarial que se implante en nuestro país. Otro de los problemas que representa SAP para nuestras empresas es que al ser un sistema

privativo, no es posible conocer o acceder a su código fuente, por lo que no se puede conocer como el mismo maneja la seguridad de los datos que en él se almacenan, la existencia o no de puertas traseras para el acceso a la información a través de software malignos por personas mal intencionadas. Todo esto hace de este sistema inadecuado para la implantación en nuestro país debido a lo delicado de la información sobre nuestra economía que se pretende manejar.

#### **1.4.1.2 Oracle JD Edwards**

Oracle JD Edwards es una suite de aplicaciones integradas de software de planificación de recursos empresariales que combina el valor del negocio, la tecnología basada en estándares, y la experiencia en la industria en una solución de negocio. (Oracle, 2011)

Este sistema, al igual que la mayoría de los sistemas ERP, se encuentra desarrollado sobre una arquitectura cliente/servidor. El mismo cuenta con una base de datos única y aplicaciones previamente integradas, esta interconexión total entre módulos se logra gracias a una base de datos integrada. (CEA, 2006)

Para el manejo de los datos este sistema cuenta con el Diccionario de datos el cual es un repositorio de información de datos. En los términos simples el diccionario de datos contiene el metadato de Oracle. El diccionario de datos realiza las siguientes funciones: (Jacot, y otros, 2009)

- Define los ítems de cada dato en el sistema incluyendo definiciones de la columna e información del texto.
- Determina cómo aparecen los ítems de los datos en informes y formas interactivas.
- Provee envío de mensajes de error.
- Contiene descripciones de las columnas y de filas. Estos pueden ser utilizados como valores predeterminados al desarrollarse en EnterpriseOne.
- Usado en conjunción con los valores de Código Creado por el Usuario (UDC), valida entrada de datos del campo (cuando usted introduce información en un campo y este se pone rojo indicando un error, el diccionario de datos fue fundamental en determinar la validez de los datos introducidos).
- Provee mensajes de ayuda de campo de especialización. Estos son mostrados cuando usted presiona a F1 en un campo.
- El sistema no almacena el decimal lugar en el archivo de la base de datos y usa el diccionario de datos para determinar el número de decimales lugares exhibidos en cálculos e interfaces diversas a todo lo largo del sistema.

La información que maneja el sistema se encuentra distribuida en los siguientes módulos:

**Administración financiera:** es uno de los módulos dentro del paquete de soluciones. El mismo es el encargado del manejo de los datos de las cuentas por pagar, cuentas por cobrar, contabilidad general, de costos avanzados y ambientales, gestión de gastos, gestión financiera y de cumplimiento, así como la administración de activos fijos. (Oracle, 2011)

**Customer Relationship Management (CRM):** entre los datos que se gestionan en este módulo se encuentran la gestión de contratos, gestión de órdenes de ventas, gestión de precios avanzada, configurador base, variantes de productos, gestión multicanal, autoservicios, gestión de pedidos, gestión multicanal, autoservicios, gestión de ventas. (CEA, 2006)

**Administración de capital humano:** gestiona todos los datos de personal del sistema, además gestiona los recursos humanos, en donde se maneja toda la información acerca del personal tales como las competencias, salario, datos personales, etc. Proporciona datos sobre la nómina, el tiempo real que ha empleado cada trabajador en las jornadas laborales, entre otros. (Oracle, 2011)

**Compras:** trata los datos relativos a la gestión de almacenes, estando asociada al módulo de Gestión de la Cadena de Suministros, con las siguientes funcionalidades: gestión de inventarios, gestión de almacenes, gestión de materiales por órdenes de trabajo, gestión de mercancía, gestión de transporte, logística de inventario, entre otras. (García, y otros, 2010)

**Gestión de proyectos:** este trabaja con los datos de las funcionalidades gestión de constructoras, contratos y facturación de proyectos y costes de proyectos. (García, y otros, 2010)

**Ventas:** en el mismo se manejan los datos de precios complejos y presupuestos, así como configuraciones de productos y pedidos y la administración de ventas cruzadas, ventas posteriores a la compra inicial, gestión de pedidos, ajustes de precios y promociones de forma precisa para segmentos de mercados específicos. También permite configurar los datos de los productos, evaluar opciones de precios, proporcionar sugerencias de ventas cruzadas y ventas posteriores, comprobar fechas de entrega, calcular el flete y comprobar el crédito de un cliente, gestión de Ventas. (CEA, 2006)

Es un sistema dirigido fundamentalmente a las pequeñas y medianas empresas, y se especializa en sectores determinados de la economía como es la construcción, la gestión de proyectos entre otros. Al igual que SAP este sistema es propietario, lo que

nos imposibilita el acceso al código fuente y no nos proporciona total información sobre la seguridad de los datos en el sistema, tema que para nuestro país es de suma importancia por la información que con un sistema como este se pretende gestionar. Aunque tiene desarrollado la funcionalidad de multiusuario, con el mismo no se puede llevar a cabo la multiempresa, lo que, para nuestro país, es muy necesario.

### **Conclusiones parciales:**

Estos sistemas privativos, además de los elevados precios que tienen las licencias de los mismos y el uso de software propietarios para el desarrollo e implantación, carecen de funcionalidades imprescindibles para la propuesta de sistema del ERP cubano, ya que las mismas no cuentan con la multientidad ni con la funcionalidad de la multimoneda con las características específicas de la economía cubana. Ninguno de estos sistemas, como se ha expuesto anteriormente, muestra el código que soporta al mismo, por lo que no se tiene certeza de que es lo que tienen implementado y como se lleva a cabo la seguridad de los datos en el mismo, así como las posibles puertas traseras que pudieran estar implementada y que le brindaría a personas mal intencionadas el acceso a información delicada para la economía de nuestro país.

## **1.4.2 Sistemas de Gestión Empresarial Libre**

### **1.4.2.1 OpenBravo-ERP**

Openbravo ERP se desarrolla utilizando estándares abiertos, alrededor de una combinación única entre los marcos de desarrollo MVC<sup>5</sup> y MDD<sup>6</sup>, los cuales han sido exhaustivamente probados, y el motor WAD<sup>7</sup> de Openbravo que los ejecuta. (open bravo, 2007-2010)

El MVC lo utiliza para desacoplar la base de datos, los elementos de la interfaz de usuario, y la lógica de negocio. La separación de estos elementos en ficheros distintos resulta en un código más estructurado, facilitando así el desarrollo y mantenimiento del mismo. Por su parte el MDD supone un modelo que depende de los metadatos almacenados en un diccionario para modelar el comportamiento de la aplicación. Esto conlleva una reducción drástica en cuanto a codificación manual y número de errores se refiere, permitiendo que expertos de negocio con poca experiencia a nivel de codificación puedan configurar la aplicación para satisfacer las necesidades de cada empresa. El diccionario MDD de la aplicación almacena el metadato que describe cada elemento de la aplicación incluyendo el comportamiento del mismo. (open bravo, 2007-2010)

---

<sup>5</sup> Modelo Vista Controlador.

<sup>6</sup> Model Driven Development

<sup>7</sup> Wizard for Application Development

Este ERP está desarrollado sobre tecnologías estándares y modernas, pero sólidas y suficientemente probadas, para cumplir los requerimientos estrictos de rendimiento y escalabilidad de cualquier entorno empresarial: Java y Javascript, SQL y PL/SQL, XML y HTML. (open bravo, 2007-2010)

Existe dentro de la arquitectura de OpenBravo un componente clave denominado WAD que es un motor propio, que genera automáticamente el código binario de la aplicación a partir del diccionario MDD. Los ficheros generados por el WAD se generan conforme al estándar MVC.

Este sistema está compuesto de aplicaciones de terceros bien conocidas como Apache http Server y Tomcat, y una base de datos PostgreSQL u Oracle. (open bravo, 2007-2010)

OpenBravo incluye una extensa cobertura funcional que proporciona un sólido fundamento para la agilidad en el negocio. Se basa en un modelo con una base de datos única e integrada que cubre todas las áreas principales de un sistema de gestión completo del negocio, incluyendo la integración con la gestión de Punto de Venta (POS).

Las funcionalidades que tiene implementada se agrupan en 9 módulos, en los cuales se manejan los datos que el mismo almacena:

**Gestión de datos maestros:** es el encargado de organizar y centralizar los datos claves del sistema (productos, clientes, proveedores, etc.) para así facilitar que la información fluya con facilidad y rapidez entre todas las áreas implicadas en los diferentes procesos de negocio. (open bravo, 2007-2010)

**Gestión de aprovisionamiento:** se encarga del control de los datos de la cadena de suministros, es decir, control de tarifas, pedidos de compra, recepción de mercancías, registro y contabilización de facturas de proveedores, planificación de los aprovisionamientos, etc. Se encuentra totalmente integrado con Gestión Económico-Financiera y de Almacén. (open bravo, 2007-2010)

**Gestión de almacenes:** este módulo es el responsable de la gestión de datos de los almacenes y ubicaciones (multi-almacén), unidades de almacén, lotes, número de serie, bultos, etiquetas, entradas, salidas, movimientos entre almacenes, inventarios, valoración de existencias, transportes, etc. (open bravo, 2007-2010)

**Gestión de proyectos y servicios:** es el responsable de gestionar la rentabilidad de los proyectos o servicios que tenga la entidad. Para ello es necesario que almacene y maneje los datos de proyectos- tipos de proyectos, tarifas de proyectos-, fases, tareas, recursos, presupuestos, control de gastos asociados a un proyecto, gastos internos,

gastos facturables y facturación, compras asociadas, categorías salariales históricas asociadas a costes de proyecto, etc. (open bravo, 2007-2010)

**Gestión de la producción:** está concebido para la gestión de datos de la estructura productiva de cada entidad (secciones, centros de coste, máquinas), así como de los datos relevantes para la producción: planes de producción (secuencias de operaciones) y productos involucrados en las mismas. Entre los datos que maneja se encuentran centros de coste, centros de trabajo y máquinas, planificación de la producción (MRP), teniendo en cuenta, previsiones, pedidos de cliente, existencias, stock mínimo y órdenes de fabricación en curso. También gestionan los datos de las órdenes de fabricación, los productos de cada orden, incidencias de trabajo, tipos de utillajes y gestión de cada utillaje individual y mantenimiento preventivo y partes de mantenimiento. (open bravo, 2007-2010)

**Gestión comercial y CRM:** está diseñada con el objetivo de permitir la máxima flexibilidad y agilidad en la ejecución del proceso comercial de una entidad. Para ellos es necesario almacenar los datos de las zonas de ventas, los pedidos de venta. También los datos de la aplicación de tarifas: precios, descuentos y control de precio límite. Los datos de los documentos necesarios para el control de los pedidos, los productos de los almacenes, facturas, etc. Otros datos que son almacenados son los referentes a los clientes de las entidades así como la gestión centralizada de las listas de precios. (open bravo, 2007-2010)

**Finanzas y contabilidad:** este módulo es el encargado de gestionar los datos de las finanzas dentro del sistema. El mismo se nutre de la información que brindan los demás módulos del sistema de manera que éstos tienen un reflejo automático en la contabilidad general, en las cuentas a cobrar y en las cuentas a pagar en cuanto se producen. Se encuentra conformado por submódulos los cuales gestionan la siguiente información:

Contabilidad general: gestiona los datos de planes contables, ejercicios contables y gestión interanual, presupuestos, categorías de impuestos, rangos de impuestos, asientos manuales, asientos tipo, balance de sumas y saldos, libro mayor, cuenta de resultados y cuadros del plan general contable.

Cuentas a pagar y cuentas a cobrar: gestiona los datos de la gestión de efectos, las remesas (según cuadernos bancarios), edición de cajas (multi-caja), apuntes de caja de tipo de gasto, ingreso, diferencia, efecto, pedido, extractos bancarios, liquidaciones.

Activos fijos: gestiona los datos de grupos de activos, activos, con su precio de adquisición correspondiente y valoración contable. También la amortización lineal en porcentaje o temporal y los planes de amortización.

Este sistema permite el soporte para múltiples monedas, múltiples esquemas contables, haciendo que la transacción sea contabilizada según reglas distintas, esquemas contables varios, distintas monedas o incluso diferentes calendarios.

OpenBravo utiliza una plataforma modular, la cual permite ajustarse a las necesidades específicas del cliente e incluyen localizaciones, extensiones funcionales, conectores con otras aplicaciones, verticales, etc. (open bravo, 2007-2010)

Entre las funcionalidades con las que cuenta figuran (open bravo, 2007-2010):

- Workspaces y Widgets según rol de usuario
- Interfaz multi-tabla
- Edición de registros con tecnología de vanguardia
- Posibilidad de adjuntar documento, imágenes o ficheros a cualquier registro de la aplicación
- Información relacionada navegable (historial, documentos relacionados, etc.)
- Niveles de acceso configurables para los usuarios según rol
- Auditoría de cada transacción

Aunque Open Bravo es una solución se dice realizada en software libre y de código abierto, para nuestro país no toda la tecnología es del todo libre, como es el caso de Java, la cual contiene una cláusula que no permite la comercialización de sistemas realizados con la misma a nuestro país. Otro de los inconvenientes es que este sistema no permite gestionar un grupo de datos que son de vital importancia y muy particulares de la economía de nuestro país como es el caso de la contabilización con varias monedas debido a la dualidad monetaria.

#### **1.4.2.2 Compiere**

Compiere es uno de los pioneros de la industria del ERP. Es una aplicación cliente-servidor escrita enteramente en Java. Soporta el procesamiento de grandes volúmenes de información y una interfaz gráfica de usuario de alta performance. El servidor de aplicaciones está implementado en Java, con la tecnología J2EE<sup>8</sup>, utilizando la infraestructura del servidor de aplicaciones JBoss<sup>9</sup>. Este servidor puede estar corriendo de manera *stand alone* o en el mismo equipo que el servidor de la base de datos. Para la administración del servidor se utiliza JMX (Java Management Extensions). El acceso a la base de datos se realiza mediante el protocolo JDBC (Java Database Connectivity). (Compiere, 2009)

---

<sup>8</sup> Java 2, Enterprise Edition. Versión avanzada de la plataforma Java de Sun Microsystems, destinada al desarrollo de aplicaciones empresariales.

<sup>9</sup> JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java.

El programa de configuración empaqueta las librerías requeridas para la instalación de clientes y servidores de aplicaciones. Este enfoque elimina la necesidad de portar Compiere a otros motores de base de datos y permite que las nuevas versiones estén disponibles para otras plataformas simultáneamente. (Compiere, 2009)

Compiere contiene un Kit de Independencia de la Base de Datos. Los componentes primarios de este Kit son:

- Analizador que convierte los DML<sup>10</sup> y DDL<sup>11</sup> a la notación original de la base de datos.
- Administración/Interfase Class de la base de datos
- Configuración de la Database Class
- Ant<sup>12</sup> y scripts del Sistema Operativo para crear la librería y efectuar las tareas de la base de datos.

Uno de los elementos que componen este sistema es el Diccionario de Datos. El mismo se encuentra alojado en la capa de meta-datos y sabe como acceder a los datos y como se relaciona la información. Contiene definiciones de entidades de datos (tipos, validaciones, etc.), como se muestran (títulos sobre pantallas y reportes, ayudas, posición relativa con respecto a otros datos, etc.) y las reglas para mostrarlos. También se almacenan aquí las reglas de seguridad y acceso. Este diccionario es “activo”, significando con ello que es utilizado en tiempo de ejecución y es sensible al contexto. Es extensible por el usuario y puede incluir reglas e información especificada por el usuario. También permite a usuarios autorizados, agregar nuevas tablas, nuevas pantallas y datos adicionales sobre pantallas ya existentes en la aplicación. Toda esta información agregada está automáticamente disponible en los listados y reportes. (Compiere, 2009)

Entre los módulos con los que cuenta este sistema se encuentran:

**Cotización a ingresos:** este módulo gestiona los datos para la creación de cotizaciones, administración de órdenes de venta, facturación y recepción de dinero por cobranzas. (Compiere, 2009)

**Requisición a Pagos:** en este módulo se encarga de la administración de los datos necesarios para la creación de órdenes de compra, procesamiento de facturas de proveedores y pagos efectuados. (Compiere, 2009)

---

<sup>10</sup> Data-Manipulation-Language

<sup>11</sup> Data-Definition-Language

<sup>12</sup> Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (build). Es similar a Make pero desarrollado en lenguaje Java y requiere la plataforma Java.

**Administración de clientes:** almacena todos los datos referentes a clientes y posibles clientes para ello se administra la creación, distribución y seguimiento del cliente, proveedor y los pedidos generados internamente. (Compiere, 2009)

**Administración de socios:** liga diferentes clientes uno al otro, posibilitando manejar la distribución principal, pedidos de servicio, y gastos de marketing. Este módulo, al igual que el de administración de clientes, recoge todos los datos referentes a los socios de las entidades, así como les permite a los mismos tener acceso a la información que le corresponde y realizar facturas en el mismo. (Compiere, 2009)

**Administración de la Cadena de Suministro:** cubre todos los datos de administración de materiales, incluyendo recepciones, entregas, movimientos y administración y procesamiento de tomas de stock. También administra la información acerca del catálogo de productos del sistema en el cual se conoce la lista de los productos y servicios con la lista de materiales y sustitutos opcionales. (Compiere, 2009)

El sistema Compiere también permite realizar multientidad, multimoneda, proporcionando un sistema flexible y adaptable para las pequeñas y medianas empresas, que son a las cuales va dirigido fundamentalmente. (Compiere, 2009)

Como en el caso de Open Bravo, este sistema, que aunque utiliza software libres como tecnologías para el desarrollo, para nuestro país existe una cláusula que no permite el uso y comercialización de la tecnología Java. También este sistema, aunque soporta la multimoneda, no permite la contabilización con varias monedas, requisito particular de nuestro sistema económico debido a la dualidad monetaria.

#### **1.4.2.3 ADempiere**

ADempiere es un software de Código Abierto (Open Source) que ofrece la funcionalidad de ERP, CRM <sup>13</sup>y SCM<sup>14</sup>. Esta aplicación, surgida a partir de Compiere, es desarrollada y mantenida por una gran comunidad de usuarios. Ha sido diseñado como una sola aplicación ERP: todas las funcionalidades comparten una arquitectura común como así también las interfaces, reglas, filosofía y todas ellas están consistentemente integradas, alejándose del tipo de aplicaciones modulares que olvidan el concepto de "Proceso de Negocios". (Openbiz, 2011)

ADempiere heredó el diccionario de aplicación del proyecto Compiere. Esta arquitectura facilita la extensión del concepto de Diccionario de Datos dentro de la aplicación, lo cual a su vez, hace posible la administración de entidades, reglas de validación, como así también que el formato de pantalla y la lógica de despliegue sean

---

<sup>13</sup> Customer-Relations-Management

<sup>14</sup> Supply-Chain-Management

controlados dentro de la misma aplicación. Se utiliza una máquina de flujos de trabajo (Workflow Engine) basada en los estándares WFMC<sup>15</sup> y OMG<sup>16</sup> para proveer de la administración de procesos de negocios. Estas características permiten una rápida modificación de la aplicación a medida que las necesidades de un negocio van evolucionando. (Adempiere, 2011)

La información que gestiona este sistema se encuentra dividida en los siguientes módulos:

**Cotización a ingresos:** es el encargado de realizar la creación de cotizaciones, administración de órdenes de venta, facturación y recepción de dinero por concepto de pago. Para poder realizar estas funcionalidades el sistema debe administrar los datos de las cotizaciones, las órdenes de ventas, las facturas de los clientes y los pagos realizados por los mismos. (Openbiz, 2009)

**Requerimientos de pagos:** en el mismo se gestionan los datos de las órdenes de compras, las facturas de los proveedores y los pagos efectuados a los mismos. Para ellos el sistema debe llevar un control de los datos de los pedidos, de las órdenes de compras, la recepción de materiales, las facturas de compras, los pagos realizados y las conciliaciones bancarias. (Openbiz, 2009)

**Administración de relaciones con los clientes:** este módulo gestiona las mismas funcionalidades que los módulos CRM<sup>17</sup> de otros sistemas, por lo que en el mismo se almacenan todos los datos correspondientes de los clientes con los que cuentan las entidades así como los potenciales clientes. En el mismo se manejan los datos referentes a los clientes como ventas realizadas a los clientes, pedidos generados, reembolso de costos, etc. (Compiere, 2009)

**Administración de relaciones con socios:** permite conocer la distribución de los socios, así como los datos de los mismos, las facturas y pedidos realizados.

**Administración de cadena de suministro:** permite administrar los datos de materiales, incluyendo recepciones, entregas, movimientos y administración y procesamiento de stock. Para ello el sistema permite llevar el control de los depósitos, conocer las listas de materiales, así como administrar los datos de los mismos. Este módulo también gestiona todos los datos referentes a los activos de la empresa, así como el inventario de la misma. (Compiere, 2009)

---

<sup>15</sup> Workflow-Management-Coalition

<sup>16</sup> Object Management Group

<sup>17</sup> Customer Relationship Management (Gestión de Relación con los Clientes)

Este sistema, al igual que el Compiere, tiene implementado las funcionalidades de los multis como ellos mismos los llaman que no son más que multilinguaje, multientidad y multimonedas.

#### **Conclusiones parciales:**

Aunque estos sistemas son libres y se encuentran bajo licencia GPL y cuentan con funcionalidades como la multimonedas y la multientidad, las mismas no cumplen con las características que precisa nuestro país tengan estas funcionalidades como es en el caso de la multimonedas, debido a la existencia en nuestra economía de la dualidad monetaria, esto conlleva a que las contabilizaciones de nuestro sistema económico sea diferente a los sistemas de contabilización tradicionales. Otro punto que también se ha reflejado es que muchos de ellos están desarrollados con tecnología java, la cual, para nuestro país no es permitido su uso.

### **1.4.3 Sistemas de Gestión Empresarial Cubanos**

#### **1.4.3.1 Versat-Sarasola**

Versat es un ERP o Sistema Económico Integrado, conformado por 12 Subsistemas o Módulos. El mismo es un producto netamente cubano, adecuado a los principios del Control Interno y demás Regulaciones vigentes en el país.

Otro aspecto importante en la concepción del Sistema lo constituye la distribución y la localización de los datos (BD). Se pretende el uso de una BD centralizada a la cual podrán acceder simultáneamente varios usuarios del Sistema, vinculados a una misma actividad o a diferentes actividades. (Carralero, y otros, 2011)

Este sistema permite la integración con otros sistemas con los que pueda contar la entidad a partir del uso de las interfaces de comunicación que tiene implementado, lo cual posibilita usar este sistema como sistema base para la gestión contable y además que pueda intercambiar información con otro sistema que tenga la responsabilidad de gestionar otro tipo de información que maneje la entidad o empresa. Estas interfaces están implementadas haciendo uso de estándares internacionales.

Los módulos que lo componen son: (Carralero, y otros, 2011)

**Configuración:** Tiene como objetivo principal que se cumplan al instalar el sistema en su conjunto, los principios de control interno establecidos en las Normas y Principios Generales de Contabilidad, así como lograr la seguridad de los datos procesados y la información obtenida, a partir de la definición de los posibles usuarios y los accesos de estos a los diferentes módulos en cada una de las Unidades Contables de la entidad.

Está compuesto por tres funcionalidades importantes:

Nomencladores: es el encargado de la definición y almacenamiento de los datos de los nomencladores del sistema, sobre todo, de aquellos que son más importantes para la

Contabilidad General y los Costos, con el objetivo de registrar cada operación contable hasta el nivel de análisis imprescindible para la administración. Los nomencladores a definir son unidades contables, áreas de responsabilidad, entidades, trabajadores, custodios y almacenes. (Carralero, y otros, 2011)

Seguridad: se encarga de garantizar la seguridad del sistema a partir del almacenamiento en el mismo de datos importantes como los usuarios que trabajarán en el Sistema, quiénes serán los únicos con accesos a las diferentes opciones de trabajo que se les han autorizado al declararles los permisos de trabajo, asignar a los usuarios los permisos de trabajo en las Unidades Contables y dentro de estas a que Subsistemas tienen acceso, definirle a cada usuario que opciones de trabajo tiene dentro de cada subsistema. (Carralero, y otros, 2011)

Parámetros: Para facilitar la obtención de datos e informaciones en tiempo real de los usuarios y además facilitar el trabajo rápido y seguro en todas las bases de datos existentes de diferentes períodos. También permite gestionar los datos de los ejercicios económicos y los períodos correspondientes a este para la realización de operaciones simultáneas, a partir de la concepción de que cada módulo trabaja en tiempo real con independencia de los cierres de cada período y ejercicio. Permite el almacenamiento de la definición de los formatos de los codificadores que se utilizarán en el trabajo de los diferentes módulos, la estructura y cantidad de caracteres que tendrá cada nivel, está abierto para que el usuario lo haga de acuerdo con sus necesidades; así como el almacenamiento de las definiciones de monedas internacionalmente conocidas que serán utilizadas para efectuar revalorizaciones a los Estados Financieros y llevar el control de los recursos materiales y financieros en diferentes monedas. (Carralero, y otros, 2011)

**Contabilidad General**: Es el encargado de procesar, registrar y contabilizar todos los datos de los hechos económicos que se realicen en el sistema. Desde este módulo es posible acceder al de Costos y Procesos para realizar aperturas en el clasificador de Centros de Costos, definir las cuentas de gastos y sus aperturas, definir dentro de las cuentas de gastos, que aperturas registran sus gastos por subelementos; definir elementos y subelementos de gastos, definir partidas de gastos y definir la relación elementos – partidas de gastos. (Carralero, y otros, 2011)

Entre los datos que gestiona se encuentra el cierre de los períodos y el ejercicio económico y para ello necesita manejar los cierres de las cuentas nominales, cierres de la cuenta de resultados, teniendo presente las regulaciones que se establezcan para cada ejercicio; cierre de las diferentes subcuentas de las cuentas de proceso y

traspasar su saldo para las de inicio del año, cierre de otras cuentas, cierre del ejercicio y apertura de nuevo ejercicio. (Carralero, y otros, 2011)

**Costos y Procesos:** maneja los datos que permiten determinar los gastos incurridos en la producción de bienes materiales, la prestación de servicios, y en el resto de las actividades que se desarrollen en una entidad. (Carralero, y otros, 2011)

**Finanzas:** gestiona los datos sobre el movimiento y la disponibilidad de efectivo a partir del procesamiento de todos los documentos, tanto de obligaciones como de pagos. También gestiona la existencia, uso y disponibilidad de los instrumentos de pagos, vinculados a las cuentas bancarias que la empresa posee, las cuales puede conciliar y procesar de ellas, los estados de cuentas correspondientes. (Carralero, y otros, 2011)

Este módulo está compuesto por los siguientes submódulos:

- **Caja:** Permite controlar los datos de todas las operaciones que se realizan sobre el efectivo resumiéndose en el control de la disponibilidad de los fondos de caja, el cual brinda información de todos sus movimientos; control de los recibos de ingresos en poder del cajero u otro personal y el registro, control y contabilización de todos los documentos primarios que mueven los fondos definidos. (Carralero, y otros, 2011)
- **Banco:** controla los datos de las operaciones que se realizan sobre las diferentes cuentas bancarias definidas, permitiendo así definir las cuentas bancarias en diferentes monedas, definir los talonarios de los diferentes instrumentos de pagos por cada cuenta bancaria, incluyendo además las letras de cambio; gestionar los datos de los estados de cuentas a partir de la información que se recibe en el mismo y las operaciones registradas con anterioridad en la entidad; realizar conciliaciones de cada una de las cuentas bancarias, procesar los datos de los cobros y pagos automáticos que se muestran en los estados de cuentas, etc. (Carralero, y otros, 2011)
- **Créditos Bancarios:** permite controlar los datos de las operaciones relacionadas con la emisión y recepción de créditos bancarios. (Carralero, y otros, 2011)
- **Cobros y Pagos:** se procesan o muestran los datos de los documentos referidos a las operaciones que afectan directamente las diferentes cuentas que controla este subsistema y que tienen relación con las obligaciones de cobros y pagos. (Carralero, y otros, 2011)
- **Otras Operaciones Financieras:** está concebida para procesar en ella las obligaciones y derechos con el presupuesto del estado, así como la liquidación

correspondiente. Permite calcular flujos de cajas a partir de los movimientos de la cuenta bancaria y a la vez realizar pronósticos con seis variantes de cálculo. (Carralero, y otros, 2011)

**Inventario:** gestiona los datos de los recursos materiales. Para ello necesita que el sistema almacene los formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, la conceptualización de los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos, las existencias y movimientos en diferentes monedas, las existencias y movimientos por custodios y la configuración de la forma de controlar los precios de las existencias para aplicarlos posteriormente a las salidas que se realicen. (Carralero, y otros, 2011)

**Activos Fijos:** gestiona los datos de los Activos Fijos Tangibles e Intangibles a partir del almacenamiento de información correspondiente a los mismos como el Control por áreas y centros de costos, los diferentes movimientos y posibilita configurar la contabilización de los mismos, los inventarios a partir de diferentes selecciones, determinar las posibles diferencias y contabilizar las mismas y el almacenamiento de los activos en diferentes monedas. (Carralero, y otros, 2011)

**Contratación y Facturación:** gestiona los datos de todo el proceso actividades de la entidad con sus clientes y suministradores, así como facturación de las actividades y servicios que comercialicen. Tiene relación con otros módulos como Inventario, Activos Fijos, Finanzas, Control y pago de la caña, que permite el completamiento de los datos necesarios para el trabajo con clientes y suministradores. (Carralero, y otros, 2011)

#### **1.4.3.2 Sistema de Gestión Empresarial UCID**

Este proyecto surge debido a la necesidad que existía de garantizar un adecuado control y gestión de los recursos humanos, materiales y financieros para las unidades encargadas de la defensa en el país. Por las características de esta institución, se hacía necesario que se desarrollara con tecnologías propias implementadas por miembros de la misma, los cuales debían garantizar la independencia y soberanía tecnológica. (ERP-Cuba, 2008)

Hasta el año 2008, el desarrollo que este sistema había alcanzado se encontraba dividido en módulos independientes que no se relacionaban entre sí, es decir, no había integración entre los mismos, debido a que cada uno era considerado un proyecto independiente. Esto tuvo como resultado, que no existiera una arquitectura de datos definida y centralizada, por lo que constaban eran modelos de datos que eran creados en dependencia a la necesidad de solución que tuviese cada uno.

A continuación se mencionan los proyectos que hasta ese momento habían sido desarrollados. Los mismos son:

**Estructura y composición:** encargado de la gestión del organigrama de la empresa y plantilla, es decir, en el mismo se definen los datos de las empresas tales como nombres, estructuras de las mismas, así como la plantilla de los trabajadores por los cuales estará conformada la entidad.

**Recursos humanos:** responsable del registro de la mayoría de los datos relativos a las personas.

**Contabilidad y finanzas:** gestiona todos los datos referentes a la contabilización de las entidades.

**Logística:** maneja la información de los almacenes definidos en las entidades así como la gestión de inventario de cada una de las mismas.

Dentro de estas soluciones no se encuentran implementadas ni la multientidad ni la multimoneda, dos de las más importantes para estos tipos de sistemas, la primera para brindar la posibilidad de la coexistencia en una misma base de datos de varias entidades, la segunda es una necesidad para la contabilización en nuestro país.

Como se puede observar estos proyectos no cumplían con todos los parámetros necesarios para el uso de estos sistemas, es decir, las soluciones que brindan distan aún mucho del total de funcionalidades y gestión de datos que deben de poseer.

### **1.5 Factibilidad de solución**

Para la realización del estudio de factibilidad se tuvieron en cuenta cuatro de los gestores de datos más usados en la actualidad y que además son los que utilizan los Sistemas de Gestión Empresarial para el almacenamiento de la información: MySQL, Oracle, SQL Server y PostgreSQL.

Para este estudio se tuvieron en cuenta los precios a pagar para poder utilizar dichos gestores así como lo sistemas operativos sobre los cuales corren. Del estudio se obtuvieron los siguientes resultados:

- Todos los gestores, excepto SQL Server son multiplataforma, lo que permite el uso de lo mismos en varios tipos de sistemas operativos. En el caso de SQL Server solo funciona sobre plataforma Windows, debido a que el mismo es desarrollado por esta empresa limitando su uso a desarrollos realizados por ellos.
- El costo por uso de cada uno de estos gestores es muy elevado, sobre todo para países en vías de desarrollo como es Cuba. Estas licencias, en su gran mayoría, definen sus precios atendiendo a la cantidad de servidores en los cuales se instalará el sistema, o cantidad de usuarios que realizarán uso del

mismo o, en la mayoría de los casos, por cantidad de núcleos usados en cada uno de los servidores en los cuales estarán corriendo estas aplicaciones, además de los precios por soporte. Solamente el gestor PostgreSQL se encuentra bajo licencia BSD, la cual nos permite hacer uso del mismo sin tener que pagar por ello, además de que el soporte al mismo es realizado por una comunidad de programadores que tienen como patrón el concepto del software libre.

Para más detalles del estudio realizado ver [anexo 11](#).

### **Conclusiones parciales:**

Aunque estos sistemas se acercan un poco más a las características de nuestra economía, los mismos aún no cubren la totalidad de las necesidades identificadas por nuestro país para este tipo de sistema. En el caso del Versat-Sarasola carece de la multientidad, de la multimonedas y en el caso del sistema desarrollado por las UCID no se encuentran ni en un por ciento ínfimo de acercarse a cubrir parte de los datos, aunque es válido destacar que las soluciones que tienen implementadas hasta el momento son muy útiles e importantes para el desarrollo del futuro sistema que se pretende obtener como es el caso de la solución brindada por el módulo de estructura y composición.

### **1.6 Conclusiones**

Luego de realizar el estudio del estado del arte de arriba a las siguientes conclusiones:

- La Arquitectura de Datos es muy importante dentro del proceso de desarrollo de software, ya que la misma dicta las pautas a seguir para la definición del almacenamiento, manejo y uso de los datos en un sistema. El establecimiento de pautas para las operaciones con los datos para predecir, modelar, guiar y controlar el flujo de datos en un sistema.
- El estudio del trabajo con los datos de distintos sistemas de gestión empresarial, tanto privativos o propietarios, como los libres, permitieron conocer las distintas filosofías para el trabajo y manejo con los datos, así como la manera que interactúan los mismos con las aplicaciones. También de los mismos se pudo obtener las principales características que debían caracterizar la arquitectura de datos para este tipo de sistemas, debido a que permitió conocer los datos que han de manejarse así como la interoperabilidad entre la información que se almacena en los módulos que conforman este tipo de sistemas.

- Este estudio también arrojó como resultado el conocer que estos sistemas que hoy se desarrollan en el mundo no se adecúan a las necesidades de las entidades de nuestro país, en parte por las características de la economía socialistas del país, ya que la mayoría de estos sistemas se encuentran enfocados principalmente a pequeñas y medianas empresas y dentro de sus soluciones no contemplan el tema de la dualidad monetaria que nos caracteriza. Es importante no olvidar que, por la importancia de los datos que son almacenados en estos tipos de sistemas, la seguridad es un factor primordial, por lo que implantar algunos de estos software en las entidades de nuestro país no se adecúa a la política de soberanía tecnológica que se está llevando a cabo, además de que no se cuenta con un equipo de trabajo nacional que se dedique al desarrollo de estos sistemas por lo que cambios en los mismos en el futuro serían complejos de gestionar.
- La importancia de definir e implantar una Arquitectura de Datos para el sistema de gestión que se pretende implementar en la Universidad de las Ciencias Informáticas, que satisfaga las funcionalidades definidas para este sistema y que además se encuentre acorde con las características de desarrollo de dicho proyecto, teniendo en cuenta los equipos de desarrollo que en el trabajarán.

## **2 CAPÍTULO 2. PROPUESTA DE SOLUCIÓN**

### **2.1 Introducción**

En el presente capítulo se describe detalladamente la propuesta de solución de la arquitectura de datos para los sistemas de gestión. El término arquitectura de datos se ve reflejado en la investigación dentro de la vista de datos de un sistema. A lo largo del desarrollo de este capítulo se hará referencia a dos escenarios como se definen a continuación:

*Escenario 1 ó de Desarrollo:* este escenario es donde se diseñan y desarrolla la base de datos.

*Escenario 2 ó de Explotación:* este escenario es donde se realiza la explotación del producto final.

### **2.2 Restricciones de diseño**

Las restricciones que aquí se detallan son impuestas por la arquitectura general del sistema; restricciones respecto a temas tecnológicos que pueden o no incidir sobre la propuesta de solución. Algunas de ellas son:

- Montaje del sistema sobre plataformas libres, tanto a nivel de aplicación como a nivel de datos para lograr independencia tecnológica.
- El sistema debe desde el punto de vista funcional trabajar sobre escenarios:
  - Multimoneda
  - Dualidad de moneda
  - Multientidad

### **2.3 Elementos de la vista de datos**

La vista de datos de un sistema refleja elementos definidos por la arquitectura de datos que al representar estos elementos en forma de vista es denominada “vista arquitectónica” que como bien plantea Lazo “son la abstracción formalizada en prosa, lenguaje de modelado, gráfico o esquema informal, desde la que se describe las características arquitectónicas de un plano específico de la solución según el interés de los involucrados”. (Ochoa, 2011)

Los elementos dentro una vista a tratar en este trabajo son:

- *Definición de las bases de datos:* Definición estructural de la base de datos.
- *Seguridad de datos:* Como garantizar la accesibilidad de los datos así como la recuperación ante caída del sistema.
- *Normas del diseño:* Normas que impone la arquitectura de datos para el diseño de la solución.
- *Réplica de datos:* Elementos para la comparación de herramientas y soluciones para garantizar la integridad de información en sistemas distribuidos.

- *Script de instalación:* Aspectos para garantizar la instalación de un sistema modular.
- *Tecnología:* Áreas tecnológicas que permiten desarrollar y desplegar la base de datos.
- *Formalización:* Es lo que debe quedar para la documentación de la solución y donde debe ir.

## **2.4 Estructura de la capa de datos**

Debido a las restricciones de la solución, la estructura de la capa de datos tendría un ORM<sup>18</sup> encargado de la manipulación de las entidades de datos y las consultas de menor complejidad dejando al gestor de base de datos PostgreSQL las operaciones más complejas, de esta forma se dividiría el trabajo y no se sobrecargaría el servidor de aplicaciones, además de aprovechar las facilidades y potencialidades del DBMS<sup>19</sup>. (Ver [anexo 1](#))

## **2.5 Normas del diseño**

Las normas del diseño ayudan a crear un lenguaje único garantizando la disminución de incongruencias entre los datos, mejorando así la comunicación entre los módulos y por ende la interoperabilidad de ellos.

### **2.5.1 Patrones de solución**

Los patrones identifican problemas comunes y soluciones alternativas presentes junto con sus formas de hacerlo ya que son el fragmento de un modelo que es recurrente, además que puede plantearse que un patrón es una solución a un problema específico que se ha mantenido a pesar del tiempo. (Blaha, 2010)

En este trabajo se tratan algunos de los escenarios más comunes en el desarrollo de la arquitectura de datos.

#### **2.5.1.1 Administración de las conexiones**

La administración de las conexiones es manejada a través del sistema de seguridad ACAXIA centralizando todas las peticiones al gestor. La solución consiste en construir tantas conexiones como sean necesarias propiciando contextos cortos que permitan una mayor concurrencia a la base de datos, pero además permite crear conexiones por usuarios con tiempos de vida limitados y por último conexiones por subsistemas y en el caso de que un usuario deba acceder a varios de ellos se debe crear un rol que herede los permisos que le permitan desarrollar sus actividades normalmente.

---

<sup>18</sup> Object-Relational-Mapping.

<sup>19</sup> Data Base Manager System.

### 2.5.1.2 Concurrencia en el acceso a datos

La concurrencia<sup>20</sup> en los sistemas informáticos es muy común y por ello es necesario controlar el acceso a los datos de forma concurrente. (Ver figura 1)

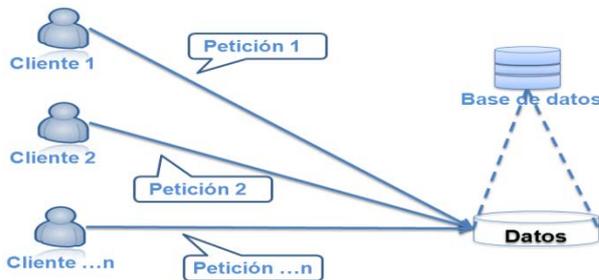


Figura 1 Concurrencia en el acceso de datos.

Para resolver esta situación se debe colocar en cada tabla (no a los nomencladores) un campo versión que permita el control de las modificaciones. El control solo debe ser sobre consultas de tipo lectura y escritura donde se chequee dicho campo. En caso de actualización sobre los datos y contiene una versión inferior a la actual debe notificar al usuario para que actualice los datos. El control de concurrencia puede ser de tipo pesimista u optimista. Para la primera versión de este sistema se va realizar un control de tipo pesimista aplicándolo desde la base de datos. Ejemplo de cómo sería en el Sistema CEDRUX:

El campo “*version*” es de tipo *NUMERIC(6,0)*, con un límite *X*, el cual informará las veces que la tupla se ha modificado. A continuación se muestra una tabla que controla el saldo de las cuentas bancarias. (Ver tabla 2.1)

Tabla 2.1 Tabla con control de concurrencia.

Campos de la tabla	Datos		
identificador	11001	11002	11003
nombre	Cuenta de ahorro	Cuenta de gastos	Cuenta de cobros
Saldo	\$1000	\$2000	\$500
version	1	2	1

Ahora mediante un *TRIGGER* se comprobará la versión que pone el cliente es igual a la antigua en caso verdadero se incrementará en uno el valor del campo versión (el *TRIGGER* ejecutado antes de realizar la modificación), de lo contrario se lanzaría una excepción. (Ver [anexo 3](#))

### 2.5.1.3 Tablas resúmenes

En todo sistema cuando se normaliza la base de datos se gana en almacenamiento puesto que se logra menor redundancia de datos, se gana en la integridad y mejora la consistencia de los datos pero a su vez se pierde en rendimiento, que para conformar

<sup>20</sup> Dos o más usuarios acceden a los mismos datos en el mismo instante.

los datos de algún objeto se debe consultar en varios lugares degradando así el tiempo de respuesta. Por ello es necesario la creación de tablas resúmenes que contengan la información consolidada en un mismo lugar, esto es para el caso de datos que contienen operaciones matemáticas que implicarían un costo adicional al tiempo de respuesta y el caso de procesamiento analítico de los datos para la toma de decisiones.

#### 2.5.1.4 Atributos dinámicos

En los sistemas de gestión empresarial es muy utilizado el trabajo con conceptos dinámicos, de manera que se permite variar la estructura de un objeto determinado en función de las configuraciones del usuario. Para resolver este escenario se definió la siguiente estructura del patrón EAV<sup>21</sup>:

- *Tabla de campos*: estructura detallada sobre los campos o atributos asociados a un objeto o clase.
- *Tabla de categoría de tabla*: clasifica a los objetos o clases.
- *Tabla de fila*: es quien identifica a un objeto o clase dentro de una tabla.
- *Tabla de valores*: persiste el valor que toma los nuevos atributos.

Esta estructura dinámica de datos se aplicó en el esquema de estructura y composición para la creación de entidades de diferentes tipos. (Ver [anexo 4](#))

#### 2.5.1.5 Árboles

Los árboles permiten definir jerarquía entre objetos donde se conoce el nivel de la estructura o no. Para la construcción de este sistema se utilizaron dos tipos de árboles los cuales se definen a continuación.

Para la estructura conocida:

- *Árboles fuertemente codificados* (Hard code dtree): Una entidad para cada nivel del árbol y con relaciones de uno a mucho. (Blaha, 2010)(Ver figura 2)

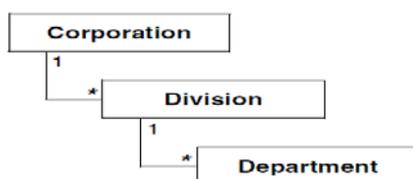


Figura 2 Árbol fuertemente codificado. (Blaha, 2010)

Para la estructura desconocida:

- *Árbol simple*: Tiene posibilidad de “n” niveles de la estructura y relaciones de “0” a mucho. (Blaha, 2010)(Ver figura 3)

---

<sup>21</sup>Entity-Attribute-Value

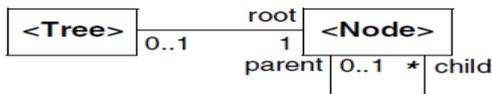


Figura 3 Árbol simple. (Blaha, 2010)

Con diferencia al que se conoce los niveles que se aplicó como planteaba la literatura, al árbol simple se le agregaron nuevos atributos que permitían acelerar los procesos de búsqueda de los nodos como son:

- **idarbols**: identificador del nodo.
- **idapadre**: identificador del padre. En caso de presentar el mismo “**idarbols**”= “**idapadre**” quiere decir que es la raíz del árbol.
- **ordenizq** y **ordender**: permite el conocimiento de todos los hijos pertenecientes a un determinado padre y viceversa. (Ver figura 4)

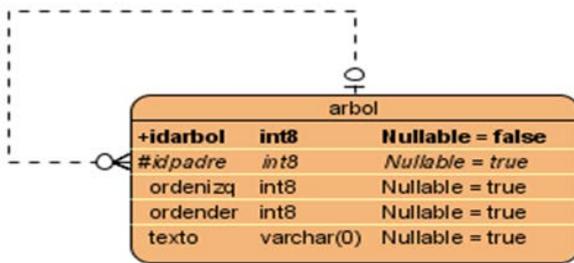


Figura 4 Árbol simple aplicado en CEDRUX.

Para el caso de la inserción quedaría así:

- Donde el hijo siempre toma el “**ordender(padre)**” del padre como su “**ordenizq(hijo)**” y su “**ordender(hijo)**” = “**ordenizq(hijo)**” + 1, quedando el “**ordender(padre)**” del padre como “**ordender(padre)**”= “**ordender(padre)**” + 2. (Ver figura 5)

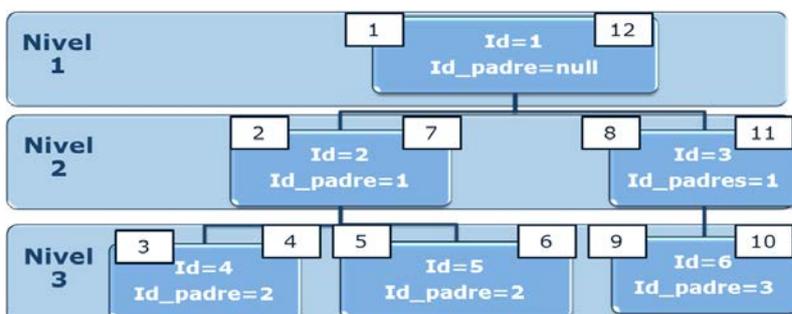


Figura 5 Árbol con hijos.

### 2.5.1.6 Procesamiento de los datos

Los análisis de la información como requieren de recursos se realizan a nivel de

funciones programadas y se muestran ya de forma resumida a la capa funcional. Dichas funciones se encargan además de rellenar las tablas resúmenes que son contenedoras de la información desnormalizada.

#### **2.5.1.7 Normalización y desnormalización de la base de datos**

Para lograr datos consistentes con el mínimo de redundancia se aplica a los diseños de base de datos los procesos de normalización y desnormalización según lo que se quiera obtener, mientras más se normaliza más se pierde en rendimiento y es por ello que se debe llevar el diseño al menos a tercera forma normal, además de identificar las consultas más lentas y aplicarle *tunning*<sup>22</sup> a través del proceso *EXPLAIN*<sup>23</sup>, también en caso de ganar en mucho más crear índices o aplicando desnormalización de forma adecuada y sin abusar.

#### **2.5.2 Comentarios**

Es una necesidad comentar todo lo que se haga dentro de la base de datos, es decir, establecer las pautas que conlleven a lograr un código más legible y reutilizable, de manera que se pueda aumentar su mantenibilidad a lo largo del tiempo. Entiéndase esto como la descripción de cada elemento que se creó en una base de datos (esquemas, funciones, relaciones entre tablas, tablas, etc...).

Para comentar:

- Los comentarios serán tabulados en forma consecutiva y con el siguiente formato:
  - Descripción seguida del 'Nombre de la persona que la realiza'.

No se debe insultar la inteligencia del lector, por lo que se deben tratar con sumo cuidado los detalles que no es necesario conocer o que pueden ser deducidos echando un vistazo al código.

Es importante ser correcto y no perder el tiempo, comentando solo lo necesario para transmitir la idea. Los comentarios deben ser simples y directos.

Se comenta mientras se programa, para lograr eliminar mayores costes en el tiempo.

Los comentarios se actualizarán al mismo tiempo que el código.

En caso de que se quiera emitir algún comentario especial en cualquier segmento del código utilizar *"/\* Comentario \*/*.

#### **2.5.3 Estándares de nomenclatura**

El estándar de nomenclatura es quien rige como debe llamarse cada objeto de base de datos de un sistema, garantizando un estándar para la comunicación entre los

---

<sup>22</sup> Técnica de reajuste y optimización para lograr mayor rendimiento.

<sup>23</sup> Herramienta del Gestor PostgreSQL para evaluar rendimiento de las consultas.

diferentes módulos que la componen.

### 2.5.3.1 Nomenclatura de la base de datos

Los nombres de las bases de datos comienzan con el prefijo “bd” seguido con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación “*Camel casing*<sup>24</sup>”. Con sólo leerlo se reconoce el propósito de la misma.

Ejemplo:

```
bdContMaterial
```

Figura 6 Denominación de las base de datos.

### 2.5.3.2 Esquemas

El nombre a emplear para los esquemas se escribe con todas las letras en minúscula, comenzando por el prefijo “mod”, a continuación el símbolo “\_”, y por último el nombre del módulo.

Ejemplo:

```
create schema 'mod_finanza';
```

Figura 7 Creación de esquemas en PostgreSQL.

### 2.5.3.3 Tablas

El nombre empleado, debe escribirse con todas las letras en minúscula para evitar problemas con el *Case Sensitive*<sup>25</sup> del gestor y con solo leerlo se reconoce el propósito de la misma.

Ejemplo:

```
create table 'nom_producto';
```

Figura 8 Creación de tabla en PostgreSQL.

#### 2.5.3.3.1 Prefijos a utilizar en la creación de tablas

Los prefijos a utilizar en la creación de tablas serán los siguientes:

- *dat\_*: Prefijo utilizado en tablas que almacenan la mayor cantidad de características de una entidad.
- *nom\_*: Prefijo utilizado en tablas nomencladoras.
- *seg\_*: Prefijo utilizado en tablas que almacenan control de acceso, usuarios y opciones de acceso de uno o varios sistemas. (Tablas de Seguridad)
- *conf\_*: Prefijo utilizado en tablas que almacenan parámetros de configuración del sistema. (Tablas de Configuración)
- *tmp\_*: Prefijo utilizado para tablas que almacenan datos transitorios. (Tablas

<sup>24</sup> Estilo de escritura que se aplica a frases o palabras compuestas.

<sup>25</sup> Sensible a las mayúsculas/minúsculas.

Temporales)

- *his\_*: Prefijo utilizado para tablas que almacenan datos por largos períodos de tiempo y que solo son utilizados para análisis esporádicos. (Tablas Históricas)
- *res\_*: Prefijo utilizado para las tablas resúmenes, empleadas en los reportes.

Ejemplo:

```
nom_producto (nomenclador).  
seg_usuarios (seguridad).  
conf_almacen (configuración).
```

Figura 9 Denominación de tablas.

#### 2.5.3.4 Campos

El nombre a emplear para los campos se escribe con todas las letras en minúscula, con solo leerlo se reconoce el propósito del mismo y debe incluir un comentario con su descripción. Si el campo es un identificador debe empezar con “id”.

Ejemplo:

```
add field 'idproducto';  
cantemb: cantidad de embalajes.
```

Figura 10 Apariencia de los campos.

##### 2.5.3.4.1 Llaves primarias

El nombre de las restricciones se escribe con minúscula. Comienza con el identificador “id” seguido el nombre de la tabla todo junto y en minúscula.

Ejemplo:

```
idcuenta (Llave primaria de la tabla dat_cuenta).  
idnomcuenta (Llave primaria de la tabla nom_cuenta).  
idsegcuenta (Llave primaria de la tabla seg_cuenta).  
idhiscuenta (Llave primaria de la tabla his_cuenta).  
idconfcuenta (Llave primaria de la tabla conf_cuenta).
```

Figura 11 Nombre de llaves primarias según nombre de la tabla.

##### 2.5.3.4.2 Llaves foráneas

El nombre de las llaves foráneas se escribe con minúscula y el nombre de la llave primaria de la tabla donde pertenece.

Ejemplo:

```
idcuenta. (Llave foránea de la tabla “dat_cuenta” en la tabla “dat_estructurae”)
```

Figura 12 Nombre de llave foránea.

#### 2.5.3.5 Nombres de las funciones, disparadores, tipos de datos y vistas

El nombre empleado permite con sólo leerlo reconocer el propósito del mismo. Se utilizan los prefijos siguientes para la denominación de funciones, disparadores, tipos de datos y vistas.

- *f\_*: Funciones (Procedimientos Almacenados.)
- *ft\_*: Funciones de disparador.

- *t\_*: Disparador.
- *td\_*: Tipo de datos.
- *v\_*: Vistas.

### 2.5.3.6 Nombre de los tablespace<sup>26</sup>

El nombre se definirá todo en minúscula, empezando con la letra “tbs”, seguido por “\_” y el nombre del módulo junto en minúscula.

Ejemplo:

tbs\_modfinanza\_tablasres (tablespace donde se almacenan las tablas resumen del módulo de finanza)  
 tbs\_modestructura\_indices (tablespace donde se almacenan los índices del módulo de estructura)

Figura 13 Nombre de los tablespace.

### 2.5.3.7 Nombre de los dominios

El nombre se definirá todo en minúscula, empezando con la letra “d” seguido por “\_” y el nombre del dominio, siempre identificando para que tipos de campos.

Ejemplo:

mod\_finanzas.d\_llavesprimarias (dominio de las llaves primarias que pertenece al módulo finanza)  
 mod\_traza.d\_denominacion (dominio de denominación que pertenece al módulo traza)  
 mod\_traza.d\_llavesforanea (dominio de las llaves foráneas que pertenece al módulo traza)

Figura 14 Nombre de los dominios.

### 2.5.3.8 Nombre de las secuencias

El nombre de una secuencia se definirá todo en minúscula empezando con “sec” seguido por “\_” y el nombre de la tabla en se utiliza.

Ejemplo:

sec\_datpersona\_seq (secuencia de la llave primaria en la tabla dat\_persona)  
 sec\_datpersona\_codigo\_seq (secuencia del campo código en la tabla dat\_persona)

Figura 15 Nombre de las secuencias.

## 2.5.4 Soluciones genéricas

### 2.5.4.1 Multientidad

Para lograr la multientidad del sistema es necesario la definición estructural de cómo se persisten las entidades dentro de la base de datos, para ello el esquema de Estructura y Composición es el contenedor de la base de esta solución. La definición del conjunto de entidades de datos para modelar esta situación son: (ver figura 16)

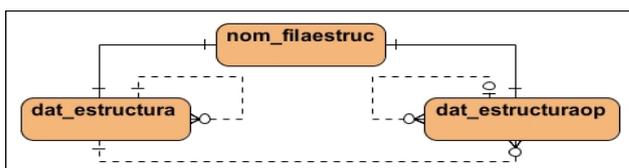


Figura 16 Tablas que comprende la Multientidad.

<sup>26</sup> Espacio donde se almacenan las tablas y los índices.

La integración con el resto del sistema es a través de la tabla “*nom\_filaestruc*”, el identificador de esta tabla debe estar presente en cada tabla donde se gestione información de distintas entidades.

#### 2.5.4.2 Multimoneda

Esta funcionalidad lo que permite es realizar operaciones con distintas monedas manteniendo siempre la tasa de conversión entre ellas. Para esto se diseñó lo siguiente:

- *nom\_moneda*: persiste la clasificación de las monedas. (Ver figura 17)
- *dat\_moneda*: persiste la configuración de la moneda contable del país. (Ver figura 17)
- *nom\_moneda\_entidad*: permite configurar monedas por entidades. (Ver figura 17)

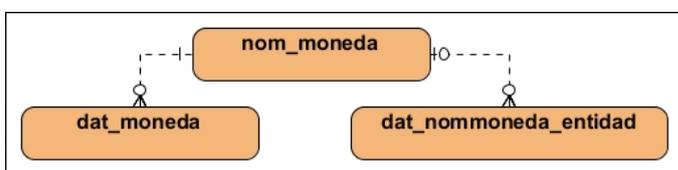


Figura 17 Multimoneda.

Para garantizar que los comprobantes generados por el sistema contengan la moneda en la que se realizó la operación debe contener el identificador de la tabla “*nom\_moneda*” y dos campos importes uno que garantice el importe en la moneda con la que se efectuó la operación y otro con el cambio de ese importe en la moneda contable según su tasa del momento, permitiendo así la dualidad de moneda en las operaciones contables.

## 2.6 Definición de las bases de datos

La definición organizacional de una base de datos es fundamental para la implementación de la misma, debido que permite conocer la estructura y la responsabilidad de cada elemento por el cual está compuesta la misma, además como acceder a ella y con qué objetivo se crean.

### 2.6.1 Estructura de la base de datos

En todo sistema de gestión existen como mínimo tres elementos básicos, los cuales permiten tener una idea general del producto. Dichos elementos son:

- *Configuración*: aspectos que permiten manejar configuración bajo distintos escenarios.
- *Nomencladores*: datos comunes y generales que clasifican parte de la información almacenada.
- *Lógica de negocio*: representación de una situación específica de cada sistema

para dar solución a un problema real.

Teniendo en cuenta estos elementos se define la siguiente estructura organizacional:  
(ver figura 18)

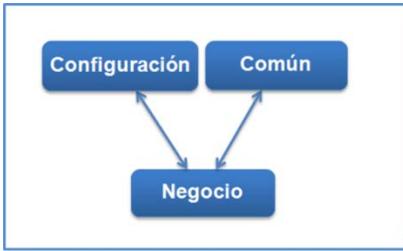


Figura 18 Estructura organizacional de la base de datos en esquemas.

Además de los elementos básicos existen otros esenciales para mantener un sistema estructurado y confiable, como son:

- *Estructura y Composición*: permite crear la estructura de cada entidad dentro de sistema.
- *Trazas*: mantienen un seguimiento de todas las operaciones realizadas por los usuarios.
- *Seguridad*: controla el acceso de los usuarios a las funcionalidades del sistema.
- *Historial*: controla la existencia de los historiales y donde localizarlos.(opcional dependiendo de la vida del sistema)

Agregando a la estructura los elementos esenciales para lograr más robustez, quedaría de la siguiente forma: (ver figura 19)

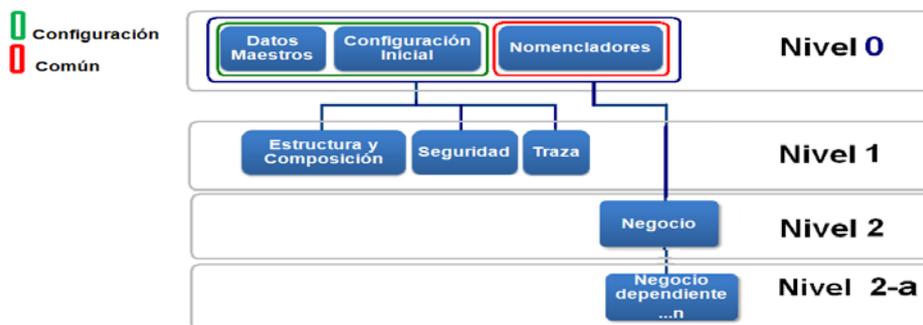


Figura 19 Estructura con robustez

Basado en la estructura planteada anteriormente a continuación se muestra el diseño estructural de la base de datos del CEDRUX en función de los módulos que los componen. (Ver figura 20)

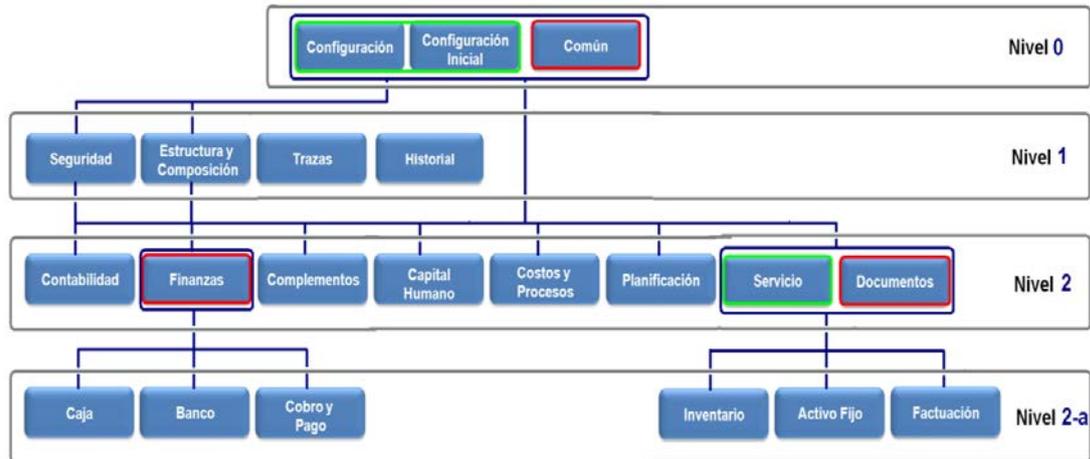


Figura 20 Diseño estructural de la base de datos.

### 2.6.1.1 Responsabilidad de cada esquema organizacional

**Datos Maestros:** Se encarga de persistir todas las configuraciones que se establecen en una entidad permitiendo la personalización y adaptabilidad del sistema, dentro de dicha configuraciones aparecen las operaciones que se realizan con la multimoneda en especial la dualidad de moneda, los formatos de nomencladores, unidades de medida, entre otros.

**Configuración Inicial:** Encargado de almacenar todo el flujo de carga inicial del sistema, desde el estado inicial hasta el final. Ejemplos:

- Flujo de Instalación: Flujo para la instalación inicial de las estructuras y los usuarios que accederán a las mismas.
- Flujo de Configuración: Flujo para realizar las configuraciones iniciales del sistema.

**Nomencladores:** Es quien engloba los nomencladores general del sistema completo, entendiéndose esto como los nomencladores genéricos que comparten varios esquemas.

**Estructura y Composición:** Tiene como objetivo definir estructuralmente la jerarquía de las entidades y la definición interna de dicha entidades; además de definir los cargos y puestos de trabajos por los cuales estará compuesta cada área de una entidad.

**Traza:** Esquema que persiste las operaciones que se realizan al sistema como son: acción, autenticación, excepciones, rendimiento entre otras, que permiten hacer auditorias para saber el estado del sistema.

**Seguridad:** Se encarga de la gestión de la seguridad del sistema, garantizando Autenticación, Autorización, Auditoría, Administración de perfiles y Administración de conexiones.

**Finanzas:** Es el encargado de persistir los elementos comunes a los módulos banco,

caja y cobros y pagos. Se persisten los documentos y los instrumentos con sus estados, sus tipos, y sus operaciones (cambios de cuenta, cambios de moneda, cancelaciones). Se persisten además los anticipos, las liquidaciones de anticipos e instrumentos, y los talonarios con sus tipos.

**Caja:** Permite a las entidades tener una caja central y otras cajas asociadas, teniendo así un mayor control sobre sus recursos. Además es el esquema que almacena todos los procesos financieros de una caja, los cuales se definen como el conjunto de operaciones que se realizan sobre los distintos fondos monetarios con que opera esta.

**Banco:** Persiste todo el control de las actividades bancarias de la entidad. Permite además el análisis del flujo del capital contable de la entidad y una buena contabilización de los procesos mediante los Estados de cuenta y las Conciliaciones bancarias, reportes que posibilitarían la toma de decisiones.

**Cobro y Pago:** Se encarga de relacionar a la entidad con sus clientes y proveedores, que ayuda la gestión y organización de las obligaciones de pago y derechos de cobro que contrae la entidad con clientes, proveedores y con el Estado. Se llevan a cabo las operaciones de liquidación de los cobros y pagos anticipados que han sido reconocidos también por la entidad y se lleva un control de cuánto va quedando pendiente.

**Capital Humano:** Persiste información referente a los recursos humanos, la estimulación moral y material. Además permite la realizar análisis de la información a través de reportes como cálculo de nómina, submayor de vacaciones, submayor de retenciones y comprobante de operaciones.

**Costos y Procesos:** Gestiona los nomencladores de centros de costo, grupos presupuestarios, elementos del gasto y objetos del gasto, además persiste la configuración de las cuentas de gastos tanto patrimoniales como presupuestados y las diferentes asociaciones que se puedan establecer entre ellos.

**Complementos:** Se guardan las funciones de complementos y las tablas que permiten el funcionamiento de los mismos.

**Documento:** Gestiona los documentos primarios que se utilizan en el control de las existencias, así como en los movimientos del producto dentro del almacén: Apertura del almacén, Recepción de productos, Despacho de productos, Inventario físico, Ajuste de inventario, Baja de productos en el almacén y Conciliación entre almacenes.

**Servicio:** Persiste los servicios que ofrece una entidad, la configuración de ellos y las operaciones que se realizan sobre los mismos.

**Inventario:** Gestiona la configuración de los almacenes y los productos asociados a ellos, así como el control del inventario físico.

**Activo Fijo:** Gestiona todas las operaciones que se pueden realizar sobre los activos fijos tangibles dentro de las entidades de forma dinámica.

**Facturación:** Gestiona las facturas que se registran por ventas de los productos, activos fijos tangibles y prestaciones de servicios.

**Contabilidad:** Permite toda la gestión de la información contable, configuración de las cuentas, registro de asientos tipos de todos los módulos que agilizan los procesos contables y la emisión de los estados financieros.

**Planificación:** Persiste la información del proceso de planificación completo, como son planes, modelos, tipos de modelos, comentarios, actividades y sus normas, indicadores y su relación con las entidades y las especialidades. También guarda los centro de balance y sus proveedores. El objetivo del esquema es almacenar la información de la planificación para permitir que esté disponible y accesible en todo momento, además de que este organizada y se pueda acceder eficientemente a la misma.

#### **2.6.1.2 Integración entre los esquemas**

Los esquemas se integran a nivel de base de datos con llaves foráneas para garantizar la integridad de los datos, pero se comunican a través de servicios desde la aplicación. Cada esquema tiene un conjunto de responsabilidades mencionadas anteriormente de donde provee servicios a los demás y según su nivel dentro de la estructura de la base de datos recibe o brinda o ambos inclusive.

### **2.7 Seguridad de datos**

La seguridad de los datos es un aspecto fundamental en cualquier sistema, debido a esto es crucial referirse a dos temas esenciales como son las políticas de los respaldos de los datos y el acceso de los mismos, para desarrollar estos dos temas se tuvo en cuenta algunos de los principios para la seguridad de un sistema como son:

- Seguridad de acceso al sistema.
- Seguridad a nivel de objetos de datos.
- Seguridad a nivel de datos.

#### **2.7.1 Acceso a los datos**

Los datos por lo general son el activo informático máspreciado para cualquier entidad por lo que es de alta prioridad la definición de la misma y su estructura. Para los sistemas de gestión como CEDRUX se definen tres opciones de seguridad de los datos, los cuales son:

- Básico:
  - El sistema debe contar con un módulo que gestione la seguridad de la aplicación (control de acceso a las funcionalidades del sistema).

- El sistema debe contar con al menos rol de conexión sobre la base de datos (rol que permite consultar los datos).
- Medio:

#### Variante 1

- El sistema debe contar con un módulo que gestione la seguridad de la aplicación.
- El sistema debe contar al menos un rol por proyecto.
- El sistema debe contar con un módulo que realice gestión y control de trazas para auditorías de datos.

#### Variante 2

- El sistema debe contar con un módulo que gestione la seguridad de la aplicación.
- El sistema debe contar al menos un rol por esquema organizacional.
- El sistema debe contar con un módulo que realice gestión y control de trazas para auditorías de datos.

- Máximo:

- El sistema debe contar con un módulo que gestione la seguridad de la aplicación (control de acceso a las funcionalidades del sistema).
- El sistema debe contar al menos un rol por cada usuario de la aplicación y roles de agrupación de privilegios sin permiso de conexión.
- El sistema debe contar con un módulo que realice gestión y control de trazas para auditorías de datos.
- La base de datos debe tener un control de acceso a los datos a través de vistas donde se controle el flujo de datos a consultar.

La selección de alguna de estas opciones está dada según el nivel de seguridad que se desee tener en cada sistema, siempre teniendo en cuenta que mientras más se adquiere en seguridad más se pierde en rendimiento.

Para que se entienda mejor el acceso, se muestra a continuación un ejemplo concreto de lo planteado. (Ver figura 21)



Figura 21 Flujo de acceso en CEDRUX.

### 2.7.1.1 Estructura de Roles

Esta estructura sirve para determinar que roles son los que se deben crear y con qué objetivo.

*Escenario 1:*

- Un rol de aplicación por cada proyecto (r\_app).
- Un rol desarrollador por cada proyecto (r\_des).
- Un rol arquitecto por cada proyecto (r\_arq).
- Un rol de administración de las base de datos en desarrollo o en prueba (r\_adminbd).
- Un rol de administración del servidor (r\_adminserver).

*Escenario 2:*

- Un rol de aplicación por cada proyecto desarrollado.
- Un rol de administración de la base de datos.
- Un rol de reporte que permita consultar información y realizar reportes.

### 2.7.1.2 Permisos sobre la base de datos

Teniendo claro los roles de base de datos tanto para el desarrollo y la explotación del software se define los distintos permisos que deben tener cada rol sobre los objetos de base de datos que existan como son tablas, funciones, vistas, etc. (Ver [anexo 2](#))

## 2.8 Réplica

La réplica de datos o copias de datos de una instancia de base de datos a otra es esencial para los sistemas distribuidos como CEDRUX, por ello este epígrafe abarca algunas posibles soluciones para que el sistema sea replicable.

### 2.8.1 Soluciones de réplica

Para que el sistema contenga una base de datos replicable debe al menos contar con algunas de las soluciones que se mencionan a continuación.

Para resolver este problema existen al menos tres escenarios:

- Servidor por rango: Para cada servidor brindar un rango distinto de números para los generadores de clave (secuencias).

Ventajas:

- Control de los servidores del país.

Desventajas:

- No se garantiza la Multientidad. (entidades con identificadores únicos dentro del sistema)
- Cantidad de entidades en un servidor desconocido.
- Tamaño que puede alcanzar la tabla es desconocido. (para poder

asignar los rangos de las llaves)

- Servidor y llaves únicas: Agregar el identificador del servidor a la clave primaria.

Ventajas:

- Simplicidad de la arquitectura.
- Identificadores únicos dentro del sistema completo.
- Control de los servidores del país.
- Control de los datos que se replican por servidores.

Desventajas:

- Costos cambiando lo que se haga mal durante el desarrollo del sistema.  
(no se cumpla con especificaciones de la arquitectura de datos)
- Replicar en tablas separadas, y acceder a los datos a través de una vista formada por la unión de ellas. Para resolver el conflicto de potenciales claves duplicadas en la unión se usará una pseudo columna que representa la base de datos fuente.

Ventajas:

- Control de los datos que se replican por servidores.

Desventajas:

- Costos agregando nuevas lógicas de negocio para el sistema.
- Costos agregando nuevas lógicas de negocio para la base de datos.
- Llaves primarias doubles: Réplica con el identificador del servidor después de la coma.

Ventajas:

- Control de los datos que se replican por servidores.

Desventajas:

- Disparador por tabla replicable.
- Costos agregando nuevas lógicas de negocio para la base de datos.

Por lo que para el sistema se escogió el escenario de servidor y llaves primarias únicas. A continuación se detallará como sería la solución completa.

Solución:

En el esquema de datos maestros se crea una tabla contenedora de todas las secuencias de las tablas replicables del sistema. Cuando se instale el sistema se generará un identificador de la secuencia único que contiene el identificador del servidor más la longitud de la secuencia. Esta tabla tendrá un disparador que se encargará de la creación dinámica de las secuencias. Dicha tabla contiene los siguientes campos: (ver tabla 2.2)

Tabla 2.2 Tabla contenedora de las secuencias de tablas a replicar.

Campos de la tabla	Datos		
nombresecuencia	"sec_version_seq"	"sec_nomtema_seq"	"sec_resumenpase_seq"
valor_inicial	14	6	10
valor_incrementar	1	1	1
valor_final	14	6	10
nombreesquema	"mod_datosmaestros"	"mod_seguridad"	"mod_contabilidad"

El identificador del servidor estaría almacenado en otra tabla con el nombre "nom\_servidor". Una vez insertado las secuencias en la tabla "nom\_secuenciasgeneral" el disparador haría el resto de la solución. (Ver [anexo 5](#))

## 2.9 Script de instalación

El proceso de instalación es algo un poco engorroso si no se tienen claro el proceso de realizar los script<sup>27</sup> puede que el sistema no funcione como debe ser. Para este proceso se define un conjunto de cuestiones que se detallan a continuación.

### 2.9.1 Clasificación de los script

Según la arquitectura que se ha ido definiendo se tiene que:

- Un sistema puede instalar un módulo o varios según el tipo de entidad que sea y su propósito.
- En un sistema ERP como lo es el CEDRUX hay que tener en cuenta las dependencias que tiene cada uno de sus módulos con el resto.

Por estos elementos se define:

- *Tipos de script*: significa el tipo de operación que deben realizar los mismos.
  - Estructurales: realizan operaciones de creación, modificación y eliminación de los objetos de base de datos. (tablas, secuencias, funciones, etc.)
  - Datos: realizan operaciones de manipulación de datos o sea inserción, actualización y eliminación.
- *Clasificación*: consiste en que tipo de información contienen los script.
  - *Rol*: toda la información referente a los roles de base de datos.
  - *Estructura*: toda la información referente a los objetos de base de datos. (tablas, secuencias, funciones, etc.)
  - *Datos*: toda la información referente a los datos iniciales del sistema.
  - *Permisos*: contiene los privilegios de tienen los roles sobre los objetos de la base de datos.
  - *Nomenclatura*: como debe denominarse cada script.

<sup>27</sup> Archivo de órdenes que se almacenan en texto plano.

- *Plantilla*: como debe estar estructura cada script según su clasificación.

Todo lo anterior se debe registrar en la base de datos para tener constancia de todo lo que se ha instalado y su versión. Para ello se define una tabla que registra todos los script que se van instalando, para que esto funcione cada script debe contener esa información. (Ver [anexo 6](#))

### 2.9.2 Plantillas de script

Las plantillas definidas para estos script son para roles y sus dependencias, estructuras y sus dependencias, datos iniciales y sus dependencias, por último privilegios y sus dependencias. Estas plantillas se aplicarían a cada esquema organizacional. (Ver [anexo 7](#))

### 2.10 Tecnología

La selección de las herramientas para el diseño, desarrollo y administración de una base de datos juega un papel muy importante durante el ciclo de vida de un producto de software. Para ello hay que tener en cuenta desde la fase inicial de todo proyecto un grupo de categoría como son:

- Gestor de base de datos
- Diseño
- Desarrollo o implementación
- Administración
- Réplica

Para el desarrollo de esta arquitectura se propone la siguiente infraestructura tecnológica. (ver tabla 2.3)

Tabla 2.3 Requerimientos tecnológicos.

<b>Software</b>	<b>Herramientas de diseño</b>		<b>Cliente de administración</b>		<b>Gestor de base de datos</b>		<b>Réplica</b>	
	Visual v3.1(mínimo)	Paradigm	PgAdmin v1.10(mínimo)	III	PostgreSQL v8.3.8(mínimo)		Reko v2.0(mínimo)	
<b>Hardware</b>	<b>Memoria RAM</b>		<b>Capacidad de disco duro</b>		<b>Procesador</b>			
	1GB (mínimo)		80 GB (mínimo)		Intel Pentium IV (mínimo)	3.0	GHz	

### 2.11 Formalización

La formalización de la solución sería uno de los elementos claves de la resolución de cualquier situación en el desarrollo de un sistema, debido que aquí queda plasmado cada decisión arquitectónica, resultado de pruebas de conceptos, elementos de integración y diccionarios de datos. La formalización garantiza la mantenibilidad de un sistema, debido a que como todo está escrito permite dar soporte a la solución y su posible evolución.

### 2.11.1 Diccionario de datos

Los diccionarios de datos contienen todo el diseño de la base de datos junto con cada descripción de los objetos, claro siempre y cuando se halla descrito anteriormente.

Los elementos básicos dentro de la descripción son:

- *DER*: Diagrama de entidad relación.
- *Resumen*: Resumen de las tablas y su significado.
- *Detalles*: Descripción de cada aspecto de las tablas.
  - Resúmenes de los campos.
  - Campos con todas sus propiedades.
  - Índices de la tabla.
  - Relaciones que contengan.
- *Funciones*: Descripción y su definición.
- *Disparadores*: Descripción y su definición.
- *Secuencias*: Descripción y su definición.

### 2.11.2 Estructura del Expediente

Una vez desarrollado y descrito las soluciones solo quedaría como se almacenaría todos los artefactos tangible para su luego consulta. La propuesta de organización del expediente sería:

- *Dat-Expediente*: contiene el documento rector para el desarrollo de la DB<sup>28</sup> y el expediente con los artefactos generados durante el desarrollo de los proyectos.
  - *Dat-Decisiones*: contiene las decisiones que se han tomado durante el proceso de desarrollo.
  - *Dat-Integración*: contiene un documento rector que se actualiza a través de las matrices de trazabilidad y se documenta toda la integración entre los subsistemas creados.
  - *Dat-Presentaciones*: contiene todas las presentaciones que se desarrollen para dar a conocer la situación de datos, así como su estructura, módulo de producción entre otros.
  - *Dat-Pruebas\_Conceptos\_Servidor*: contiene las pruebas de conceptos realizadas al servidor para lograr obtener una configuración óptima del servidor.
  - *Dat-Vista*: documento rector de datos.
  - *Dat-Proyectos*: contiene todos los artefactos generados por los arquitectos de datos de desarrollo.

---

<sup>28</sup> Data Base.

- *La matriz de trazabilidad:* contiene el flujo de datos entre esquemas.
  - *La matriz de clasificación de entidad de base de datos:* para conocer los tipos de tablas.
  - *Modelos de datos de los proyectos:* diseño de cada solución.
  - *Diccionario de datos:* descripción de la solución.
  - *Pruebas de concepto:* pruebas realizadas para la validación del diseño.
- *Dat-Gestión :* contiene todo lo referente a la gestión del desarrollo de la DB
  - *Dat-Calidad:* contiene las métricas para medir la calidad del desarrollo de la DB y las listas de chequeo para llevar el control del desarrollo.
  - *Dat-Desarrollo:* contiene los contratos para el desarrollo en las líneas de producción, los procedimientos de trabajos asociados a los datos y el sistema de trabajo del grupo de datos.
  - *Dat-Formación:* contiene un documento con el plan de formación previsto para la superación y capacitación del grupo de datos.
- *Dat-Preparación:* contiene todos los materiales y cursos para la capacitación o superación del grupo de datos.
- *Dat-Script-Instalación:* contiene todo lo referente a los script de la DB para la salida de despliegue, así como el control sobre ellos.
  - *Dat-Script\_Control:* contiene el control que se lleva sobre los script de instalación y actualización.
  - *Dat\_Script\_General:* contiene todos los script de instalación y actualización de la DB.

## **2.12 Conclusiones parciales del capítulo**

En el presente capítulo se realizó la definición de la arquitectura de datos para sistemas de gestión. Para la misma se emplean buenas prácticas utilizadas en el mundo de la arquitectura de datos, así como se definen otras muy particulares a partir de soluciones a problemas específicos detectados durante las definiciones tecnológicas o durante la etapa de desarrollo.

Con estas definiciones se logra:

- Utilizar las potencialidades del gestor de base de datos para la manipulación de transacciones concurrentes así como la capacidad del almacenamiento de los datos.
- Delegar responsabilidades garantizando la integridad y seguridad de los datos.
- Generación de reportes a través de funciones o procedimientos de alta

complejidad en el gestor.

- La estandarización de las base de datos.
- La implementación de los niveles de seguridad para los datos en combinación con la aplicación.
- La construcción de un expediente donde almacenar la documentación de la solución.

### **3 CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN**

#### **3.1 Introducción**

En este capítulo se mostrarán los resultados obtenidos a partir de la evaluación de la arquitectura utilizando el método QUASAR. Además se analizarán los resultados arrojados por la aplicación de la solución dentro de un experimento contando con 13 de los 17 proyectos del programa ERP, para ello se hará uso del método estadístico Wilcoxon, el cual utiliza dos muestras relacionadas o apareadas para realizar comparaciones que permitirán conocer si realmente, la aplicación de la solución que se propone en este trabajo ejerce cambios significativo en el proceso de desarrollo de la arquitectura de datos de los proyectos en los cuales se aplicó.

#### **3.2 Método de evaluación de arquitectura. QUASAR<sup>29</sup>**

Los modelos de calidad resultan de utilidad para la predicción de confiabilidad y en la gerencia de calidad durante el proceso de desarrollo, así como para efectuar la medición del nivel de complejidad de un sistema de software. La organización y descomposición de los atributos permite el establecimiento de modelos específicos para efectos de la evaluación de la calidad arquitectónica. (Codorniú, 2010)

El método seleccionado para la evaluación de la propuesta realizada en este trabajo es QUASAR. Para la evaluación se utilizarán las técnicas basadas en escenarios, así como la aplicación del modelo de calidad propio del método. QUASAR lo representa como un modelo jerárquico que formaliza los conceptos de la calidad de un sistema en otros términos que representan los siguientes elementos: (Firesmith, y otros, 2006) (Codorniú, 2010)

- Factor de calidad: es una característica, atributo o propiedad de un sistema o subsistema que lo caracteriza. (Firesmith, y otros, 2006)

Los factores de calidad seleccionados son:

- Interoperabilidad
- Seguridad
- Normas de Diseño
- Portabilidad
- Diseño Estructural

El modelo de calidad representa cada uno de los factores definidos en casos de calidad. Los casos de calidad especifican todas las características que se estudian y validan en la evaluación de un atributo o factor de calidad. Un caso de calidad se especifica con los siguientes elementos: (Firesmith, y otros, 2006)

---

<sup>29</sup> QUASAR: (Evaluación de Calidad de Arquitecturas de Sistemas Complejos por sus siglas en inglés)

- **Afirmación:** constituye una definición que hace la arquitectura donde se especifica que el sistema o subsistema alcanza adecuadamente uno o más factores de calidad. (Codorniú, 2010)
- **Argumentos:** Son las claras y convincentes razones de los desarrolladores que justifican las afirmaciones asociadas. (Firesmith, y otros, 2006) Los Argumentos son una combinación de decisiones arquitectónicas como el uso apropiado de estilos, patrones o mecanismos y los razonamientos por los cuales dichas decisiones son apropiadas y adecuadas. (Codorniú, 2010)
- **Evidencia:** Es la documentación generada que da veracidad a los argumentos especificados. (Firesmith, y otros, 2006) Las evidencias pueden presentarse mediante documentación oficial (diagramas, modelos, etc.) y demostraciones prácticas (escenarios, pruebas, modelos matemáticos, simulaciones, etc.). (Codorniú, 2010)

### **3.2.1 Evaluación de los factores de calidad**

#### **3.2.1.1 Interoperabilidad**

La interoperabilidad es la capacidad que tienen los sistemas de funcionar, de manera satisfactoria, con otros sistemas externos ya sea a través de servicios o accediendo a los datos de los mismos de manera directa. En Cedrux esto se garantiza a través de la coexistencia de los datos y de la integridad. La coexistencia no es más que la aplicación de la multientidad, es decir, la convivencia dentro de una misma base de datos de la información de varias entidades, esto tiene en cuenta que las mismas solo pueden acceder a sus datos y no a los de otras. La integridad es la encargada del intercambio de datos correctamente.

#### **Afirmación**

El almacenamiento de datos del sistema Cedrux permite la coexistencia de los datos de varias entidades así como el intercambio de los mismos de manera correcta.

#### **Argumentos**

Para cumplir con el factor de calidad de interoperabilidad el sistema debe de contar con:

- Módulo que permita definir la estructura de una entidad dentro de un esquema de base de datos, teniendo en cuenta la base de la multientidad abordada en el epígrafe [2.5.4.2](#).
- Esquema que permita gestionar las configuraciones de todas las entidades, conteniendo tablas relacionadas con las llaves únicas de las mismas.
- Para la creación de un nuevo módulo primero debe de integrarse con el módulo que gestiona la definición de la entidad a través de llaves primarias únicas

dentro de todo el sistema.

- Donde quiera que exista una relación entre tablas debe contar con restricciones de llaves foráneas o de restricciones a nivel de negocio para el caso de la integración entre módulos.

### **Evidencia**

Para evidenciar la interoperabilidad entre los datos del sistema el mismo transitó por una rigurosa auditoría de base datos de la cual se obtuvo los siguientes resultados: el 100% de los proyectos de CEDRUX cumplen con la aplicación de la multientidad y en cuanto al intercambio de datos sucede lo mismo, ya que también en el 100% de los proyectos existe el intercambio de datos correctamente. (Ver [anexo 8](#))

#### **3.2.1.2 Seguridad**

Esta variable indica el grado de seguridad con que el sistema reduce o previene la probabilidad o severidad, detecta, reacciona o se adapta al daño malicioso para activos valiosos causados por ataques. Los incidentes de seguridad, las amenazas o los riesgos de seguridad asociados con que el sistema o el subsistema son aceptablemente bajos. (Firesmith, y otros, 2006)

### **Afirmación**

El sistema garantiza el control y acceso a los datos.

### **Argumentos**

El sistema debe contar con:

- Un módulo para llevar el control de la seguridad tanto en la capa operacional y la de datos, a través de la implementación del patrón RBAC<sup>30</sup>. (Blaha, 2010)
- Se debe definir una estructura de roles para garantizar el acceso a las estructuras de datos y los privilegios requeridos.
- Debe al menos tener un nivel de seguridad previamente definido.

### **Evidencia**

En el caso de seguridad la evidencia es mostrada a partir del diagrama del nivel medio de seguridad de acceso a datos, el cual muestra el flujo de acceso a la estructura de datos a cada uno de los módulos que conformarán el sistema. Esto será posible a través del esquema de Seguridad, en el cual se encuentran definidos los roles, las responsabilidades y los privilegios de acceso.

---

<sup>30</sup> Control de Acceso Basado en Roles

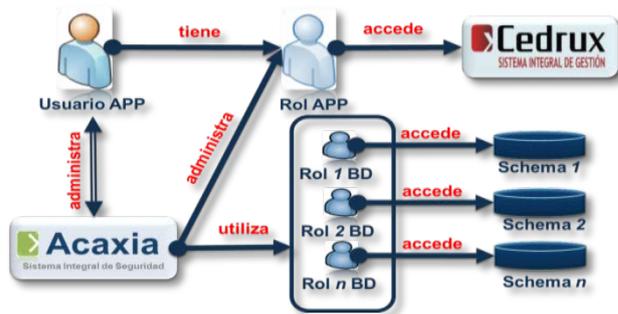


Figura 22 Modelo de datos del módulo seguridad.

### 3.2.1.3 Normas de Diseño

Las normas de diseño logran la estandarización del desarrollo de la base de datos, permitiendo establecer un lenguaje único para el entendimiento de la solución entre el equipo de desarrollo, así como la uniformidad en la estructuración del código que se genere.

#### Afirmación

Con la implantación de las normas de diseño se conocerán la clasificación de las estructuras de datos para luego la localización física de los mismos dentro de las bases de datos. Además evitan que se diseñen soluciones inentendibles, sin un lenguaje común entre el equipo de desarrollo.

#### Argumentos

Las normas de diseño contienen definiciones fundamentales las cuales ayudan a crear un lenguaje único garantizando la disminución de incongruencias entre los datos, mejorando así la comunicación entre los módulos y por ende la interoperabilidad de ellos. Con la misma se logra:

- Administración de las conexiones.
- Concurrencia en el acceso a datos.
- La generación de tablas resúmenes, estructuras arbóreas y atributos dinámicos.
- Procesamiento de los datos.

#### Evidencia

Las normas de diseño fueron aplicadas a cada uno de los subsistemas con los que cuenta Cedrux en estos momentos. Luego de realizar una auditoría a los mismos se arrojaron los siguientes resultados. (ver tabla 3.2)

Tabla 3.1 Aplicación de las normas de diseño.

Objeto	Proyectos	Aplicación de las normas del diseño
1	Estructura y Composición	95
2	Configuración	90
3	Seguridad	100
4	Marco de trabajo	100

5	Contabilidad	99
6	Costos y Procesos	100
7	Caja	91
8	Banco	100
9	Cobros y Pagos	100
10	Capital Humano	100
11	Auditoría	100
12	Inventario	95
13	Mantenimiento	95

Los resultados evidencian que el 97,3% de los objetos de todos los proyectos cumplen con las normas del diseño definido.

#### **3.2.1.4 Portabilidad**

El grado de facilidad con la cual un sistema software o un subsistema pueden ser movidos a un ambiente específico de hardware o software.

##### **Afirmación**

Las definiciones de portabilidad realizadas permiten la instalación del sistema CedruX de manera modular.

##### **Argumentos**

Para lograr la capacidad de instalación modular cada módulo debe tener:

- Script de instalación en su versión modular
  - Contar con sus cuatro script básicos y los que tienen relación con otros módulos.

##### **Evidencia**

Caso 1:

Paquete: Estructura y Composición, Configuración, Contabilidad, Costo.

Entidades: CubaTaxi (despliegue), MINFAR Contabilidad (despliegue).

Caso 2:

Paquete: Estructura y Composición, Configuración, Contabilidad, Costo, Capital Humano.

Entidades: ICID, Empresa Militar Industrial Yuri Gagarin, Centro de Gestión de 10 de Octubre, Hospital Naval, Empresa de Envases Rafael Trejo, UCI.

Caso 3:

Paquete: Estructura y Composición, Configuración, Contabilidad, Costo, Capital Humano, Finanzas.

Entidades: CubaTaxi (primera actualización), MINFAR Contabilidad (primera actualización).

Caso 4:

Paquete: Estructura y Composición, Configuración, Contabilidad, Costo, Capital

Humano, Finanzas, Inventario, Facturación.

Entidades: CubaTaxi (segunda actualización).

Caso 5:

Paquete: Estructura y Composición, Configuración, Inventario, Facturación.

Entidades: MINFAR Logística.

### 3.2.1.5 Diseño Estructural

El diseño estructural permite conocer la estructura y la responsabilidad de cada elemento por el cual se encuentra compuesta, además el acceso y el objetivo con el cual son creadas.

#### Afirmación

El sistema Cedrux se encuentra compuesto por un número de módulos y, gracias a las definiciones estructurales, el mismo es capaz de adicionarlos y desactivarlos sin que ocurra algún fallo en su funcionamiento.

#### Argumentos

La base de datos es distribuida por esquemas que son los contenedores de las especificaciones del negocio. Esto permite contar una base de datos modular, contribuyendo a la modularidad del sistema. Para ello es preciso que se hayan definido los esquemas del sistema por cada área de negocio teniendo en cuenta la complejidad del mismo.

#### Evidencia

El diseño estructural queda evidenciado en la figura 23, en la cual se muestran todos los módulos o subsistemas que integran el sistema Cedrux en los distintos niveles de acuerdo con su importancia, impacto y dependencia dentro de la solución.

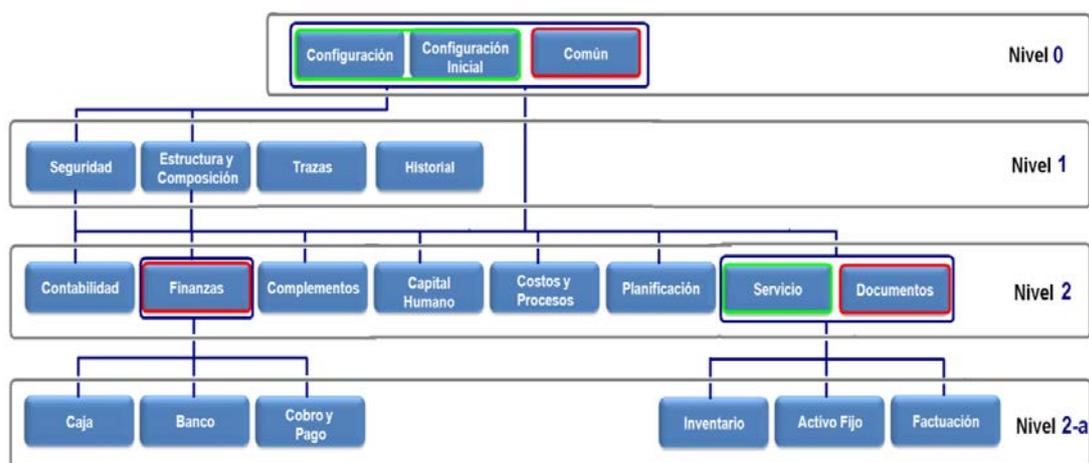


Figura 23 Diseño de base de datos del sistema Cedrux.

### 3.3 Experimento

Para la evaluación de los indicadores se definen dos casos, el primero se aplica al proyecto de Sistema de Gestión Empresarial UCID ya que forma la base de Cedrux de donde se reutilizó información que dio inicio a la definición de esta arquitectura que se plantea y durante el experimento se denomina Cedrux v0.0. (Ver tabla 10). El otro caso es ya aplicado a los proyectos que conforman el programa ERP formando el sistema CEDRUX v1.0.

Tabla 3.2 Reutilización del SGE-UCID

Nombre del proyecto	Porcentaje de Reutilización
<b>Estructura y Composición</b>	15%
<b>Seguridad</b>	10%
<b>Configuración</b>	15%
<b>Contabilidad</b>	50%
<b>Inventario</b>	80%

#### 3.3.1.1 Análisis de los resultados

Según la auditoría, observación y control de la base de datos del programa ERP se obtuvo los siguientes resultados de los indicadores de las variables dependientes e independientes.

##### 3.3.1.1.1 Variables dependientes

##### Interoperabilidad entre los módulos y los datos

Tabla 3.3 Variable Interoperabilidad entre los módulos y los datos.

Objeto	Proyectos	Aplicación de la multientidad (1)	Aplicación de la multientidad (2)	Intercambio de datos correctamente (1)	Intercambio de datos correctamente (2)
1	Estructura y Composición	Nula	Total	Nula	Total
2	Configuración	Nula	Total	Nula	Total
3	Seguridad	Nula	Total	Nula	Total
4	Marco de trabajo	Nula	Total	Nula	Total
5	Contabilidad	Nula	Total	Nula	Total
6	Costos y Procesos	Nula	Total	Nula	Total
7	Caja	Nula	Total	Nula	Total
8	Banco	Nula	Total	Nula	Total
9	Cobros y Pagos	Nula	Total	Nula	Total
10	Capital Humano	Nula	Total	Nula	Total
11	Auditoría	Nula	Total	Nula	Total
12	Inventario	Nula	Total	Nula	Total
13	Mantenimiento	Nula	Total	Nula	Total

Según los resultados arrojados se plantea que el 100% de los proyectos de Cedrux v0.0 no permitían la coexistencia de dos o más entidades en una misma base de datos y en CEDRUX v1.0 se cumple en su totalidad. En cuanto al intercambio de datos

sucede lo mismo, en el 100% de los proyectos de Cedrux v0.0 no existe y en cuanto a CEDRUX si en su totalidad. Por lo que al aplicar la definición de arquitectura de datos se obtiene que sí sea valida la Interoperabilidad entre los módulos y los datos.

### Cubrimiento de necesidades de gestión de datos

Tabla 3.4 Cubrimiento de necesidades de gestión de datos.

Objeto	Proyectos	Grado de cumplimiento de las necesidades de datos (1)	Grado de cumplimiento de las necesidades de datos (2)
1	Estructura y Composición	90	100
2	Configuración	15	100
3	Seguridad	15	100
4	Marco de trabajo	0	100
5	Contabilidad	90	100
6	Costos y Procesos	0	100
7	Caja	0	100
8	Banco	0	100
9	Cobros y Pagos	0	100
10	Capital Humano	0	100
11	Auditoría	0	100
12	Inventario	90	100
13	Mantenimiento	0	100

Analizando los valores del indicador Cubrimiento de necesidades de gestión de datos sobre los casos se muestra que para CEDRUX v1.0 cumple todos los proyectos en un 100% mientras que Cedrux v0.0 solamente en un 22,11% demostrando que la arquitectura planteada cubre las necesidades de la gestión de datos para sistemas de gestión empresarial.

#### 3.3.1.1.2 Variables independientes

##### Estructura organizacional por esquemas

Tabla 3.5 Estructura organizacional por esquemas

Objeto	Proyectos	Estructura organizacional por esquemas (1)	Estructura organizacional por esquemas (2)
1	Estructura y Composición	1	1
2	Configuración	0	1
3	Seguridad	1	1
4	Marco de trabajo	0	1
5	Contabilidad	1	1
6	Costos y Procesos	0	1
7	Caja	0	1
8	Banco	0	1
9	Cobros y Pagos	0	1
10	Capital Humano	0	1
11	Auditoría	0	1
12	Inventario	1	1
13	Mantenimiento	0	1

Analizando los resultados se puede decir que Cedrux v0.0 presenta solamente un 30,76% modular y el CEDRUX v1.0 se encuentra en su totalidad.

## Nivel de seguridad

Tabla 3.6 Nivel de seguridad.

Objeto	Proyectos	Nivel de seguridad (1)	Nivel de seguridad (2)
1	Estructura y Composición	1	2
2	Configuración	1	2
3	Seguridad	1	2
4	Marco de trabajo	1	2
5	Contabilidad	1	2
6	Costos y Procesos	1	2
7	Caja	1	2
8	Banco	1	2
9	Cobros y Pagos	1	2
10	Capital Humano	1	2
11	Auditoría	1	2
12	Inventario	1	2
13	Mantenimiento	1	2

Estos resultados demuestran el nivel de seguridad alcanzado en ambos casos, observando que en el segundo el nivel de seguridad es superior al primero.

## Aplicación de las normas de diseño

Tabla 3.7 Aplicación de las normas de diseño.

Objeto	Proyectos	Aplicación de las normas del diseño (1)	Aplicación de las normas del diseño (2)
1	Estructura y Composición	85	95
2	Configuración	0	90
3	Seguridad	85	100
4	Marco de trabajo	0	100
5	Contabilidad	90	99
6	Costos y Procesos	0	100
7	Caja	0	91
8	Banco	0	100
9	Cobros y Pagos	0	100
10	Capital Humano	0	100
11	Auditoría	0	100
12	Inventario	85	95
13	Mantenimiento	0	95

La tabla anterior demuestra, que después de aplicar las definiciones de la arquitectura de datos a los proyectos se logró una estandarización de las soluciones, elementos que solo se llegaban al 26,53% del cumplimiento.

## Formalización de la solución

Tabla 3.8 Formalización de la solución.

Objeto	Proyectos	Formalización de la solución (1)	Formalización de la solución (2)
1	Estructura y Composición	10	52
2	Configuración	5	54
3	Seguridad	5	26
4	Marco de trabajo	0	20
5	Contabilidad	10	52
6	Costos y Procesos	0	52
7	Caja	0	52

8	Banco	0	52
9	Cobros y Pagos	0	52
10	Capital Humano	0	52
11	Auditoría	0	20
12	Inventario	10	26
13	Mantenimiento	0	38

Luego de la implantación se logró, aunque no en la totalidad, la generación de artefactos referentes la arquitectura de datos en cada uno de los proyectos, con lo que se obtiene el almacenamiento de toda la información que en los mismos se genera. Todo esto es gracias a las definiciones del expediente de arquitectura de datos implantadas en el centro.

### Capacidad de instalación modular

Tabla 3.9 Capacidad de instalación modular.

Objeto	Proyectos	Capacidad de instalación modular (1)	Capacidad de instalación modular (2)
1	Estructura y Composición	1	1
2	Configuración	0	1
3	Seguridad	1	1
4	Marco de trabajo	0	1
5	Contabilidad	1	1
6	Costos y Procesos	0	1
7	Caja	0	1
8	Banco	0	1
9	Cobros y Pagos	0	1
10	Capital Humano	0	1
11	Auditoría	0	1
12	Inventario	1	1
13	Mantenimiento	0	1

Los proyectos del caso 1 se veían como sistemas aislados e independientes, entre los cuales no existía ningún tipo de integración, a partir de estas definiciones es posible tratar a los proyectos como sistemas integrados y no como elementos aislados, creando el sistema integral que se espera obtener como resultado final.

#### 3.3.1.2 Procesamiento estadístico

Aplicando el test de Wilcoxon arrojan los siguientes resultados sobre las variables dependientes. (Ver figura 24)

Tabla 3.10 Siglas de las variables dependientes:

Siglas	variables
AMI	Aplicación de la multientidad Inicial
AMF	Aplicación de la multientidad Final
IDI	Intercambio de datos correctamente Inicial
IDF	Intercambio de datos correctamente Final
GCI	Grado de completamiento de las necesidades de datos Inicial
GCF	Grado de completamiento de las necesidades de datos Final

## Wilcoxon Signed Ranks Test

### Ranks

		N	Mean Rank	Sum of Ranks
AMF - AMI	Negative Ranks	0 <sup>a</sup>	,00	,00
	Positive Ranks	13 <sup>b</sup>	7,00	91,00
	Ties	0 <sup>c</sup>		
	Total	13		
IDF - IDI	Negative Ranks	0 <sup>d</sup>	,00	,00
	Positive Ranks	13 <sup>e</sup>	7,00	91,00
	Ties	0 <sup>f</sup>		
	Total	13		
GCF - GCI	Negative Ranks	0 <sup>g</sup>	,00	,00
	Positive Ranks	13 <sup>h</sup>	7,00	91,00
	Ties	0 <sup>i</sup>		
	Total	13		

- a. AMF < AMI      d. IDF < IDI      g. GCF < GCI  
 b. AMF > AMI      e. IDF > IDI      h. GCF > GCI  
 c. AMF = AMI      f. IDF = IDI      i. GCF = GCI

### Test Statistics<sup>b,c</sup>

		AMF - AMI	IDF - IDI	GCF - GCI
Z		-3,606 <sup>a</sup>	-3,606 <sup>a</sup>	-3,270 <sup>a</sup>
Asymp. Sig. (2-tailed)		,000	,000	,001
Monte Carlo Sig. (2-tailed)	Sig.	,000	,000	,000
	99% Confidence Interval	Lower Bound	Lower Bound	Lower Bound
		Upper Bound	Upper Bound	Upper Bound
		,298	,298	,298
Monte Carlo Sig. (1-tailed)	Sig.	,000	,000	,000
	99% Confidence Interval	Lower Bound	Lower Bound	Lower Bound
		Upper Bound	Upper Bound	Upper Bound
		,298	,298	,298

- a. Based on negative ranks.  
 b. Wilcoxon Signed Ranks Test  
 c. Based on 13 sampled tables with starting seed 1314643744.

Figura 24 Test de Wilcoxon sobre variables dependientes

En la figura anterior se refleja las diferencias que existe al aplicar la definición de arquitectura de datos al sistema CEDRUX v1.0 como varía según los indicadores medidos anteriormente, donde se puede decir que el caso 2 presenta un 100% de las muestras superiores en cuanto al caso1 para la primera variable dependiente y un 100% de superioridad para la segunda variable dependiente. (Ver figura 25

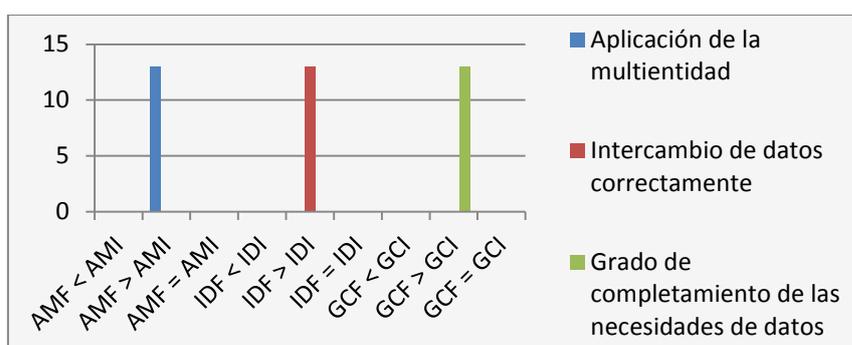


Figura 25 Análisis estadísticos de las variables dependientes.

Aplicando el test de Wilcoxon arrojan los siguientes resultados sobre las variables independientes. (Ver figura 26)

Tabla 3.11 Siglas de las variables independientes:

siglas	Variables
EOEI	Estructura organizacional por esquemas Inicial
EOEF	Estructura organizacional por esquemas Final
NSI	Nivel de seguridad Inicial
NSF	Nivel de seguridad Final
ANDI	Aplicación de las normas de diseño Inicial
ANDF	Aplicación de las normas de diseño Final
FSI	Formalización de la solución Inicial
FSF	Formalización de la solución Final
CII	Capacidad de instalación modular Inicial
CIF	Capacidad de instalación modular Final

### Wilcoxon Signed Ranks Test

		Ranks		
		N	Mean Rank	Sum of Ranks
EOEF - EOEI	Negative Ranks	0 <sup>a</sup>	,00	,00
	Positive Ranks	9 <sup>b</sup>	5,00	45,00
	Ties	4 <sup>e</sup>		
	Total	13		
NSF - NSI	Negative Ranks	0 <sup>d</sup>	,00	,00
	Positive Ranks	13 <sup>e</sup>	7,00	91,00
	Ties	0 <sup>f</sup>		
	Total	13		
ANDF - ANDI	Negative Ranks	0 <sup>g</sup>	,00	,00
	Positive Ranks	13 <sup>h</sup>	7,00	91,00
	Ties	0 <sup>i</sup>		
	Total	13		
FSF - FSI	Negative Ranks	0 <sup>j</sup>	,00	,00
	Positive Ranks	13 <sup>k</sup>	7,00	91,00
	Ties	0 <sup>l</sup>		
	Total	13		
CIF - CII	Negative Ranks	0 <sup>m</sup>	,00	,00
	Positive Ranks	9 <sup>n</sup>	5,00	45,00
	Ties	4 <sup>o</sup>		
	Total	13		

- a. EOEF < EOEI
- b. EOEF > EOEI
- c. EOEF = EOEI
- d. NSF < NSI
- e. NSF > NSI
- f. NSF = NSI
- g. ANDF < ANDI
- h. ANDF > ANDI
- i. ANDF = ANDI
- j. FSF < FSI
- k. FSF > FSI
- l. FSF = FSI
- m. CIF < CII
- n. CIF > CII
- o. CIF = CII

### Test Statistics<sup>b,c</sup>

			EOEF - EOEI	NSF - NSI	ANDF - ANDI	FSF - FSI	CIF - CII
Z			-3,000 <sup>a</sup>	-3,606 <sup>a</sup>	-3,200 <sup>a</sup>	-3,201 <sup>a</sup>	-3,000 <sup>a</sup>
Asymp. Sig. (2-tailed)			,003	,000	,001	,001	,003
Monte Carlo Sig. (2-tailed)	Sig.		,000	,000	,000	,000	,000
	99% Confidence Interval	Lower Bound	,000	,000	,000	,000	,000
		Upper Bound	,298	,298	,298	,298	,298
Monte Carlo Sig. (1-tailed)	Sig.		,000	,000	,000	,000	,000
	99% Confidence Interval	Lower Bound	,000	,000	,000	,000	,000
		Upper Bound	,298	,298	,298	,298	,298

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

c. Based on 13 sampled tables with starting seed 2000000.

Figura 26 Rango y diferencias entre las muestras de las variables independientes

Analizando los resultados arrojados por el test, se alcanza un alto grado de satisfacción por los valores obtenidos después de aplicar la definición arquitectónica de datos, donde casi todas las variables evaluadas cumplen en un 100% excepto la capacidad de instalación modular y estructura organizacional por esquema que solo

cumplen con un 69,23%, pero aun así existe un gran grado de significancia positiva de la aplicación de la solución. (Ver figura 27)

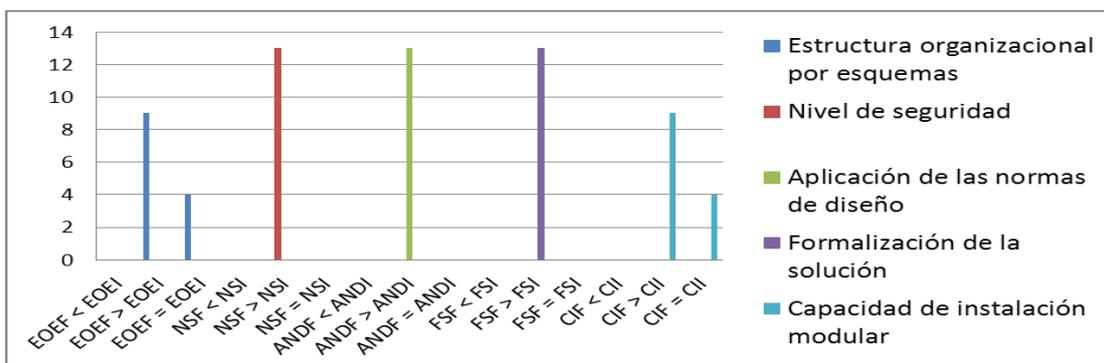


Figura 27 Estadísticas de los indicadores de las variables independientes

### 3.4 Certificación del Sistema

El sistema CERUX es certificado a los 2 días de noviembre de 2011 por la empresa certificadora Conavana donde plantea que el sistema contiene un grado aceptable de adaptación a las normas contables cubanas ( Ver [anexo 9](#)), de donde se concluye:

Los subsistemas analizados:

- Se adaptan en un grado aceptable a las normas cubanas.
- Se garantiza una seguridad razonable a los usuarios.
- Se cumple con las exigencias de ofrecer un procesamiento oportuno y seguro de las operaciones contables.
- Constituye una herramienta de apoyo al trabajo de supervisión y auditoría de los organismos y entidades fiscalizadoras.

Además de Conavana la empresa de seguridad informática (Segurmatica) certifica al sistema CEDRUX con un nivel de **riesgo bajo** a los 3 días del mes de octubre del año 2010 (Ver [anexo 10](#)), donde se concluye:

El sistema:

- Cumple con las exigencias establecidas para las contraseñas, al exigir claves de 7 caracteres, con mayúsculas, minúsculas y caracteres especiales. También se puede agregar que estas características son parametrizables.
- Permite el control de acceso de los usuarios a distintos módulos, asociándolos a grupos, a los que a su vez se les puede restringir a que funciones del sistema están autorizados.
- Considera el almacenamiento de las trazas de auditoría, y permite su consulta estableciendo para la búsqueda diversos parámetros: usuarios, fecha, etc.

- Las salvadas de información no constituyen una información propia del sistema debiendo realizarlas de forma independiente, configurándose externamente por el administrador de base de datos.

### **3.5 Impacto**

La aplicación de la solución propuesta ha permitido estandarizar todos los elementos concernientes a la arquitectura de datos de los proyectos del centro CEIGE como parte de las políticas tecnológicas del mismo. De esta manera no solo se le ha dado respuesta a todos los cómo, sino que también se ha brindado un grupo de soluciones lo más genéricas posibles, a problemas particulares de los sistemas de gestión, identificados, muchos, durante el desarrollo de los proyectos. Esto ha permitido llevar a feliz término los desarrollos, muchos de los cuales se encuentran hoy en proceso de certificación por parte de la entidad Conavana, la cual, dentro de nuestro país, es la entidad de certificadora de software por excelencia y a la espera de la implantación de las mismas en todas las entidades para la gestión de los procesos que en las mismas existen.

Otro de los beneficio es que estas definiciones han ayudado a viabilizar las verticalizaciones realizadas de muchos los proyectos por su genericidad y que hoy se encuentran comercializándose tanto en nuestro país como en la República Bolivariana de Venezuela, como es el caso de Inventario y Mantenimiento Vehicular, aportando importantes sumas que tan necesarias son para el desarrollo de nuestra economía.

Cabe destacar que de estas definiciones no solo se han beneficiado los proyectos del CEIGE sino que las mismas han sido de utilidad en otras áreas dentro de la universidad como es la Dirección Técnica, la cual, en algunas de sus definiciones, ha tomado, como referencia, soluciones aquí planteadas como ejemplo de buenas prácticas y técnicas a seguir en las definiciones de arquitectura que propone específicamente en la vista datos y en el centro de desarrollo de Holguín como parte de sus políticas tecnológicas para sus desarrollos. En muchos de los proyectos que se desarrollan hoy en la UCI también se encuentran presenta estas definiciones, ya que todos aquellos sistemas que utilicen SAUXE como parte de sus soluciones está aplicando y utilizando los elementos descritos en este documento.

### **3.6 Conclusiones**

Después de realizada la validación de la definición de arquitectura de datos a través de las variables dependientes e independientes definidas se puede arribar a las siguientes conclusiones:

- Antes de aplicar la definición de la arquitectura de datos se tenía, entre los proyectos analizados, solo un 22,11%de completitud de los datos, cifra que

cambió a un 100% luego de que se aplicaran estas definiciones. Esta información nos indica que todos los elementos recogidos en el capítulo 2 permiten alcanzar un alto nivel de cubrimiento de los datos una vez que se aplican.

- Otro de los beneficios de los proyectos en los cuales se aplicó el estudio se encontró en la interoperabilidad entre los módulos y los datos. La aplicación de estas definiciones arquitectónicas posibilitaron que el 100% de los proyectos de CEDRUX cumplieren con la aplicación de la multientidad y en cuanto al intercambio de datos sucede lo mismo el 100% de los proyectos existe el intercambio de datos aumentando la posibilidad de lograr estas funcionalidades.
- También se obtuvieron resultados positivos una vez aplicadas estas estas definiciones como:
  - Se logra crear una estructura organizacional por esquema, lo que potencia la portabilidad del sistema permitiendo instalar los proyectos por módulos en los lugares que se necesite.
  - Aumento del nivel de seguridad del sistema de básico a medio, con lo cual se garantiza que los datos almacenados en los mismos no sean dañados ni modificados a no ser a nivel de la aplicación y teniendo los permisos correspondientes para hacerlo. El nivel de seguridad ha sido probado por entidades certificadoras de nuestro país como es Segurmatica, la cual valida y avala, con conclusiones satisfactorias, valida y avala las definiciones realizadas.
  - Con la aplicación de las normas de diseño se logra una estandarización y homogeneización de los datos de un 97,3% dentro de los sistemas. Se ha logrado realizar la estructura organizacional por esquemas en un 100%. Esta solución logra la obtención de la capacidad de instalación modular en un 100%
  - Aunque la formalización de la solución se ha logrado aplicar solo en un 42,15% el mismo ha posibilitado la obtención del 100% de la aplicación de la multientidad, el 100% del intercambio de datos correctamente así como el 100% del completamiento de las necesidades de los datos para el sistema, lo que dice mucho a favor de las definiciones realizadas.

Estos resultados muestran que la arquitectura de datos que se plantea para dar solución al problema da solución al mismo con resultados satisfactorios.

## CONCLUSIONES GENERALES

El desarrollo de este trabajo permitió:

- Conocer, mediante el estudio del estado del arte, en qué consiste la arquitectura de datos y cuáles son los elementos a tener en cuenta para poder definirla correctamente. Realizar una comparación sobre las arquitecturas de datos de algunos de los Sistemas de Gestión Empresarial más usados hoy, tanto a nivel nacional como internacional llevando a importantes conclusiones como que las arquitecturas de datos de estos sistemas no son dependientes de los gestores de bases de datos, es decir, ellos se abstraen de esta capa ya que los mismos son manipulados y manejados por motores de datos que se encuentran dentro de las plataformas que soportan a estos sistemas.
- A partir de las definiciones de la arquitectura de datos realizadas para dar solución a la problemática planteada se lograron, mediante el uso de buenas prácticas y definiciones ajustadas a problemas particulares de nuestro entorno, implantar los elementos definidos en este trabajo a todos los proyectos del centro CEIGE obteniendo resultados satisfactorios con el empleo de la misma que han garantizado el cumplimiento de funcionalidades pactadas en un alto por ciento y con la calidad esperada. Con todo esto se obtuvieron resultados satisfactorios como el 97,31% de la estandarización de las normas de diseño, el 100% de la estructura organizacional por esquemas, así como el 100% de la capacidad de instalación modular. También se logró formalizar esta propuesta en un 42,15%, valor muy significativo dado que el tiempo que lleva de uso de la misma. Además 100% de la aplicación de la multientidad en todos los sistemas, el 100% del intercambio de datos correctamente así como el 100% del completamiento de las necesidades de los datos para el sistema son otros de los logros obtenidos con la aplicación de las definiciones planteadas en este trabajo.

A partir de todos los elementos obtenidos y expuestos en este trabajo se da cumplimiento al objetivo general planteado en el mismo, ya que se realizan las definiciones de arquitectura de datos necesarias para dar solución al problema de la gestión de datos, la interoperabilidad entre los módulos planteado en el arrojando resultados significativos para el centro.

## **RECOMENDACIONES**

En el desarrollo del trabajo existen algunos temas que no han sido analizados con la profundidad requerida a pesar de poseer marcada importancia. Por lo que se recomienda un conjunto de ideas para darle continuidad al trabajo:

- Llevar un riguroso control sobre el desarrollo de los sistemas para lograr una mayor documentación de las soluciones, evitando costo de tiempo en el entendimiento de lo desarrollado para su mantenimiento y evolución.
- Diseñar funciones o procedimientos genéricos para ganar en reutilización y en almacenamiento en disco.
- Implantar el nivel tres o máximo de seguridad en la base de datos de CEDRUX.
- Incorporar métricas para asegurar la calidad del diseño de la solución y desempeño de los arquitectos de datos.

## BIBLIOGRAFÍA

- Adempiere. 2011.** Adempiere. Ecuador. [En línea] 2011. <http://www.adempiere.com.ec/que-es-adempiere.html>.
- ADempiere Ecuador. 2011.** ADempiere Ecuador. [En línea] 2011. <http://www.adempiere.com.ec>.
- Arza, Lizandra. 2006.** *Conclusiones del Taller del ERP cubano*. Dirección Técnica, Universidad de las Ciencias Informáticas. La Habana : s.n., 2006. pág. 13, Acta de evento.
- Blaha, Michael. 2010.** *PATTERNS OF DATA MODELING*. Primera. Boca Raton : Taylor & Francis Group, 2010. pág. 248. Vol. I. ISBN: 978-1-4398-1989-0.
- Brito, Henry Raúl González. 2011.** *Desarrollo del ERP Cubano en la Universidad de las Ciencias Informáticas*. [entrev.] Ing Ariel Torres Gálvez. La Habana, 12 de noviembre de 2011.
- CAMACHO, ERIKA, CARDESO, FABIO y NUÑEZ, GABRIEL. 2004.** *Arquitectura de Software. Guía de Estudio*. 2004.
- Carnicero, Antonio Toledo y Pérez, Pablo Pérez. 2004.** *Gestión de sistemas de información*. s.l. : Universidad de Deusto, 2004.
- Carralero, Lic. Luis Mario, y otros. 2011.** SISTEMA ECONÓMICO INTEGRADO VERSAT-SARASOLA. *FORUM: Fórum de ciencia y técnica*. [En línea] 2011. <http://www.forum.villaclara.cu/ponencias/trabajo/71>.
- CEA. 2006.** CARACTERÍSTICAS HERRAMIENTA E-BUSINESS. Oracle JD Edwards Enterprise One. [En línea] 31 de octubre de 2006. <http://www.cea.es/upload/ebusiness/pdf/herramientas/oracle.pdf>.
- Codorniú, Cesar Lage. 2010.** *Solución arquitectónica de la Configuración*. La Habana : s.n., 2010.
- Compiere. 2009.** Vista general de Compiere ERP & CRM. [En línea] 2009. <http://www.openbiz.com.ar/Compiere%20ERP%20&%20CRM%20-%20Vista%20General.pdf>.
- CONCLUSIONES del Taller ERP Cubano.* **Brito, Henry Raúl González. 2006.** Ciudad de la Habana : s.n., 2006.
- Desoft Habana- Desoft Matanzas. 2007.** *Informe sobre ERPs Libres*. Desoft. Habana : Desoft, 2007. pág. 12, Informe Técnico.
- ERP-Cuba, Proyecto. 2008.** *Presentación ERP-Cuba. Reunión de trabajo con la Ministra del MFP*. La Habana : s.n., 2008.
- Estudios comparativos de paquetes ERP.* **González, Llanos Cuanca y García, Andrés Boza. 2006.** Valencia : X Congreso de Ingeniería de Organización, 2006.
- Fernández, Osmar Leyet. 2011.** *Propuesta metodológica para la obtención de los componentes de software en los proyectos del sistema Cedrux*. 2011.
- Firesmith, Donald, y otros. 2006.** *QUASAR: A Method for the Quality Assessment of Software-Intensive System Architectures*. CMU/SEI-2006-HB-001 Estados Unidos, julio de 2006. Software. Evaluación de arquitectura.

**Gálvez, Ariel Torres y Pérez, Erich M. Gómez. 2008.** *Administración y optimización de un Sistema de Base de Datos Descentralizado, en PostgreSQL.* Universidad de las Ciencias Informáticas. La Habana : Universidad de las Ciencias Informáticas, 2008. pág. 148, Tesis de grado.

**Gálvez, Ariel Torres y Villavicencio, Temis Betancourt. 2010.** *MODELO DE PRODUCCIÓN PARA EL FLUJO DE ARQUITECTURA DE DATOS DEL PROYECTO ERP-CUBA.* 2010.

**Garcia, Ariana Marllery Pinzon y Lopez, Katterine Alzate. 2010.** SOLUCIONES ORACLE JD EDWARDS ENTERPRISEONE. [En línea] 2010. [http://autosystem2010.wikispaces.com/file/view/Soluciones+Oracle+\\_+JD+Edwards+EnterpriseOne.pdf](http://autosystem2010.wikispaces.com/file/view/Soluciones+Oracle+_+JD+Edwards+EnterpriseOne.pdf).

**González, Alberto Carreiras. 2010.** Universidad Autónoma de Barcelona. [En línea] 22 de junio de 2010. [http://ddd.uab.cat/pub/trerecpro/2010/hdl\\_2072\\_114654/PFC\\_AlbertoCarreirasGonzalez.pdf](http://ddd.uab.cat/pub/trerecpro/2010/hdl_2072_114654/PFC_AlbertoCarreirasGonzalez.pdf).

*IMPLEMENTACIÓN DE SISTEMAS ERP, SU IMPACTO EN LA GESTIÓN DE LA EMPRESA Y OTRAS TIC.* Vera, **Ángelo Benvenuto. 2006.** s.l. : CAPIV REVIEW, 2006, CAPIV REVIEW, Vol. 4, pág. 16. ISSN: 0718-4662.

**Jacot, Allen, y otros. 2009.** *JD Edwards EnterpriseOne: The Complete Reference.* s.l. : The McGraw-Hill Companies, 2009. ISBN: 978-0-07-159874-3.

**Jorrín, Michael González. 2011.** *MODELO PARA DISEÑAR ARQUITECTURAS DE SOFTWARE DURANTE EL DESARROLLO DE SISTEMAS DE INFORMACIÓN.* Dirección Técnica de Desarrollo, Universidad de las Ciencias Informáticas. La Habana : Compumat, 2011. pág. 10, Artículo de evento.

**Juaristi, Ana. 2007.** *Procesos de negocio de SAP.* 2007. Toda la información contenida en este manual se ha obtenido de la Web oficial de SAP: [help.sap.com](http://help.sap.com) y de [www.mundosap.com](http://www.mundosap.com).

**Kazman, R., Clements, P. y Klein, M. 2001.** *Evaluating Software Architectures. Methods and case studies.* 2001.

**León, Rolando Alfredo Hernández y González, Sayda Coello. 2002.** *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA.* Primera. Ciudad de la Habana : Editorial Universitaria Eduniv, 2002. pág. 114. Vol. I. ISBN: 959-16-0343-6.

**Ochoa, René Lazo. 2011.** *Modelo de referencia para el desarrollo arquitectónico de sistemas de software en dominios de gestión.* 2011.

**Open Bravo. 2011.** Manual de usuario. *Open Bravo. Opening ERP's future!* [En línea] 1.5, 2011. <http://www.openbravo.com/docs/openbravo-manual-de-Usuario-v1.1.pdf>.

**open bravo. 2007-2010.** OpenBravo opening ERP's future! *OpenBravo opening ERP's future!* [En línea] 2007-2010. <http://www.openbravo.com/es/product/erp/technology/>.

**Openbiz. 2011.** Adempiere. *Openbiz + simple.* [En línea] 2011. <http://www.openbiz.com.ar/adempiere.php>.

—. 2009. *Vista General. ADempiere Business Solution.* Argentina : s.n., 2009.

**OpenERP Spain. 2009.** Informe de evaluación de ERP. *OpenERP Spain*. [En línea] 1.0, 16 de marzo de 2009. [http://www.openerspainspain.com/descargas/ERP\\_report.pdf](http://www.openerspainspain.com/descargas/ERP_report.pdf).

**Oracle. 2011.** Oracle Spain. [En línea] Oracle, 2011. <http://www.oracle.com/es/products/applications/jd-edwards-enterpriseone/index.html>.

—. **2011.** PeopleSoft Enterprise Applications. [En línea] 2011. <http://www.oracle.com/us/products/applications/peoplesoft-enterprise/index.html>.

**Pamungkas, Bayu Cahya. 2009.** *ADempiere 3.4 ERP Solutions. Design, configure, and implement a robust enterprise resource planning system in your organization by using ADempiere*. Birmingham : Packt Publishing Ltd., 2009. ISBN 978-1-847197-26-9.

**Portal de Radio Habana. 2007.** Seminario Nacional con directores de empresas, grupos y uniones. *Radio Habana*. [En línea] 30 de agosto de 2007. <http://www.radiohc.cu/espanol/economia/ago07/reglamento.htm>.

**Porteiro, MSc. Marisel Sosa y Morales, Lic. Pedro H. Cobo.** El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. [En línea] <http://www.disaic.cu/modules.php?name=Content&pa=showpage&pid=818>.

**Ríos, José Carlos del Toro. 2008.** *Documento Visión. Proyecto ERP Cuba*. La Habana : s.n., 2008.

**SAP-España. 2011.** SAP-España. [En línea] 2011. <http://www.sap.com/spain/platform/index.epx>.

**SEI, y otros. 2001.** An Enterprise Information System Data Architecture Guide. [electrónico]. s.l., Estados Unidos : SEI, Octubre de 2001.

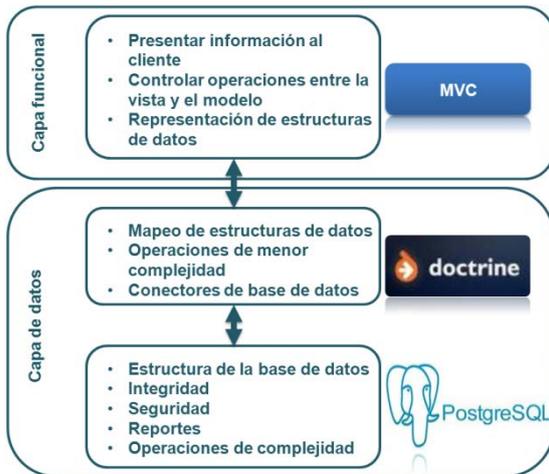
**Systems Limited. 2011.** ORACLE JD Edwards EnterpriseOne ERP System. *Systems Limited*. [En línea] SYSTEMS LIMITED, 2011. <http://systemsLtd.com/products/oracle-jd-edwards-erp.html>.

**Tavira, Hector Solano. 2010.** Estudio, análisis e implementació de SAP Business One en grupo Atlas. [En línea] 2010. [http://bibliotecavirtual.dgb.umich.mx:8083/jspui/bitstream/123456789/1338/1/ESTUDIO\\_ANALISISEIMPLEMENTACIONDESAPBUSINESSONEENGRUPOATLAS.pdf](http://bibliotecavirtual.dgb.umich.mx:8083/jspui/bitstream/123456789/1338/1/ESTUDIO_ANALISISEIMPLEMENTACIONDESAPBUSINESSONEENGRUPOATLAS.pdf).

**UCID. 2008.** *Resumen de proyecto ERP-MINFAR*. La Habana : s.n., 2008.

**Woods, Dan y Word, Jeffrey. 2004.** *SAP NetWeaver For Dummies*. s.l. : John Wiley & Sons, 2004. ISBN:0764568833.

## ANEXOS



Anexo 1 Estructura de CedruX.

[Anexos\Permisos sobre la base de datos.docx](#)

Anexo 2 permisos sobre las base de datos.

[Anexos\Disparador-version.sql](#)

Anexo 3 Disparador de control de versión.

[Anexos\Patrones\Aplicación del patrón EAV para dar soluciones genéricas en el proyecto ERP.pdf](#)

Anexo 4 EAV aplicado en Estructura y Composición.

[Anexos\Secuencias-dinámicas.sql](#)

Anexo 5 Función de creación de secuencias dinámicas.

[Anexos\Script-Instalación\Script-de instalación-modular.docx](#)

Anexo 6 Descripción de Script de Instalación Modular.

[Anexos\Script-Instalación\Plantillas](#)

Anexo 7 Plantillas para los script de instalación.

<https://repositorio.ceige.prod.uci.cu/svn/repo/CSGTEC/Proyectos/Datos/Expediente-Datos/>

Anexo 8 Expediente de Datos.

[Anexos\Certificados\Conavana](#)

Anexo 9 Certificación de Conavana.

[Anexos\Certificados\Segurmatica](#)

Anexo 10 Certificación de Segurmatica.

[Anexos\Estudio de factibilidad .docx](#)

Anexo 11 Estudio de factibilidad.